LOT SIZING WITH PERISHABLE ITEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

By
Nazlıcan Arslan
July 2019

LOT SIZING WITH PERISHABLE ITEMS
By Nazlıcan Arslan
July 2019

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Hande Yaman Paternotte(Advisor)

Oya Karaşan

Esra Koca

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

LOT SIZING WITH PERISHABLE ITEMS

Nazlıcan Arslan
M.S. in Industrial Engineering
Advisor: Hande Yaman Paternotte
July 2019

We address the uncapacitated lot sizing problem for a perishable item that has a deterministic and fixed lifetime. In the first part of the study, we assume that the demand is also deterministic. We conduct a polyhedral analysis and derive valid inequalities to strengthen the LP relaxation. We develop a separation algorithm for the valid inequalities and propose a branch and cut algorithm to solve the problem. We conduct a computational study to test the effectiveness of the valid inequalities.

In the second part, we study the multistage stochastic version of the problem where the demand is uncertain. We use the valid inequalities we found for the deterministic problem to strengthen the LP relaxation of the stochastic problem and test their effectiveness. As the size of the stochastic model grows exponentially in the number of periods, we also implement a decomposition method based on scenario grouping to obtain lower and upper bounds.

Keywords: lot sizing, perishable items, valid inequalities, multistage stochastic programming, decomposition.

ÖZET

ÇABUK BOZULAN ÜRÜNLER İÇİN KAFİLE BÜYÜKLENDİRME

Nazlıcan Arslan
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanı: Hande Yaman Paternotte
Temmuz 2019

Bu çalışmada çabuk bozulan ürünler için kafile büyüklendirme problemi incelenmiştir. Çabuk bozulan ürünlerin belirgin ve sabit raf ömrüne sahip olduğu kabul edilmiştir. Çalışmanın ilk kısmında talebin de bilindiği kabul edilmiştir. Bu problem için çokyüzlü çözümleme yapılmış ve doğrusal programlama gevşetmesini güçlendirmek için geçerli eşitsizlikler formüle edilmiştir. Geçerli eşitsizlikler için ayrıştırma algoritması geliştirilmiş ve problem dal-kesi algoritması kullanılarak çözülmüştür. Geçerli eşitsizliklerin etkisini ölçmek için deneysel çalışmalar yapılmıştır.

Çalışmanın ikinci kısmında ise talebin belirsiz oldugu durum icin çok aşamalı rassal programlama problemi ele alınmıştır. İlk kısımda bulunan geçerli eşitsizlikler rassal problem için de kullanılmış ve deneysel çalışma ile test edilmiştir. Planlama dönemi sayısı arttıkça rassal problemin boyutu üssel olarak artmaktadır. Büyük boyutlu çözülmesi zor problemler senaryo grupları üzerinden ayrıştırılmış ve daha küçük boyutta problemler elde edilmiştir. Küçük boyutlu problemler çözülerek asıl problem için alt ve üst sınırlar elde edilmiştir.

Anahtar sözcükler: kafile büyüklendirme, çabuk bozulan ürünler, geçerli eşitsizlikler, çok aşamalı rassal programlama, ayrışım.

Acknowledgement

I am genuinely grateful to have Prof. Hande Yaman as an advisor. Her continuous guidance, support and encouragement helped me in various ways in my academic journey. I am proud to be one of her students.

I would like to thank Prof. Oya Karaşan and Asst. Prof. Esra Koca for accepting to be in my thesis committee and sharing their valuable suggestions.

It was an honor to be a member of Bilkent University Department of Industrial Engineering. I offer my sincere gratitude to all professor for their efforts. I am thankful for all the graduate students and particularly members of office EA305 for creating a cheerful and pleasant environment.

I feel extremely lucky to have Çağın Ürü because without his support and understanding I wouldn't handle struggles so easily.

I am grateful for my parents Şeminur Arslan, Can Arslan and my sister Pelin Arslan Bilgen for their constant support, encouragement and endless love. I couldn't thank enough for sacrifices they have made throughout my whole studies and life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Finding the best order size is an important problem in production, transportation and procurement systems. Harris [1] introduces this problem to the literature and develops the economic order quantity (EOQ) model where the objective is to find the best order quantity that minimizes fixed setup cost and total holding cost.

EOQ model assumes that the demand is constant over the whole production horizon. The lot sizing problem where the demand varies over time can be seen as a generalization of the EOQ problem.

As mentioned by Jans and Degraeve [2], many variants of the lot sizing problem are studied in the literature to model and solve various industrial and business problems. One of the extensions of the lot sizing problem is obtained by considering the perishable nature of products. Perishable inventory refers to products that lose their value and quality over time and that are disposed after a certain time.

Items can be perishable because of the rapid changes and developments in the technology as in the case of military aircraft or smart-phones. On the other hand, they may deteriorate over time like dairy products, fruits and vegetables [3]. Other examples of perishable items are newspapers, flowers, concert and game tickets.

Deteriorating inventory is also an important concept in health-care systems. Vaccines are perishable products. They are stored in vials and when a vial is opened vaccines can be used only during a safe-for-use time [4].

Blood is also a perishable item. According to the American National Red Cross [5], whole blood is composed of red and white blood cells, platelets and plasma. Whole blood can be kept in the inventory for 21- 35 days after being drawn from the human body. Platelets are highly perishable with five days of shelf-life. Red blood cells can be kept up to forty two days in the inventory. White blood cells are generally not used for transfusion because they can carry viruses that can be harmful to the receiver. Other blood components can be kept in the inventory up to one year.

In the presence of deteriorating inventory minimizing wastage is an important issue. On the other hand, satisfying the demand on time and minimizing the number of shortages are also crucial. For instance, 17% of platelets are disposed in the US in 2004 and there are 492 reported surgery cancellations because of blood shortages in 1700 US hospitals in 2007 [6]. Platelets are very expensive and scarce items to waste. On the other hand, not being able to meet the demand on time may put human lives at risk and it may require more costly solutions. [7].

In addition to the costs incurred due to wastage and shortage, there are other costs involved in the inventory management of perishable items. Suppose we continue with the blood example. Blood units are generally collected in blood centers or mobile units. Collected blood units are tested to make sure that they do not contain any viruses. If they are safe for use, then they are either stored as whole blood or separated into their components [6]. Material and laboratory costs for these procedures can be interpreted as procurement costs. Each blood component requires different inventory conditions to be safely preserved. Inventory holding costs are incurred to create and maintain the necessary conditions [7]. Hospitals place orders to blood centers. Distribution of blood components to the hospitals also has some costs which can be interpreted as fixed cost of procurement.

In this study, we consider the lot sizing problem with perishable items and we take into account both the possibility of wastage and shortage. More precisely, we would like to find a procurement plan for a product with a fixed life time. We consider a given planning horizon over which we know the demand, the fixed and variable costs of procurement as well as costs related to inventory holding, shortage and disposal. Additional constraints on the composition of an order in terms of product ages are also considered.

In Chapter 2, we review the related literature on lot sizing, perishable inventory and stochastic lot sizing.

In Chapter 3, we study the deterministic variant of our problem. We first give a natural formulation of the problem. Then using an optimality property, we derive an uncapacitated facility location formulation. Our computational experiments show that this formulation is much more effective compared to the natural formulation in solving the problem. However, this formulation cannot be extended to take into account any age related constraints. As these are rather common in practice, we study the natural formulation further to improve its strength.

In Chapter 4, we consider the Wagner-Whitin relaxation of the lot sizing formulation and derive valid inequalities for the relaxed problem. The number of these valid inequalities grows exponentially both in the size of the planning horizon and the length of the shelf-life of products. For that reason, we implement a branch and cut algorithm where we separate the valid inequalities using a heuristic. The results of our computational study show that the proposed inequalities are very effective in solving the problem and that the branch and cut algorithm is capable of solving large instances that are out of the reach of a general purpose solver with the natural formulation.

In Chapter 5, we restrict our model by adding constraints on the composition of an order in terms of age and test the performance of the valid inequalities on this extension of the problem.

In Chapter 6, we incorporate demand uncertainty into our problem using multi-stage stochastic programming. We test the effectiveness of the valid inequalities we found for the deterministic problem on solving the stochastic problem and present computational results.

The size of the stochastic problem grows exponentially in the size of the planning horizon. In order to solve large problems, we decompose the scenario set into groups and use these group subproblems to compute lower and upper bounds for the stochastic problem. In Chapter 7, we present this decomposition approach and the outcomes of our computational study where we test the quality of the bounds.

We conclude our study with a summary of our contributions and further research directions in Chapter 8.

Chapter 2

Literature Review

In the classical lot sizing problem, the objective is to find a minimum cost production plan over a planning horizon of n periods. The demand is time-dependent, there is a unit ordering cost, an inventory holding cost and a fixed setup cost for each period. In the uncapacitated lot sizing problem, it is also assumed that the production capacity is infinite.

In their seminal work, Wagner and Whitin [8] study the single-item uncapacitated lot sizing problem. They analyze the properties of optimal solutions and develop a forward algorithm to solve the problem.

Krarup and Bilde [9] show that the uncapacitated lot sizing model can be reformulated as a facility location problem and the linear programming relaxation of this formulation has an integral optimal solution. Barany et al. [10] describe the convex hull of solutions of the uncapacitated lot sizing with the so called (l, S) inequalities.

Pochet and Wolsey [11] examine the single-item lot sizing problem under Wagner-Whitin cost structure. They present integral polyhedra of uncapacitated lot sizing and uncapacitated lot sizing with startup costs with $O(n^2)$ facets. They also describe the integral polyhedra of uncapacitated lot sizing with backlogging and lot sizing with constant capacities.

There are several other studies on lot sizing and its extensions in the literature. Researchers who are interested in these topics can refer to the book of Pochet and Wolsey [12] where detailed analysis on MIP (Mixed Integer Programming) based formulations and algorithms for various production planning problems can be found. There is also a literature review of Brahimi et al. [13] about studies done on single-item lot sizing and its extensions in between years 2004 and 2016.

The concept of perishable inventory is extensively studied in the literature. Nahmias [14] reviews ordering policies for perishable inventory problems in his survey. He examines problems for both single and multiple products with deterministic and stochastic demands. He categorizes the deteriorating inventory problem into two categories; fixed life perishability and random lifetime. Fixed life perishability refers to problems where products have fixed time of expiration, whereas random lifetime refers to problems in which items may decay at any time. In this study, we assume fixed life perishability.

Hsu [15] studies the uncapacitated lot sizing problem with perishable products where inventory decays with a deterioration rate at each period and the unit holding cost is age-dependent. The author represents the problem as a network flow problem with flow loss and comes up with a dynamic programming algorithm to solve it. He assumes that items are produced, not ordered, so they enter the inventory in the production period when they are 1 period old. In our study, we assume that products are purchased. So, they can be at any age when they enter the inventory.

Sargut and Işık [16] consider a similar problem as Hsu [15]. They extend the problem by adding capacity constraints. They explore properties of the optimal solutions and present a dynamic programming based heuristic for their problem.

Önal et al. [17] also study the lot sizing problem with perishable items. They assume that items have a deterministic deterioration rate and examine different mechanisms to allocate products to the customers. They show that the uncapacitated problem can be solved in polynomial time with all allocation mechanisms whereas the capacitated problem is NP-Hard for some mechanism. Önal [18] considers lot sizing with perishable items in a two-level supply chain. He assumes that customers buy the product with the longest shelf-life and present algorithms to solve the problem.

Günpınar and Centeno [6] presents integer programming formulation for the inventory management of blood products at hospitals where the objective is to minimize the total cost as well as cost of shortage and wastage. They also consider the demand uncertainty and model the problem as a stochastic program.

Readers who are interested in perishable inventory can refer to the literature review of Janssen et al. [19] on deteriorating inventory as well as the book of Nahmias [20] on perishable inventory systems.

There is a large number of studies on lot sizing with random problem parameters. Aloulou et al. [21] present various lot sizing problems where uncertainty is involved and state that stochastic programming is a popular way in the literature to represent randomness.

Guan et al. [22] consider the multistage stochastic uncapacitated lot sizing problem. They prove that (l, S) inequalities are also valid for the stochastic problem and extend these inequalities to a more general class of inequalities that are called (Q, S_Q) inequalities. Guan et al. [23] show that (Q, S_Q) inequalities are sufficient to describe the convex hull of the problem with two periods. Summa and Wolsey [24] show that (Q, S_Q) inequalities can be extended as mixing inequalities and examine the cases when the mixing sets describe the convex hull. Zhou and Guan [25] study the stochastic lot sizing problem with Wagner Whitin costs but they consider uncertain costs rather than uncertain demand. They present the integral polyhedron for the problem.

As seen in this review, there are many studies on lot sizing problems and perishable inventory. However, we are not aware of any studies on deriving strong formulations for this problem which is the aim of this thesis.

Chapter 3

Formulations for the Problem of Lot Sizing with Perishable Items

In this chapter, we study the uncapacitated lot sizing problem with perishable items where demand is known. In the first section, we present the notation and a natural formulation. In Section 3.2, we derive an uncapacitated facility location formulation. In Section 3.3, we compare the two formulations with a computational study and observe that the facility location formulation is much more effective than the natural formulation. We conclude with our motivation for further studying the natural formulation.

3.1 Notation and Mathematical Formulation

We are given a planning horizon of n periods and the product has a shelf life of m periods. The demand in period t is d_t units. The ordering and the inventory holding cost in period t for a product of age i are c_{it} and h_{it}, respectively. The disposal cost for deteriorated items is embedded in the holding cost h_{mt}. The fixed cost of procurement in period t is denoted by f_t and the unit cost for shortage in period t is denoted by g_t. For two integers $k \leq l$, we let $[k, l] = \{k, \ldots, l\}$,

$y_{kl} = \sum_{v=k}^{l} y_v$ and $d_{kl} = \sum_{v=k}^{l} d_v$. We define the following decision variables :

x_{it} : the amount of product of age i purchased in period t.

y_t : 1 if there is a setup in period t, 0 otherwise.

q_{it} : the amount of product of age i used to satisfy the demand in period t.

s_{it} : the amount of product of age i in inventory at the end of period t.

r_t : the amount of demand lost (shortage) in period t.

With these variables, the problem can be formulated as follows:

$$\min \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} g_t r_t + \sum_{i=1}^{m}\sum_{t=1}^{n} c_{it} x_{it} + \sum_{i=1}^{m}\sum_{t=1}^{n} h_{it} s_{it} \tag{3.1}$$

$$\text{s.t. } r_t + \sum_{i=1}^{m} q_{it} = d_t \qquad t \in [1, n], \tag{3.2}$$

$$s_{i-1,t-1} + x_{it} = q_{it} + s_{it} \qquad t \in [1, n], i \in [1, m], \tag{3.3}$$

$$x_{it} \leq M y_t \qquad t \in [1, n], i \in [1, m], \tag{3.4}$$

$$s_{it} = 0 \qquad i = 0, t \in [0, n] \text{ or } i \in [1, m], t = 0, \tag{3.5}$$

$$x_{it}, q_{it}, s_{it} \geq 0, \qquad t \in [1, n], i \in [1, m], \tag{3.6}$$

$$r_t \geq 0, y_t \in \{0, 1\}, \qquad t \in [1, n]. \tag{3.7}$$

Let P be natural formulation. Objective (3.1) is the summation of fixed and variable procurement costs, inventory holding and shortage costs. Constraints (3.2) ensure that demand in period t can be satisfied using products of age at most m or can be lost. Constraints (3.3) are the inventory balance equations. If there is procurement in period t then there is a setup in that period due to constraints (3.4). Multiplier M corresponds to a large number to indicate that there is no capacity restriction. In computations, we take M as d_{tl} where $l = t + m - 1$ for period t. Constraints (3.5) indicate that there is no initial inventory and there is no product that has age 0. Constraints (3.6) and (3.7) are range constraints.

3.2 Uncapacitated Facility Location Formulation

In this section, we formulate the problem as a facility location problem. Let z_{jt} be the order given in period j to satisfy the demand in period t. For $t \in [1, n]$ and $j \in [t-m+1, t]$, the minimum cost of satisfying a unit demand in period t from an order in period j can be computed as $c^*_{jt} = \min_{i \in [1, m+j-t]}(c_{ij} + \sum_{u=0}^{t-j-1} h_{i+u,j+u})$. Then the problem can be modeled as follows:

$$\min \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} g_t r_t + \sum_{t=1}^{n} \sum_{j=t-m+1}^{t} c^*_{jt} z_{jt} \tag{3.8}$$

$$\text{s.t. } r_t + \sum_{j=t-m+1}^{t} z_{jt} = d_t \qquad t \in [1, n], \tag{3.9}$$

$$z_{jt} \leq d_t y_j \qquad t \in [1, n], j \in [t-m+1, t], \tag{3.10}$$

$$z_{jt} \geq 0 \qquad t \in [1, n], j \in [t-m+1, t], \tag{3.11}$$

$$r_t \geq 0, y_t \in \{0, 1\} \qquad t \in [1, n]. \tag{3.12}$$

The objective (3.8) is the summation of fixed costs of procurement, shortage costs and variable costs of satisfying demand from an earlier procurement. Constraints (3.9) make sure that demand in period t is either satisfied from earlier orders or it is lost. Constraints (3.10) indicate that if there is an order in period j, then there will be a setup in that period and the products ordered to satisfy the demand of period t would not be more than the demand of that period. Constraints (3.11) and (3.12) are range constraints.

Remark 1. *Both formulations have $O(mn)$ constraints and $O(mn)$ decision variables. However, facility location formulation has fewer number of decisions variables and constraints.*

3.3 Computational Study

We present a computational study for comparing the two formulations. We conduct all computational experiments on Java using Cplex 12.7 on a 64-bit machine with Intel Xeon E5-2630 v2 processor at 2.60 GHz and 96 GB of RAM.

We set a time limit of 1800 seconds. We consider time-invariant costs. Products are grouped into 5 different age categories. Products in an age group have the same procurement cost. Oldest product group has a cost of 1.2 and the cost increases by 0.05 as product groups get younger. We add the disposal cost to the holding cost of period m and it is 0.3. Holding cost is randomly chosen between (0,0.6), shortage cost is chosen between (2,7). Fixed cost is randomly chosen between (41,64) for data set 1 and between (83,127) for data set 2. So only difference between two data sets is that the second on has higher fixed costs. We consider problems with planning horizon $(50, 100, 150, 200)$ and shelf-life $(15, 20, 25)$. We generate instances with four different seeds and present the average results.

We solve the instances using both the natural formulation and the uncapacitated facility location formulation and report the results in Table 3.1. It shows optimality gaps (Opt. Gap), number of branch and bound nodes (Nodes) and the solution times in seconds.

We can see in the table that all our instances are solved to optimality within seconds using the facility location formulation. However, the solver stops with very large optimality gaps when we use the natural formulation. Even though this analysis clearly shows the superiority of the facility location formulation, this formulation fails to model age related constraints as the age of the products are decided a priori by only taking into account the cost. For this reason, in the remaining part of this thesis, our focus is on strengthening the natural formulation to make it more effective in solving realistic problems.

Table 3.1: Comparison of the Facility Location and the Natural Formulations

			Facility Location			Natural		
Data Set	n	m	Opt. Gap	Nodes	Time	Opt. Gap	Nodes	Time
1	50	15	0	0	0.13	3.72	8142314	1829.74
	50	20	0	0	0.03	5.28	7685454	1822.80
	50	25	0	0	0.03	5.35	4833217	1814.75
	100	15	0	0	0.04	5.77	285995	1801.24
	100	20	0	0	0.04	7.07	129448	1801.00
	100	25	0	0	0.04	8.54	54078	1800.49
	150	15	0	0	0.06	6.78	145796	1800.31
	150	20	0	0	0.05	8.19	65218	1801.48
	150	25	0	0	0.07	9.52	34843	1800.53
	200	15	0	0	0.05	7.36	95956	1800.27
	200	20	0	0	0.07	8.53	43327	1800.38
	200	25	0	0	0.08	9.89	20672	1800.45
2	50	15	0	0	0.13	10.59	8226153	1832.75
	50	20	0	0	0.03	10.31	6899808	1813.81
	50	25	0	0	0.04	13.49	5868908	1814.99
	100	15	0	0	0.04	13.62	93672	1800.48
	100	20	0	0	0.05	14.90	66253	1803.51
	100	25	0	0	0.04	16.74	37035	1800.32
	150	15	0	0	0.06	15.34	35234	1800.70
	150	20	0	0	0.05	16.96	28995	1801.03
	150	25	0	0	0.07	18.31	19243	1800.32
	200	15	0	0	0.06	16.85	18912	1800.30
	200	20	0	0	0.09	18.38	8966	1800.29
	200	25	0	0	0.09	19.43	10313	1800.67

Chapter 4

Lot Sizing with Perishable Items Under Wagner-Whitin Costs

In this chapter, we consider the uncapacitated lot sizing problem with Wagner-Whitin costs and derive valid inequalities to strengthen the LP relaxation.

4.1 Wagner Whitin Relaxation

We can substitute $r_t = d_t - \sum_{i=1}^{m} q_{it}$ and $x_{it} = q_{it} + s_{it} - s_{i-1,t-1}$ to obtain a formulation in the space of s, q and y.

After the substitution, the ordering and storage costs become:

$$\sum_{i=1}^{m}\sum_{t=1}^{n} c_{it}x_{it} + \sum_{i=1}^{m}\sum_{t=1}^{n} h_{it}s_{it}$$

$$= \sum_{i=1}^{m}\sum_{t=1}^{n} c_{it}(q_{it} + s_{it} - s_{i-1,t-1}) + \sum_{i=1}^{m}\sum_{t=1}^{n} h_{it}s_{it}$$

$$= \sum_{i=1}^{m}\sum_{t=1}^{n} c_{it}q_{it} + \sum_{i=1}^{m}\sum_{t=1}^{n}(c_{it} + h_{it} - c_{i+1,t+1})s_{it},$$

where $c_{i+1,n+1} = 0$ for $i = 1, .., m$ and $c_{m+1,t+1} = 0$ for $t = 1, .., n$.

The shortage cost becomes:

$$\sum_{t=1}^{n} g_t r_t = \sum_{t=1}^{n} g_t(d_t - \sum_{i=1}^{m} q_{it}) = \sum_{t=1}^{n} g_t d_t - \sum_{t=1}^{n}\sum_{i=1}^{m} g_t q_{it}.$$

We define $K = \sum_{t=1}^{n} g_t d_t$, $h'_{it} := (c_{it} + h_{it} - c_{i+1,t+1})$ and $c'_{it} := c_{it} - g_t$. The equivalent model for problem P in the space of q, s, and y is as follows:

$$K + \min \sum_{t=1}^{n} f_t y_t + \sum_{i=1}^{m}\sum_{t=1}^{n} c'_{it} q_{it} + \sum_{i=1}^{m}\sum_{t=1}^{n} h'_{it} s_{it} \tag{4.1}$$

$$\text{s.t. } \sum_{i=1}^{m} q_{it} \leq d_t \qquad t \in [1, n], \tag{4.2}$$

$$q_{it} + s_{it} \geq s_{i-1,t-1} \qquad t \in [1, n], i \in [1, m], \tag{4.3}$$

$$q_{it} + s_{it} - s_{i-1,t-1} \leq M y_t \qquad t \in [1, n], i \in [1, m], \tag{4.4}$$

$$s_{it} = 0 \qquad i = 0, t \in [0, n] \text{ or } i \in [1, m], t = 0, \tag{4.5}$$

$$q_{it}, s_{it} \geq 0 \qquad t \in [1, n], i \in [1, m], \tag{4.6}$$

$$y_t \in \{0, 1\} \qquad t \in [1, n]. \tag{4.7}$$

Now we consider a relaxation of our problem. Let problem R be

$$K + \min \sum_{t=1}^{n} f_t y_t + \sum_{i=1}^{m}\sum_{t=1}^{n} c'_{it} q_{it} + \sum_{i=1}^{m}\sum_{t=1}^{n} h'_{it} s_{it} \tag{4.8}$$

$$\text{s.t. } s_{i-1,t-1} + \sum_{v=t}^{l} M y_v \geq \sum_{v=t}^{l} q_{i+v-t,v} \qquad t \in [1, n], i \in [1, m], l \in [t, t+m-i], \tag{4.9}$$

$$\sum_{i=1}^{m} q_{it} \leq d_t \qquad t \in [1, n], \tag{4.10}$$

$$s_{it} = 0, \qquad i = 0, t \in [0, n] \text{ or } i \in [1, m], t = 0, \tag{4.11}$$

$$q_{it}, s_{it} \geq 0, \qquad t \in [1, n], i \in [1, m], \tag{4.12}$$

$$y_t \in \{0, 1\}, \qquad t \in [1, n], \tag{4.13}$$

Example 1. *Suppose that $m = 5$, $t = 7$, $l = 8$ and $i = 4$. Constraint (4.9) is*

$$s_{36} + M(y_7 + y_8) \geq q_{47} + q_{58}.$$

This constraint says that the products of age 4 used to satisfy the demand in period 7 and the ones of age 5 used to satisfy the demand in period 8 were either in the stock in period 6 and they had age 3 or there was a purchase in periods 7 and/or 8. Note that the inventory balance equations

$$s_{36} + x_{47} = q_{47} + s_{47}$$

and

$$s_{47} + x_{58} = q_{58} + s_{58}$$

together give

$$s_{36} + x_{47} + x_{58} = q_{47} + q_{58} + s_{58}.$$

Now since $s_{58} \geq 0$, $x_{47} \leq My_7$ and $x_{58} \leq My_8$, we get $s_{36}+M(y_7+y_8) \geq q_{47}+q_{58}$.

Proposition 1. *Problem R is a relaxation of problem P.*

Proof. It is easy to check that the objective functions of the two problems have the same value for all feasible solutions of P. Let (x, s, q, r, y) be a feasible solution of P. We would like to show that (s, q, y) is a feasible solution of R. To this aim, it is sufficient to show that (s, q, y) satisfies constraints (4.9). Let $t \in [1, n], i \in [1, m], l \in [t, t+m-i]$. Summing constraints (3.3) from (i, t) to $(i+l-t, l)$ gives

$$s_{i-1,t-1} + \sum_{v=t}^{l} x_{i+v-t,v} = \sum_{v=t}^{l} q_{i+v-t,v} + s_{i+l-t,l}.$$

Using $s_{i+l-t,l} \geq 0$ and $x_{i+v-t,v} \leq My_v$ for $v \in [t, l]$, we obtain (4.9). □

Proposition 2. *If $h'_{it} \geq 0$ for all $i \in [1, m]$ and $t \in [1, n]$, then problems P and R have the same optimal value.*

Proof. Let (s, q, y) be an optimal solution to problem R such that for all $i \in [1, m]$ and $t \in [1, n]$ for which $s_{i-1,t-1} > 0$, there exists $l \in [t, n]$ with $i + l - t \leq m$ for which constraint (4.9) is tight. We will show that the solution (x, s, q, r, y) where $x_{it} = q_{it} + s_{it} - s_{i-1,t-1}$ for $i \in [1, m]$ and $t \in [1, n]$ and $r_t = d_t - \sum_{i=1}^{m} q_{it}$ for $t \in [1, n]$ is feasible for P. As the two solutions have the same objective function value in their respective problems, this will prove that (x, s, q, r, y) is optimal for P and that the two problems have the same optimal value.

It is easy to see that $r_t \geq 0$ for all $t \in [1, n]$. Now, we will show that $x_{it} \geq 0$ and $x_{it} \leq My_t$ for all $i \in [1, m]$ and $t \in [1, n]$.

Let $i \in [1, m]$ and $t \in [1, n]$. We first show that $x_{it} \geq 0$. If $s_{i-1,t-1} = 0$, then $x_{it} = q_{it} + s_{it} - s_{i-1,t-1} \geq 0$. If $s_{i-1,t-1} > 0$, then there exists $l \in [t, n]$ such that $i + l - t \leq m$ and $s_{i-1,t-1} = \sum_{v=t}^{l} q_{i+v-t,v} - \sum_{v=t}^{l} My_v$. So $x_{it} = s_{it} - \sum_{v=t+1}^{l} q_{i+v-t,v} + \sum_{v=t}^{l} My_v$. As $s_{it} + \sum_{v=t+1}^{l} My_v \geq \sum_{v=t+1}^{l} q_{i+v-t,v}$ due to constraint (4.9) and $My_t \geq 0$, we have $x_{it} \geq 0$.

Next we show that $x_{it} \leq My_t$. If $s_{it} = 0$, then $x_{it} = q_{it} - s_{i-1,t-1} \leq My_t$ by constraint (4.9) where $l = t$. If $s_{it} > 0$, then there exists $l \in [t+1, n]$ such that $i + l - t \leq m$ and $s_{it} = \sum_{v=t+1}^{l} q_{i+v-t,v} - \sum_{v=t+1}^{l} My_v$. Now $x_{it} = \sum_{v=t}^{l} q_{i+v-t,v} + My_t - \sum_{v=t}^{l} My_v - s_{i-1,t-1}$. Since $\sum_{v=t}^{l} q_{i+v-t,v} - \sum_{v=t}^{l} My_v - s_{i-1,t-1} \leq 0$ by constraint (4.9), $x_{it} \leq My_t$ holds. $\square$

4.2 Valid Inequalities

In this section, we present valid inequalities for the uncapacitated lot sizing problem with perishable items. Suppose that X is the feasible set of problem R.

Theorem 1. *Let $\emptyset \subset L \subseteq [1, n]$ be such that $l_2 - l_1 + 1 \leq m$ where l_1 and l_2 are the smallest and largest elements of L, respectively, $\emptyset \subset V \subseteq [1, m]$, $t \in [(l_2 - m)^+ + 1, l_1]$ and $u_i \in [0, \min\{m - i, l_1 - t\}]$ for all $i \in V$. The inequality*

$$\sum_{i \in V} s_{i-1+u_i, t-1+u_i} + \sum_{v=t}^{l_2} \sum_{l \in L: l \geq v} d_l y_v \geq \sum_{l \in L} \left(\sum_{j=1}^{\min\{m, l-t+1\}} q_{jl} + \sum_{i \in V \setminus \{1\}: i+l-t \leq m} q_{i+l-t,l} \right) \tag{4.14}$$

is valid for set X.

Proof. Let v' be the first buying period in the interval $[t, l_2]$ with $v' = l_2 + 1$ if no such period exists. We have $\sum_{v=t}^{v'-1} y_v = 0$. For $l \in [v', l_2] \cap L$,

$$\sum_{v=v'}^{l_2} \sum_{l \in L: l \geq v} d_l y_v \geq \sum_{l \in L: l \geq v'} \left(\sum_{j=1}^{\min\{m, l-t+1\}} q_{jl} + \sum_{i \in V \setminus \{1\}: i+l-t \leq m} q_{i+l-t,l} \right) \tag{4.15}$$

is satisfied. This is true since for $l \in L$ with $l \geq v'$, either the demand of period l can be satisfied from the purchase period v', i.e., $l - v' + 1 \leq m$, or it cannot be satisfied from any stock purchased earlier than period v' and can only be satisfied from later purchases.

If $v' \leq l_1$, then (4.14) is satisfied since $\sum_{i \in V} s_{i-1+u_i, t-1+u_i} \geq 0$, $\sum_{v=t}^{v'-1} y_v = 0$ and $l \geq v'$ for all $l \in L$.

Now suppose that $v' \geq l_1 + 1$. A demand in period $l \in [l_1, v'-1] \cap L$ can only be satisfied from the stock in period $t-1$ as there is no purchase between periods t and l. For a product of age j in period l to be in stock in period $t-1$, we need $j \geq l - t + 2$. In other words,

$$\sum_{l \in L: l \leq v'-1} \sum_{j=1}^{\min\{m, l-t+1\}} q_{jl} = 0. \tag{4.16}$$

For $i \in V$, we have $t + u_i \in [1, n]$, $i + u_i \in [1, m]$ and $\min\{v'-1, t+m-i\} \in [t+u_i, t+m-i]$. Hence constraint (4.9), which is $s_{i-1+u_i, t-1+u_i} + \sum_{l=t+u_i}^{\min\{v'-1, t+m-i\}} My_l \geq \sum_{l=t+u_i}^{\min\{v'-1, t+m-i\}} q_{i+l-t,l}$, is satisfied for all $i \in V$. Using $\sum_{l=t+u_i}^{\min\{v'-1, t+m-i\}} My_l = 0$, $q \geq 0$ and $t + u_i \leq l_1$, we obtain

$$\begin{aligned} \sum_{i \in V} s_{i-1+u_i, t-1+u_i} &\geq \sum_{i \in V \setminus \{1\}} \sum_{l \in L: l \leq \min\{v'-1, t+m-i\}} q_{i+l-t,l} \\ &= \sum_{l \in L: l \leq v'-1} \sum_{i \in V \setminus \{1\}: i+l-t \leq m} q_{i+l-t,l} \end{aligned}$$

Now adding (4.15), (4.16), $\sum_{v=t}^{v'-1} \sum_{l \in L: l \geq v} d_l y_v = 0$ and

$$\sum_{i \in V} s_{i-1+u_i, t-1+u_i} \geq \sum_{l \in L: l \leq v'-1} \sum_{i \in V \setminus \{1\}: i+l-t \leq m} q_{i+l-t,l}$$

gives inequality (4.14). $\square$

Next we identify cases in which valid inequalities (4.14) are dominated.

Proposition 3. *Assume L is not a singleton and $V \setminus \{1\} \neq \emptyset$. Inequalities* (4.14) *are dominated if for $k \in L$, there exist no $i \in V \setminus \{1\}$ and $\bar{k} \in L \setminus \{k\}$ such that $i+k-t \leq m$ and $i+\bar{k}-t \leq m$ (i.e., $i-1$ age old products at the stock in period $t-1$ would not deteriorate before periods k and $\bar{k}$, so they can be used in both periods).*

Proof. Suppose that L is not a singleton, $V \setminus \{1\} \neq \emptyset$ and there exists $k \in L$ for which no i and $\bar{k}$ satisfying the conditions exist. We divide the set V into two disjoint sets V_1 and V_2 such that $V_1 = \{i \in V : i+k-t > m\}$, $V_2 = \{i \in V : i+k-t \leq m\}$. Products from V_1 deteriorate before period k and cannot be used in that period, whereas products from V_2 can be used in period k.

As $\emptyset \subset V_1, V_2 \subseteq [1, m]$, the inequalities

$$\sum_{i \in V_2} s_{i-1+u_i, t-1+u_i} + d_k y_{tk} \geq \sum_{j=1}^{\min\{m, k-t+1\}} q_{jk} + \sum_{i \in V_2 \setminus \{1\} : i+k-t \leq m} q_{i+k-t,k}$$

and

$$\sum_{i \in V_1} s_{i-1+u_i, t-1+u_i} + \sum_{v=t}^{l_2} \sum_{l \in L \setminus \{k\} : l \geq v} d_l y_v \geq \sum_{l \in L \setminus \{k\}} \left(\sum_{j=1}^{\min\{m, l-t+1\}} q_{jl} + \sum_{i \in V_1 \setminus \{1\} : i+l-t \leq m} q_{i+l-t,l} \right)$$

are valid. Also, notice that for $l \in L \setminus \{k\}$, the set $\{i \in V_2 \setminus \{1\} : i+l-t \leq m\}$ is empty. The summation of these two inequalities give the inequality with L. $\square$

Proposition 4. *If there exists $i' \in V$ with $i'+l_1-t \geq m+1$, then inequality* (4.14) *is dominated.*

Proof. If a product has age i' in period t, then it will have age $i'+l_1-t$ in period l_1. Remember that l_1 is the earliest period in set L. If $i'+l_1-t \geq m+1$, then products of age i' in period t deteriorate before period l_1 and cannot be used in any period $l \in L$. Then,

$$\sum_{i \in V \setminus \{i'\}} s_{i-1+u_i, t-1+u_i} + \sum_{v=t}^{l_2} \sum_{l \in L : l \geq v} d_l y_v \geq \sum_{l \in L} \left(\sum_{j=1}^{\min\{m, l-t+1\}} q_{jl} + \sum_{i \in V \setminus \{1, i'\}} q_{i+l-t,l} \right)$$

holds. Together with $s_{i'-1+u_{i'}, t-1+u_{i'}} \geq 0$, it implies (4.14). $\square$

Proposition 5. *Suppose that* $1 \in V$*. Inequality* (4.14) *is dominated if either one of the followings holds;* $|V| \geq 2$*,* $|L| \geq 2$ *or* $u_1 \neq 0$*.*

Proof. First observe that if $1 \in V$ and $|V| \geq 2$, then the inequality for $V \setminus \{1\}$ dominates the one for V. Now, consider the case where $V = \{1\}$. Then, as $s_{0,t-1} = 0 \leq s_{u_1,t-1+u_1}$, the inequality is dominated if $u_1 \neq 0$. Finally suppose that $|V| = 1$, $u_1 = 0$ and $|L| \geq 2$. Then there exists no stock variable in inequality (4.14). Let $L = \{l_1, ..., l_a\}$. The inequality is the summation of the inequalities for $L_k = \{l_k\}$, $k \in [1, a]$. □

Proposition 6. *Suppose that* $t = 1$*. Inequality* (4.14) *is dominated if any of the following conditions does not hold:* $u_i = 0$ *for all* $i \in V$*,* $L = \{l\}, l \in [1, m]$ *and* $V = [2, m - l + t]$*.*

Proof. Let $t = 1$. For $i \in V$, $s_{i-1+u_i,t-1+u_i} = s_{i-1+u_i,u_i} \geq 0 = s_{i-1,0}$. So the inequality is dominated if there exists $i \in V$ such that $u_i \neq 0$. Now suppose that $u_i = 0$ for all $i \in V$, this means that there exists no stock variable in the inequality with $t = 1$. If $|L| \geq 2$, similar to the proof of Proposition 5, let $L = \{l_1, ..., l_a\}$. The inequality is the summation of the ones with $L_k = \{l_k\}$, $k \in [1, a]$. So for the inequality not to be dominated L must be a singleton.

We know by the definition of inequality (4.14) that $(l_2 - m)^+ + 1 \leq t$. For $t = 1$, $(l_2 - m)^+ \leq 0$. Then $l_2 \leq m$. From L being a singleton, $l_1 = l_2 = l$ and $l \in [1, m]$.

Finally suppose that $u_i = 0$ for all $i \in V$ and $L = \{l\}$ for some $l \in [1, m]$. For $V = [2, m - l + t]$, $t = 1$,

$$\sum_{j=1}^{\min\{m,l-t+1\}} q_{jl} + \sum_{i \in V \setminus \{1\}: i+l-t \leq m} q_{i+l-t,l} = \sum_{j=1}^{l} q_{jl} + \sum_{j=l+1}^{m} q_{jl} = \sum_{i=1}^{m} q_{il}$$

For any other subset $\bar{V} \neq V$, the inequality is implied by the one with V and $q_{il} \geq 0$ for $i \in V \setminus \bar{V}$. □

4.3 Separation of Valid Inequalities

Let (s, q, y) be a vector that satisfies (4.9)-(4.12) and $y \in [0,1]^n$. The separation problem asks to find an inequality (4.14) that is violated by (s, q, y) or to show that all the inequalities (4.14) are satisfied.

First observe that the inequality (4.14) can be rewritten as follows:

$$\sum_{i \in V} s_{i+u_i-1,t+u_i-1} \geq \sum_{l \in L} \left(\sum_{j=1}^{\min\{m,l-t+1\}} q_{jl} + \sum_{i \in V \setminus \{1\}: i+l-t \leq m} q_{i+l-t,l} - d_l y_{tl} \right) \quad (4.17)$$

If $\emptyset \subset V \subseteq [2, m]$, $1 < t \leq l_1 \leq n$ are given, then it is easy to find the set L that maximizes the right-hand side value and the vector u that minimizes $\sum_{i \in V} s_{i+u_i-1,t+u_i-1}$. The corresponding inequality is the most violated one for given V, t and l_1. Unfortunately, the number of possible sets V grows exponentially as the number m increases and enumerating them would be inefficient in terms of computations. Another way to rewrite inequality (4.14) is as follows:

$$\sum_{i \in V} \left(s_{i-1+u_i,t-1+u_i} - \sum_{l \in L: i+l-t \leq m} q_{i+l-t,l} \right) \geq \sum_{l \in L} \left(\sum_{j=1}^{\min\{m,l-t+1\}} q_{jl} - d_l y_{tl} \right) \quad (4.18)$$

If $\emptyset \subset L \subseteq [1, n]$ and $1 < t \leq n$ are given, it is again easy to find the set V and the vector u that minimize the left hand-side of the inequality and the corresponding inequality would be the most violated one for given L and t values. However, it is also inefficient to enumerate all possible sets L.

As a heuristic approach, we propose the following: We assume V has consecutive elements at the beginning. Let $i_1 \in [2, m], i_2 \in [i_1, m]$, $V = [i_1, i_2]$ and $1 < t \leq l_1 \leq n$. We need to find L^* that maximizes the right hand side of inequality (4.17). We know by Proposition 3 that, in order to obtain a non-dominated inequality, for $l_2 \in L$, there should be i and $\bar{k} \in L \setminus \{l_2\}$ such that $i + l_2 - t \leq m$ and $i + \bar{k} - t \leq m$. Without loss of generality, we can assume $i + l_2 - t \leq m$ holds for i_1, (i.e., $i_1 + l_2 - t \leq m$). Then, $L^* \subseteq [l_1, \min\{t + m - i_1, n\}]$ and L^* can be computed as follows:

$$L^* = \left\{ l \in [l_1, \min\{t{+}m{-}i_1, n\}] : \sum_{j=1}^{\min\{m,l-t+1\}} q_{jl} + \sum_{i \in V \setminus \{1\}: i+l-t \le m} q_{i+l-t,l} - d_l y_{tl} > 0 \right\}$$

Now that we have the set L^*, we can calculate the set V^* and the vector u^* that minimize the left-hand side of the inequality (4.18). By Proposition 4, for $i \in V$, $i + l_1 - t \le m$ holds for non-dominated inequalities. Then $V^* \subseteq [2, \max\{m - l_1 + t, 2\}]$ and we can calculate V^* as follows:

$$V^* = \left\{ i \in [2, \max\{m - l_1 + t, 2\}] : s_{i-1+u_i^*, t-1+u_i^*} - \sum_{l \in L: i+l-t \le m} q_{i+l-t,l} < 0, \right\}$$

where $u_i^* = \underset{u_i \in [0, \min\{m-i, l_1-t\}]}{\arg\min} s_{i+u_i-1, t+u_i-1}$ for all $i \in [2, \max\{m - l_1 + t, 2\}]$.

For L^*, V^* and u^* we check if (4.18) is violated or not. If it is violated, then we add the inequality to the model.

Note that $V = \{1\}$ and $t = 1$ are not included in the above separation. By Proposition 5, we know that for $1 \in V$, if $|V| = 1$, $|L| = 1$ and $u_1 = 0$, then inequality (4.14) is non-dominated. Suppose that $V = \{1\}$, $u_1 = 0$ and $t \in [2, n]$. From the definition of inequality (4.14), $(l_2 - m)^+ + 1 \le t$. For $l = l_1 = l_2$, $l \in [t, t + m - 1]$. Inequality (4.14) is $d_l y_{tl} \ge \sum_{j=1}^{m,l-t+1} q_{jl}$ for the given V, u_1, t and L. We check these inequalities for any violation and add the violated ones the the model.

For $t = 1$, if $V = [2, m - l + t]$, $u_i = 0$ for all $i \in V$ and $L = \{l\}$ for $l \in [1, m]$, then inequality (4.14) is non-dominated. Suppose that these conditions hold. The inequality is $d_l y_{1l} \ge \sum_{j=1}^{m} q_{jl}$. We check these inequalities for any violation and add the violated ones to the model.

4.4 Computational Results

We implement a branch and cut algorithm in which violated valid inequalities are added to the problem at the root node of the branch and cut tree.

Problem instances are same with the ones in Chapter 3. Results in Table 4.1 show the average results of four instances with the branch and cut algorithm and Cplex with default settings and dynamic search method. Detailed results for each problem instances can be found in the Appendix. In Table 4.1, we compare the optimality gaps, number of branch and cut nodes and solution times. Adding the valid inequalities reduces the number of branch and cut nodes, improve the optimality gaps and for some instances optimal solution is found before the time limit.

In data set 2, there is a huge variation between optimality gaps for larger instances with the same m and n values but with different seeds. For that reason, for these instances we do not report the average results. Rather, we present results of each instance in Table 4.2.

Both tables 4.1 and 4.2 show the results where the cutting plane phase is repeated until no violated cut is found. Let this be strategy 1. In the majority of the instances strategy 1 gives good results. However, in some instances this creates a tailing-off effect, that is, after some iterations of adding inequalities to the model, adding new ones does not improve the lower bound significantly. Variations in Table 4.2 occur due to tailing-off in some instances. In strategy 2, we stop generating cuts after 5 iterations in which improvement in the optimality gap is less than 1%.

Table 4.3 shows the results for strategy 2 for the instances in data set 2 that actually yield good results with strategy 1. Results are averages of the four different seeds. Although, both strategies give better results than the default Cplex, strategy 1 is better than strategy 2 for these instances because it yields better optimality gaps and fewer number of branch and cut nodes.

Table 4.4 shows the strategy 2 results for the larger instances that do not have good results in strategy 1. Variations in the optimality gaps diminish and the gaps are smaller than default Cplex results with strategy 2.

Table 4.5 shows details of the results of strategy 1 for data set 1 and some instances in data set 2. LP Gap refers to the gap between the best upper bound or the optimal value if it is found and the optimal value of the LP relaxation of the problem. Root gap refers to the gap between the best upper bound or the optimal value and the lower bound obtained before the branching starts. User cuts shows the total number of violated cuts added to the model. Separation time (Sep. Time) shows how much time solver spent on adding violated inequalities. Although, Cplex adds its own cuts and has preprocessing at the root node, LP gap and root gap results show that our inequalities improve the lower bound significantly. Also, the problem is solved in fewer number of branch and cut nodes with the algorithm. However, separation times show that there are still room for improvement in our algorithm.

Table 4.6 shows the detailed results of the two strategies (Strat.) for the larger instance in data set 2. As expected, there are more branch and cut nodes in strategy 2. On the other hand, variations are higher both in the root gap and optimality gap on strategy 1. So strategy 2 is a better approach for these instances.

Computational results of the natural formulation with the branch and cut algorithm are still not as good as the results of the facility location formulation. However, branch and cut algorithm yields better results than solving the natural formulation on Cplex with default settings. After all, our purpose in this chapter was to strengthen the natural formulation so that we would be able to solve more complex problems. In the rest of this thesis, we consider two such examples.

Table 4.1: Results for Branch and Cut Algorithm

			Branch and Cut Algorithm			Cplex		
Data Set	n	m	Opt. Gap	Nodes	Time	Opt. Gap	Nodes	Time
1	50	15	0.00	3	16.84	3.72	8142314	1829.74
	50	20	0.00	5	281.21	5.28	7685454	1822.80
	50	25	0.00	3	191.47	5.35	4833217	1814.75
	100	15	0.02	20	511.46	5.77	285995	1801.24
	100	20	0.01	17	698.99	7.07	129448	1801.00
	100	25	0.00	13	1114.08	8.54	54078	1800.49
	150	15	0.00	47	417.43	6.78	145796	1800.31
	150	20	0.00	30	693.37	8.19	65218	1801.48
	150	25	0.01	12	1534.34	9.52	34843	1800.53
	200	15	0.00	75	370.32	7.36	95956	1800.27
	200	20	0.02	26	1122.41	8.53	43327	1800.38
	200	25	0.23	0	1801.37	9.89	20672	1800.45
2	50	15	0.00	13	52.20	10.59	8226153	1832.75
	50	20	0.00	7	176.86	10.31	6899808	1813.81
	50	25	0.00	9	566.91	13.49	5868908	1814.99
	100	15	0.05	54	559.13	13.62	93672	1800.48
	100	20	0.22	15	1505.00	14.90	66253	1803.51
	100	25	0.26	3	1769.39	16.74	37035	1800.32
	150	15	0.14	49	824.75	15.34	35234	1800.70
	150	20	0.40	14	1788.04	16.96	28995	1801.03
	200	15	0.23	83	1151.41	16.85	18912	1800.30

Table 4.2: Results of Larger Instances in Data Set 2

			Branch and Cut Algorithm			Cplex		
Seed	n	m	Opt.Gap	Nodes	Time	Opt.Gap	Nodes	Time
46	150	25	0.90	0	1800.46	18.93	20870	1800.15
96	150	25	0.06	0	1800.54	15.37	21700	1800.28
146	150	25	4.70	0	1800.25	19.49	9386	1800.43
196	150	25	23.56	0	1800.07	19.43	25017	1800.40
46	200	20	1.17	0	1801.13	19.70	6527	1800.23
96	200	20	0.06	0	1801.12	14.88	8298	1800.25
146	200	20	1.55	0	1801.68	18.73	6061	1800.35
196	200	20	25.84	0	1800.12	20.22	14977	1800.32
46	200	25	21.47	0	1800.27	21.29	6373	1801.43
96	200	25	0.55	0	1802.32	16.52	13001	1800.35
146	200	25	14.57	0	1800.14	19.44	5474	1800.53
196	200	25	25.39	0	1800.12	20.47	16405	1800.36

Table 4.3: Results for both Strategies on Data Set 2

		Strategy 1			Strategy 2		
n	m	Opt.Gap	Nodes	Time	Opt.Gap	Nodes	Time
50	15	0.00	13	52.20	0.00	11639	157.75
50	20	0.00	7	176.86	0.91	77208	1609.03
50	25	0.00	9	566.91	1.68	33551	1800.07
100	15	0.05	54	559.13	1.65	59471	1800.05
100	20	0.22	15	1505.00	2.89	23376	1800.04
100	25	0.26	3	1769.39	3.77	11401	1800.07
150	15	0.14	49	824.75	2.21	33290	1800.12
150	20	0.40	14	1788.04	3.32	15647	1800.08
200	15	0.23	83	1151.41	2.72	25652	1800.13

Table 4.4: Results of both Strategies for Larger Instances in Data Set 2

			Strategy 1			Strategy 2		
seed	n	m	Opt.Gap	Nodes	Time	Opt.Gap	Nodes	Time
46	150	25	0.90	0	1800.46	4.18	4407	1800.07
96	150	25	0.06	0	1800.54	3.79	11631	1800.07
146	150	25	4.70	0	1800.25	4.35	4531	1800.05
196	150	25	23.56	0	1800.07	4.69	2800	1800.07
46	200	20	1.17	0	1801.13	3.31	9226	1800.08
96	200	20	0.06	0	1801.12	3.18	12435	1800.04
146	200	20	1.55	0	1801.68	3.33	8500	1800.66
196	200	20	25.84	0	1800.12	4.92	6200	1800.06
46	200	25	21.47	0	1800.27	4.85	1764	1800.09
96	200	25	0.55	0	1802.32	4.14	5800	1800.10
146	200	25	14.57	0	1800.14	4.57	1241	1800.10
196	200	25	25.39	0	1800.12	5.28	107	1800.12

Table 4.5: Detailed Results for the Branch and Cut Algorithm with Strategy 1

Data Set	n	m	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
1	50	15	0.00	25.34	0.02	196035	3	16.84	8.45
	50	20	0.00	25.72	0.05	636394	5	281.21	212.20
	50	25	0.00	25.95	0.04	1061969	3	191.47	147.07
	100	15	0.02	25.45	0.06	506316	20	511.46	361.99
	100	20	0.01	25.83	0.06	1308924	17	698.99	541.76
	100	25	0.00	26.03	0.04	2455803	13	1114.08	889.67
	150	15	0.00	25.53	0.07	800217	47	417.43	298.83
	150	20	0.00	25.91	0.06	1783133	30	693.37	471.78
	150	25	0.01	26.10	0.03	3706397	12	1534.34	1108.60
	200	15	0.00	25.47	0.05	931479	75	370.32	265.03
	200	20	0.02	25.86	0.05	2432627	26	1122.41	755.04
	200	25	0.23	26.19	0.23	4668878	0	1801.37	1232.61
2	50	15	0.00	33.77	0.11	305966	13	52.20	19.60
	50	20	0.00	34.35	0.15	942863	7	176.86	78.56
	50	25	0.00	34.66	0.10	1960752	9	566.91	258.37
	100	15	0.05	34.04	0.19	752233	54	559.13	133.23
	100	20	0.22	34.74	0.30	2754591	15	1505.00	664.16
	100	25	0.26	35.04	0.27	4827527	3	1769.39	926.23
	150	15	0.14	34.20	0.27	1596305	49	824.75	321.20
	150	20	0.40	35.02	0.48	3088022	14	1788.04	845.33
	200	15	0.23	34.19	0.34	1941669	83	1151.41	551.24

Table 4.6: Detailed Results of Larger Instances in Data Set 2 for both Strategies

Strat.	seed	n	m	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
1	46	150	25	0.90	36.21	0.95	4764026	0	1800.46	683.84
	96	150	25	0.06	38.03	0.07	4917225	0	1800.54	1207.96
	146	150	25	4.70	39.07	4.78	5288963	0	1800.25	579.45
	196	150	25	23.56	46.63	23.58	5148694	0	1800.07	472.02
	46	200	20	1.17	36.40	1.19	3501692	0	1801.13	740.23
	96	200	20	0.06	37.38	0.06	3375790	0	1801.12	1244.58
	146	200	20	1.55	36.49	1.54	4416280	0	1801.68	727.49
	196	200	20	25.84	48.15	25.88	4427869	0	1800.12	352.16
	46	200	25	21.47	48.96	21.47	4484525	0	1800.27	474.30
	96	200	25	0.55	37.96	0.51	6430862	0	1802.32	997.74
	146	200	25	14.57	44.26	14.51	4398448	0	1800.14	437.01
	196	200	25	25.39	46.69	25.39	4451876	0	1800.12	385.84
2	46	150	25	4.18	35.81	4.47	2867921	4407	1800.07	297.10
	96	150	25	3.79	38.05	4.29	2771757	11631	1800.07	298.91
	146	150	25	4.35	36.46	4.59	2633518	4531	1800.05	255.39
	196	150	25	4.69	30.79	4.89	2753396	2800	1800.07	240.78
	46	200	20	3.31	35.89	3.50	1515766	9226	1800.08	100.16
	96	200	20	3.18	37.38	3.40	1620132	12435	1800.04	100.98
	146	200	20	3.33	35.77	3.55	1562505	8500	1800.66	100.09
	196	200	20	4.92	30.77	5.12	1639543	6200	1800.06	96.45
	46	200	25	4.85	36.52	5.17	3761460	1764	1800.09	399.61
	96	200	25	4.14	37.75	4.37	3698513	5800	1800.10	427.63
	146	200	25	4.57	36.35	4.86	3706038	1241	1800.10	387.17
	196	200	25	5.28	31.29	5.44	3855334	107	1800.12	356.68

Chapter 5

Lot Sizing Problem for a Perishable Item with Composition Constraints

In this part of the study, we introduce additional constraints to our problem. In the earlier chapters, we assume that the supplier has infinite number of products from each product age. However, as in the case of blood products [6], the supplier may have limited available products within certain age groups. Alternatively, product quality may decrease over time before the product expires. In order to have a certain level of quality while minimizing the total cost, the inventory manager may want to have products from different age groups in the inventory.

5.1 Composition Constraints

Motivated by the above examples, we divide products into different age groups and assume that each group has five different product ages. For instance, products that are younger than 6 periods are grouped into the first category, products that are 6 to 10 periods old are grouped into the second category and so on. b is the number of groups, i_j is the smallest element of group $j \in [1, b]$ and C_{jt} is the maximum proportion of age group j in the order at period t. The composition constraints are as follows:

$$\sum_{i=i_j}^{i_{j+1}-1} x_{it} \leq C_{jt} \sum_{i=1}^{m} x_{it} \qquad j \in [1, b], t \in [1, n] \tag{5.1}$$

Constraints (5.1) make sure that the ratio of each age group in an order does not exceed the allowed ratio.

5.2 Computational Study

In this section, we test the effectiveness of the valid inequalities on the problem with composition constraints. We use the data set 2 from Chapter 4 and assume that the maximum proportions are the same for all age groups and are time-invariant. Table 5.1 shows the average results of three instances with data set 2 and maximum proportion of 50%. Table 5.2 shows the average results for the same data set with maximum proportion of 30%.

We compare the optimality gaps and branch and cut nodes for the algorithm and the default Cplex. Results show that optimality gaps are improved and there are fewer number of branch and cut nodes with the algorithm. Hence the valid inequalities are effective for the model with composition constraints, too.

Similar to Chapter 4, in some of the instances there are huge optimality gaps. We apply the 5 iteration rule (strategy 2) to these instances to prevent tailing-off. Table 5.3 shows the results of both strategies. Although, it has more number of branch and cut nodes, 5 iteration rule improves the optimality gaps for these instances and it yields better results than the default Cplex.

Tables 5.4 and 5.5 show the detailed results of the branch and cut algorithm.

Although there are default preprocessing and Cplex cuts, lower bounds are significantly improved on the root node with the valid inequalities. Table 5.6 shows the details of the algorithm for the larger instances with $C = 0.5$ for both strategies. As expected, time spent on separation is shorter on the results with strategy 2. Also, there are fewer cuts added and more branch and cut nodes. Strategy 2 results have better gaps then the results of strategy 1.

Table 5.1: Branch and Cut Results for $C = 0.5$

		Branch and Cut Algorithm			Cplex		
n	m	Opt. Gap	Nodes	Time	Opt. Gap	Nodes	Time
50	15	0.00	5	40.42	12.28	7192314	1821.95
50	20	0.05	1	1286.49	15.02	8057803	1861.78
50	25	0.38	0	1800.39	14.78	5719407	1827.30
100	15	0.00	8	128.79	14.39	89519	1800.43
100	20	0.22	6	1705.55	17.22	46265	1800.18
100	25	3.55	0	1800.44	18.85	24194	1800.33
150	15	0.00	17	297.32	16.41	33538	1801.08
150	20	1.69	0	1800.87	19.07	13053	1800.28
200	15	0.03	21	1155.20	18.12	11854	1800.64
200	20	2.95	0	1800.76	19.64	6430	1800.68

Table 5.2: Branch and Cut Results for $C = 0.3$

		Branch and Cut Algorithm			Cplex		
n	m	Opt.Gap	Nodes	Time	Opt.Gap	Nodes	Time
50	20	0.57	0	1800.14	14.00	6714230	1834.99
50	25	0.52	0	1800.22	15.90	5900033	1818.12
100	20	2.85	0	1800.11	17.36	35835	1800.57
100	25	3.71	0	1800.10	20.37	23306	1800.25
150	20	2.16	0	1800.35	19.96	11442	1800.35
150	25	3.67	0	1800.97	23.05	10546	1800.32
200	20	3.70	0	1800.20	22.01	7195	1800.69
200	25	5.52	0	1800.24	24.51	5882	1800.46

Table 5.3: Branch and Cut Results of Larger Instances for $C = 0.5$

				Branch and Cut			Cplex		
Stra.	seed	n	m	Opt. Gap	Nodes	Time	Opt. Gap	Nodes	Time
1	46	150	25	20.20	0	1800.09	22.11	9856	1801.16
	96	150	25	0.68	0	1800.26	18.64	12613	1800.34
	146	150	25	8.98	0	1800.08	22.32	11312	1800.27
	46	200	25	21.98	0	1800.17	23.34	5289	1800.39
	96	200	25	3.90	0	1800.72	19.61	6838	1800.99
	146	200	25	22.50	0	1800.48	21.95	6418	1800.40
2	46	150	25	5.65	2910	1800.08	22.11	9856	1801.16
	96	150	25	5.29	5328	1800.23	18.64	12613	1800.34
	146	150	25	5.41	3092	1800.09	22.32	11312	1800.27
	46	200	25	5.89	74	1800.11	23.34	5289	1800.39
	96	200	25	5.72	3000	1800.11	19.61	6838	1800.99
	146	200	25	5.51	159	1800.12	21.95	6418	1800.40

Table 5.4: Detailed Results for Branch and Cut with $C = 0.5$

n	m	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
50	15	0.00	34.78	0.08	304157	5	40.42	19.03
50	20	0.05	35.83	0.06	2144740	1	1286.49	752.55
50	25	0.38	36.42	0.38	6888719	0	1800.39	1035.20
100	15	0.00	35.07	0.03	469178	8	128.79	61.78
100	20	0.22	36.22	0.26	3592111	6	1705.55	817.12
100	25	3.55	38.58	3.55	5965859	0	1800.44	787.74
150	15	0.00	35.11	0.07	776794	17	297.32	134.63
150	20	1.69	37.17	1.69	4820472	0	1800.87	774.87
200	15	0.03	34.99	0.09	1370315	21	1155.20	668.56
200	20	2.95	37.82	2.95	4317300	0	1800.76	645.80

Table 5.5: Detailed Results for Branch and Cut with $C = 0.3$

n	m	Opt. Gap	LP Gap	Root Gap	User Cuts	Nodes	Time	Sep. Time
50	20	0.57	35.62	0.57	11189751	0	1800.14	945.83
50	25	0.52	35.96	0.52	12354654	0	1800.22	997.97
100	20	2.85	37.36	2.85	8086733	0	1800.11	849.49
100	25	3.71	38.14	3.71	7947870	0	1800.10	850.83
150	20	2.16	36.86	2.16	6300402	0	1800.35	734.83
150	25	3.67	38.02	3.67	7247946	0	1800.97	841.78
200	20	3.70	37.66	3.70	5403976	0	1800.20	632.94
200	25	5.52	38.92	5.52	6095923	0	1800.24	845.67

Table 5.6: Detailed Results of Branch and Cut Algorithm for Larger Instances with Strategy 1 and 2. $C = 0.5$

Stra.	seed	n	m	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
1	46	150	25	20.20	47.99	20.20	3692975	0	1800.09	497.13
	96	150	25	0.68	38.09	0.68	6352428	0	1800.26	887.07
	146	150	25	8.98	41.17	8.98	4326068	0	1800.08	402.36
	46	200	25	21.98	48.21	21.98	3182651	0	1800.17	504.91
	96	200	25	3.90	39.60	3.90	5701297	0	1800.72	689.62
	146	200	25	22.50	48.73	22.50	3499123	0	1800.48	335.37
2	46	150	25	5.65	35.96	5.82	2067131	2910	1800.08	329.84
	96	150	25	5.29	37.98	5.72	1999468	5328	1800.23	207.53
	146	150	25	5.41	36.46	5.65	2148806	3092	1800.09	207.38
	46	200	25	5.89	36.53	6.14	2849248	74	1800.11	474.33
	96	200	25	5.72	37.87	5.94	2616718	3000	1800.11	280.35
	146	200	25	5.51	36.26	5.75	2955255	159	1800.12	293.66

Chapter 6

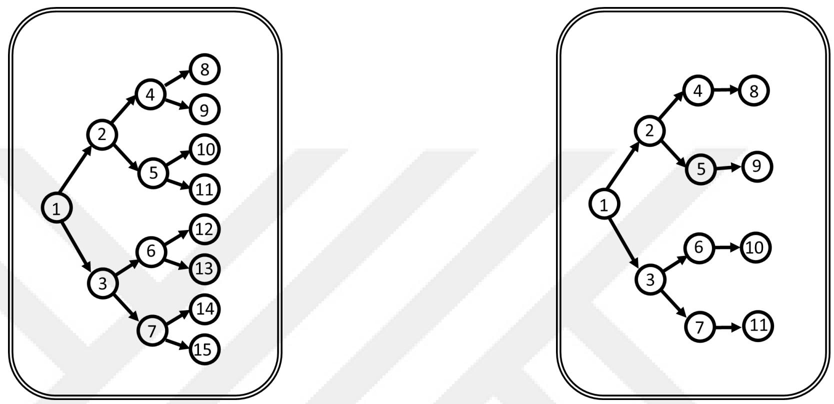
Multistage Stochastic Lot Sizing with Perishable Items

6.1 Problem Formulation

In this chapter, we consider the lot sizing with perishable items where demand is random. The objective of the problem is to find a procurement plan that minimizes the expected total cost. We model the problem as a multistage stochastic program where stages correspond to planning periods. We use the node representation to model the scenario tree. d_w denotes the demand of node w and p_w is the probability associated with node w. Decisions in each stage depend on the history of decisions as well as the realized demand values.

The fixed cost of procurement in node w is denoted by f_w and the unit cost for shortage in node w is denoted by g_w. The ordering and the inventory holding cost in node w for a product of age i are c_{iw} and h_{iw}, respectively.

In practice, the procurement plan of the earlier periods are more important because decisions of the later periods updated in a rolling horizon. In order to represent this notion in the model, we assume that scenario tree branches until period t' and for the remaining periods after t', we assume that we have deterministic demand values. Figure 6.1 shows problems with 4 stages. In Figure 6.1a, t' is also 4. So the scenario tree branches in all stages, whereas in Figure 6.1b $t' = 3$. So the demand values for the last stage are taken as expected values.



(a) Problem with 8 Scenarios (b) Problem with 4 Scenarios

Figure 6.1: Multistage Stochastic Problems with 4 Stages

The number of nodes in the scenario tree is denoted by ν. The direct predecessor of node w is represented by $a(w)$. The path from node k to node l is denoted by $P(k,l)$ and $d_{kl} = \sum_{u \in P(k,l)} d_u$ where $k, l \in [1, \nu]$. The corresponding stage of node w is denoted by $t(w)$. The set of all descendant of node w is represented by $D(w)$. We define the following decision variables:

x_{iw} : the amount of product of age i purchased in node w.

y_w : 1 if there is a setup in node w, 0 otherwise.

q_{iw} : the amount of product of age i used to satisfy the demand in node w.

s_{iw} : the amount of product of age i in inventory in node w.

r_w : the amount of demand lost (shortage) in node w.

With these decision variables, the stochastic problem can be formulated as follows:

$$\min \sum_{w=1}^{\nu} p_w(f_w y_w + g_w r_w + \sum_{i=1}^{m} c_{iw} x_{iw} + \sum_{i=1}^{m} h_{iw} s_{iw}) \tag{6.1}$$

$$\text{s.t. } r_w + \sum_{i=1}^{m} q_{iw} = d_w \qquad w \in [1, \nu], \tag{6.2}$$

$$s_{i-1,a(w)} + x_{iw} = q_{iw} + s_{iw} \qquad w \in [1, \nu], i \in [1, m], \tag{6.3}$$

$$x_{iw} \leq M y_w \qquad w \in [1, \nu], i \in [1, m], \tag{6.4}$$

$$s_{iw} = 0 \qquad i = 0, w \in [0, \nu] \text{ or } i \in [1, m], w = 0, \tag{6.5}$$

$$x_{iw}, q_{iw}, s_{iw} \geq 0, \qquad w \in [1, \nu], i \in [1, m], \tag{6.6}$$

$$r_w \geq 0, y_w \in \{0, 1\}, \qquad w \in [1, \nu]. \tag{6.7}$$

The objective (6.1) is the expected total cost. It is the summation of fixed and variable costs, inventory holding costs and shortage costs times the probabilities. Demand in node w is either satisfied with products at most at the age of m, or the demand is lost, due to the constraints (6.2). Constraints (6.3) are the inventory balance equations. Costraints (6.4) indicate that if there is procurement in period t then there is a setup in that period. Multiplier M corresponds to a large number to indicate that there is no capacity restriction. For node w, we take M as $\max_{l \in \bar{D}w,} \{d_{wl}\}$, where $\bar{D_w} = \{l \in D(w) : t(l) = t(w) + m - 1\}$. Constraints (6.5) indicate there is no initial inventory and there is no 0 period old stock. Constraints (6.6) and (6.7) are the range constraints.

Proposition 7. *Inequality* (4.14) *is valid for the stochastic problem.*

Proof. Each path from the root node to a leaf node represents a scenario for the stochastic problem. Some scenarios share common nodes in the scenario tree and therefore have the same decisions variables. Assume that we duplicate decision variables of those nodes for each scenario and relax the requirement that they need to have the same solutions. In other words, consider that we give decisions for each scenario independently, then each scenario corresponds to a deterministic problem. By Theorem 1, (4.14) is valid for each separate scenario and therefore for the relaxed problem. □

6.2 Computational Study

We implement our branch and cut algorithm for the stochastic problem. We generate different instances with three different seeds. The problem parameters are similar to data set 1 in the earlier chapters. We assume that demands of different nodes are independent and have a uniform distribution in between $(10, 100)$. There are two possible outcomes for each node and the probability of each of the outcomes is the same.

We stop generating cuts after 5 iterations with no improvement to prevent tailing-off. We first consider the problem where scenario tree branches in each stage. Table 6.1 shows the average results of those problems with branch and cut algorithm and default Cplex. The inequalities are not as effective as they are for the deterministic problem. However, they still yield better optimality gaps than Cplex with default settings solution in the majority of the instances. Afterwards, we consider problems with different t' values. Table 6.2 shows the results of those problems. Branch and cut algorithm again yields better optimality gaps than Cplex with default settings solutions but it could not reach optimality in the given time.

Table 6.3 shows the detailed results of branch and cut algorithm for $t' = n$ and Table 6.4 shows the detailed results for other problem instances. Although there is default cuts and preprocessing of Cplex, difference between the LP gap and the root gap in both tables indicate that the inequalities improve the lower bound at the root node. However, there are still too many branch and cut nodes. Although separation does not take too much time for the small instances, it takes approximately half of the solution time for the large instances. These results show that inequalities are effective in solving the stochastic problem but there is still room for strengthening the valid inequalities for the stochastic problem.

Table 6.1: Results for the Stochastic Problem with $t' = n$

		Branch and Cut			Cplex		
n	m	Opt. Gap	Nodes	Time	Opt. Gap	Nodes	Time
9	4	1.27	663502	1800.20	0.71	1575791	1820.52
9	6	1.53	415461	1800.47	2.92	279912	1800.38
10	4	1.67	300116	1800.65	1.53	506634	1805.94
10	6	2.19	165505	1800.63	5.77	35289	1800.88
11	4	2.63	123246	1800.56	3.17	135873	1803.92
11	6	2.69	69967	1800.48	10.28	9938	1802.64
12	4	2.86	49453	1800.51	5.38	28712	1803.53
12	6	3.73	19518	1800.56	14.53	3651	1801.81
13	4	4.02	17100	1805.20	8.89	6733	1802.09
13	6	9.76	4905	1848.61	17.69	2455	1802.15

Table 6.2: Results for the Stochastic Problem

			Branch and Cut			Cplex		
n	t'	m	Opt. Gap	Nodes	Time	Opt. Gap	Nodes	Time
15	7	6	0.36	208549	1800.36	4.03	122045	1802.41
15	7	9	1.75	47840	1800.33	7.89	15162	1800.92
15	8	6	0.70	78400	1800.39	6.59	41541	1803.64
15	8	9	2.69	17811	1800.65	9.90	7191	1803.63
15	9	6	1.44	50150	1800.78	9.96	7906	1805.36
15	9	9	3.75	11953	1800.98	13.00	5468	1830.03
20	7	6	0.40	111039	1800.40	4.66	78511	1802.42
20	7	9	2.20	28483	1800.89	8.47	5531	1801.91
20	8	6	0.92	56214	1800.68	6.87	24452	1800.88
20	8	9	2.64	9819	1800.86	11.17	5522	1826.53
20	9	6	1.36	28619	1800.72	13.13	5540	1805.99
20	9	9	9.68	1007	1813.65	17.54	4156	1801.73

Table 6.3: Results of Branch and Cut Algorithm in detail for the Stochastic Problem where $t' = n$

n	m	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
9	4	1.27	25.06	2.57	25477	663502	1800.20	5.02
9	6	1.53	26.15	2.43	68749	415461	1800.47	10.10
10	4	1.67	25.28	2.59	54512	300116	1800.65	21.24
10	6	2.19	26.37	2.87	150362	165505	1800.63	30.56
11	4	2.63	25.59	2.98	111170	123246	1800.56	59.22
11	6	2.69	26.59	2.94	186173	69967	1800.48	70.69
12	4	2.86	25.91	3.35	158739	49453	1800.51	229.71
12	6	3.73	27.34	4.04	406155	19518	1800.56	271.18
13	4	4.02	26.54	4.34	440828	17100	1805.20	826.83
13	6	9.76	31.76	9.83	1298551	4905	1848.61	1166.26

Table 6.4: Results of Branch and Cut Algorithm in detail for the Stochastic Problem

n	t'	m	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
15	7	6	0.36	16.93	1.64	48673	208549	1800.36	8.06
15	7	9	1.75	16.63	2.60	105048	47840	1800.33	20.88
15	8	6	0.70	16.64	1.64	96424	78400	1800.39	31.91
15	8	9	2.69	16.92	2.74	199847	17811	1800.65	69.65
15	9	6	1.44	17.98	2.06	100967	50150	1800.78	56.04
15	9	9	3.75	17.92	3.93	265660	11953	1800.98	154.57
20	7	6	0.40	31.20	0.98	72636	111039	1800.40	20.20
20	7	9	2.20	31.26	2.53	171703	28483	1800.89	51.94
20	8	6	0.92	31.48	1.52	75023	56214	1800.68	44.22
20	8	9	2.64	32.04	2.86	365225	9819	1800.86	219.10
20	9	6	1.36	32.00	1.88	158777	28619	1800.72	195.51
20	9	9	9.68	36.47	9.84	702621	1007	1813.65	693.83

Chapter 7

Decomposition of the Stochastic Lot Sizing Problem

In this chapter, we model the problem of multistage stochastic lot sizing with perishable items by using scenario representation and the so-called nonanticipativity constraints. Some scenarios share common nodes and therefore common decision variables. In scenario representation, those common decision variables are duplicated for each scenario. Nonanticipativity constraints make sure that all the duplicates have the same value.

Parameters of the problem are as follows:

ϑ : Number of scenarios.

p_w : Probability of realization of scenario w.

d_{wt} : Demand in period t under scenario w.

Z_w^t : The set of scenarios having the same history as scenario w at time t.

The multistage stochastic problem can be modeled as follows:

$$\min \sum_{w=1}^{\vartheta} p_w (\sum_{t=1}^{n} g_t r_{wt} + f_t y_{wt} + \sum_{t=1}^{n}\sum_{i=1}^{m}(c_{it}x_{wit} + h_{it}s_{wit})) \tag{7.1}$$

$$\text{s.t. } r_{wt} + \sum_{i=1}^{m} q_{wit} = d_{wt} \qquad t \in [1,n], w \in [1,\vartheta] \tag{7.2}$$

$$x_{wit} \leq M y_{wt} \qquad t \in [1,n], i \in [1,m], w \in [1,\vartheta] \tag{7.3}$$

$$s_{w,i-1,t-1} + x_{wit} = q_{wit} + s_{wit} \qquad t \in [1,n], i \in [1,m], w \in [1,\vartheta] \tag{7.4}$$

$$x_{wit} = x_{w'it} \qquad t \in [1,n], i \in [1,m], w \in [1,\vartheta], w' \in Z_w^t \tag{7.5}$$

$$r_{wt} = r_{w't} \qquad t \in [1,n], w \in [1,\vartheta], w' \in Z_w^t \tag{7.6}$$

$$y_{wt} = y_{w't} \qquad t \in [1,n], w \in [1,\vartheta], w' \in Z_w^t \tag{7.7}$$

$$s_{wit} = s_{w'it} \qquad t \in [1,n], i \in [1,m], w \in [1,\vartheta], w' \in Z_w^t \tag{7.8}$$

$$q_{wit} = q_{w'it} \qquad t \in [1,n], i \in [1,m], w \in [1,\vartheta], w' \in Z_w^t \tag{7.9}$$

$$s_{wit} = 0 \qquad i = 0, t \in [0,n] \text{ or } i \in [1,m], t = 0, w \in [1,\vartheta] \tag{7.10}$$

$$x_{wit}, q_{it}, s_{wit} \geq 0, \qquad t \in [1,n], i \in [1,m], w \in [1,\vartheta] \tag{7.11}$$

$$r_{wt} \geq 0, y_{wt} \in \{0,1\} \qquad t \in [1,n], w \in [1,\vartheta] \tag{7.12}$$

The objective (7.1) is the expected total cost. Constraints (7.2) - (7.4) and (7.10) - (7.12) are similar to constraints in the deterministic model. However in this model each constraint is defined for each scenario. Constraints (7.5)-(7.9) are nonanticipativity constraints. These constraints are useful when decomposing the problem into subproblems.

The number of scenarios in the problem grows exponentially as the number of planning horizon increases. Decomposing the problem into smaller and easier subproblems is one way to deal with large instances. Sandıkçı et al. [26] show that by scenario-wise decomposing the problem, it is possible to obtain bounds for the two stage stochastic problems. Sandıkçı and Özaltın [27] extend this study to multistage stochastic models. Let $S = [1, \vartheta]$ be the set of scenarios. They select a subset of scenarios $\Gamma \subset S$ which is referred to as block and define blockset G such that $\sum_{\Gamma \in G} = S$. A particular scenario can be included in more than one subset Γ in a blockset but probabilities must be updated so that objective function would be the expected value of the total cost function. They show that by solving the problems in a blockset, one can obtain bounds for the original problem.

Zenarosa et al. [28] also extend study of Sandıkçı et al. [26] to multistage stochastic program and divide the problem into smaller subproblems by separating the scenario tree. However, instead of separating different scenarios, they separate the root node from leaves.

In order to obtain bounds to our problem, we relax some of the nonanticipativity constraints and decompose our problem into smaller subproblems as it is stated in [27] and for simplicity we assume that every scenario is included only in one subproblem.

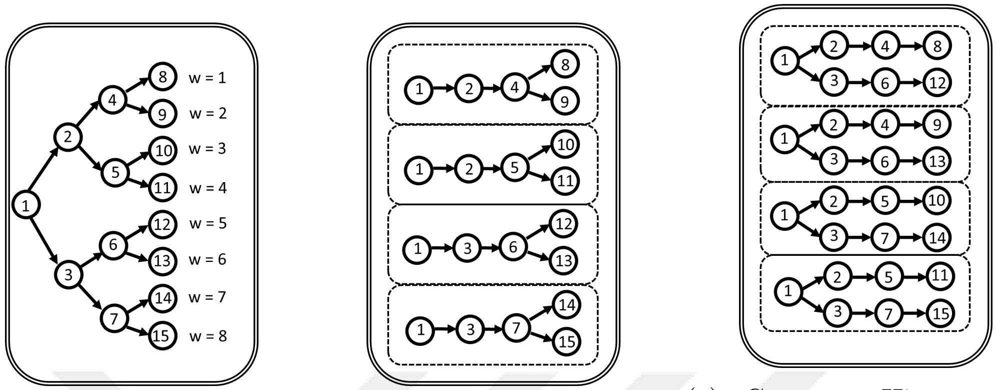
Figure 7.1a shows a problem with 8 scenarios and 4 periods of planning. Figure 7.1b and Figure 7.1c show possible decomposition options for the problem.

Sub-figure 7.1b shows the sequential approach where scenarios are included in a partition consecutively as they appear in the scenario tree. In this strategy, similar scenarios that share more common nodes in the scenario tree are grouped into the same subproblem.

Sub-figure 7.1c displays the common history approach where scenarios are chosen from different branches of the scenario tree. In this method, subproblems represent a broader spectrum of possible scenarios and might carry more information about the original problem. Both approaches have their advantages and disadvantages. The subproblems obtained with sequential approach are smaller and hence easier to solve, whereas common history approach might yield better bounds because it represents various demand realizations.

7.1 Obtaining Upper Bound

In this part, we use the method described in [27] to obtain good feasible solutions to the problem. Let λ be the number of subproblems and $\hat{x}_j$ be the optimal



(a) the Original Problem (b) Sequential Approach (c) Common History Approach

Figure 7.1: Problem with 8 scenarios and 4 stages and different scenario partitioning Methods

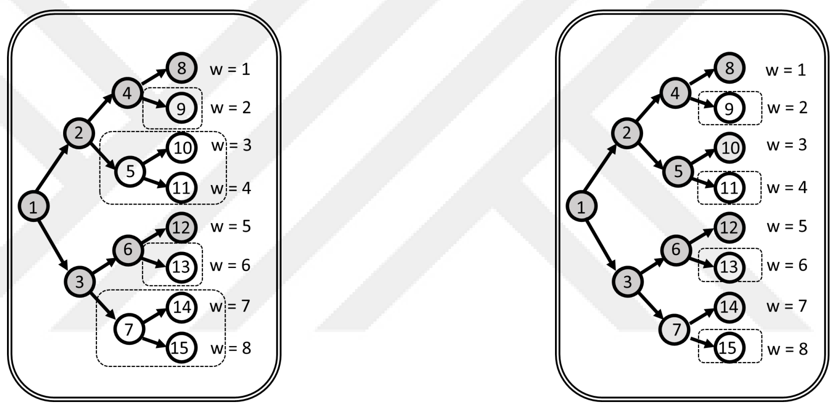
solution of a subproblem $j \in [1, \lambda]$. We substitute the solution $\hat{x}_j$ into the original problem. The remaining problem is decomposed into smaller residual problems. After solving these residual problems, we obtain a feasible solution to the original problem. Grey nodes in Figure 7.2 represent the subproblem j and rectangles with dashed edges show the remaining residual problems after substituting $\hat{x}_j$.

The quality of the upper bound depends on the partitioning method used to obtain subproblems. We obtain subproblem j with common history approach in this study. It is also possible to obtain it with the sequential approach. However, common history approach includes more diverse scenarios in the initial subproblem. Hence, we expect that the upper bound obtained with the sequential approach would not be better than the one found with common history approach.

It is also important to decide on which subproblem to use as a starting point. In a time limit, we try different initial subproblems j. Assume that the subproblem that includes the first scenario is subproblem 1, that includes the second scenario is subproblem 2 and so no. We start with the first subproblem and iterate until either we try all the possible initial subproblems or the limit is reached. We choose the minimum upper bound we obtain.

Another important factor that affects the quality of upper bounds is the size of the problem j. Figure 7.2a shows a subproblem j with two branches and Figure

7.2b shows a subproblem j with four branches. As the number of branches in the subproblem increases, the size of the problem also increases. This means that more decision variables will be fixed in the original problem. The lower the number of fixed decision variables, the better the obtained upper bound is. So it is better to have smaller subproblem j to obtain good upper bounds. However, for large problem instances, there should be enough scenarios in subproblem j so that after substituting its solution to the original problem, the residual subproblems are small enough to be solved.



(a) Initial Subproblem j with 2 Branches (b) Initial Subproblem j with 4 Branches

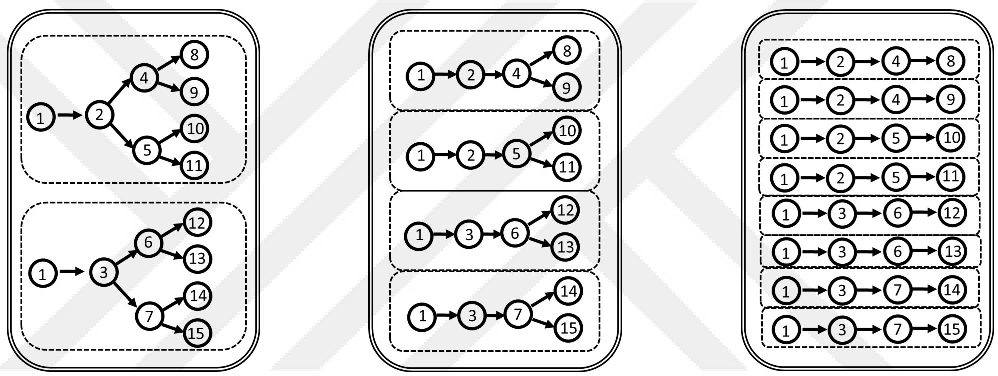
Figure 7.2: Obtaining Upper Bound with Common History approach

7.2 Obtaining Lower Bound

In this section, we obtain lower bounds to our problem as described by [27]. We divide the problem into several subproblems and solve each of the subproblems to obtain lower bounds. Let W_j be the set of scenarios in the subproblem $j \in [1, \lambda]$. The probability of a subproblem is the summation of the probabilities of scenarios included in that subproblem, i.e., $p(j) = \sum_{w \in W_j} p_w$ and the probability of a scenario included in the subproblem j is adjusted as follows, $\hat{p}_w = \frac{p_w}{p(j)}$.

Let z_j be the optimal value for subproblem $j \in [1, \lambda]$ and z^* be the optimal value of the original problem. Then $\sum_{j=1}^{\lambda} p(j) z_j \leq z^*$ holds by the results in [27].

Quality of the lower bounds depends on the partitioning strategy. We use the sequential approach for finding lower bounds because it provides good quality lower bounds in a short time. The size of the subproblems is important for finding good lower bounds. We partition the original scenario tree to different size subproblems as it can be seen in Figure 7.3. Quality of lower bounds is better when the size of the subproblems are larger. On the other hand, for large instances, size of the subproblem should be small enough to have small and easy subproblems.



(a) Subproblems with 4 Scenarios

(b) Subproblems with 2 Scenarios

(c) Subproblems with 1 Scenario

Figure 7.3: Finding lower bound: subproblems with different number of scenarios

7.3 Computational Study

In this section, we present computational results of the decomposition method. We create data sets similar to the ones in Chapter 6. We generate three instances for each n, m and the number of possible outcomes at each node (p.node).

We put time limit of 15 minutes for the upper bound part. For some instances, finding lower bound takes too much time. In order to find both the upper and lower bounds in half an hour time limit, we allocate less time for finding upper bounds for those instances. In most of the problem instances, solver spents too much time on reducing small optimality gaps of the subproblems while finding upper bounds. This prevents us from exploring different initial subproblems and finding upper bounds in a short time. Hence, we put optimality gap limit of 10%

to subproblems for most of the problem instances. Stars(.*) on n values in Table 7.1 indicates that we solve the subproblems to optimality for those instances.

Table 7.1 shows the average results of the decomposition method and the default Cplex. In the majority of the instances, the decomposition method yields better optimality gaps than solving the problem without decomposing. For some instances, we obtain lower and upper bounds in less than half an hour. As expected, the problems become more challenging to solve when the number of possible outcomes at each node increases. Also increasing the shelf-life of products makes the problem more difficult to solve.

Table 7.2 shows the details of upper bound results. τ refers to the stage where the subproblem j stops branching on the scenario tree. For instance, for the problem in Figure 7.2a, τ is 2 whereas, for Figure 7.2b, τ is 3. So for higher τ values, more decision variables are fixed on the original problem. That is why optimality gaps are generally better for smaller τ values in the table.

As stated earlier, we try different initial subproblems j to obtain better upper bounds. Best-Worst refers to the gap between the best and worst upper bounds we find under the time limit. These gaps show how the initial choice of the problem effects the resulting upper bound. For some of the instances, Best-Worst values are zero. It means that in the limited time, algorithm can iterate only for one initial subproblem.

Table 7.3 shows the details of lower bound results. This time τ refers to the stage where the scenario tree starts to branch. For example, for Figure 7.3a, τ is 2 whereas, for Figure 7.3b, τ is 3. As τ increases, subproblems get smaller but the number of subproblems increases. In almost all instances, smaller τ values yield better optimality gaps. On the other hand, it generally takes more time to find a lower bound because the subproblems are larger. For τ values that are smaller than the ones on the table, it may take too much time to solve subproblems because of their sizes.

Table 7.1: Comparison of the Decomposition Method and the Default Cplex

			Decomposition		Cplex	
n	m	p.node	Opt. Gap	Time	Opt. Gap	Time
10*	4	2	2.16	1253.93	1.53	1805.94
10*	6	2	2.59	1185.05	5.77	1800.88
11	4	2	4.55	1425.88	3.17	1803.92
11	6	2	3.78	1373.27	10.28	1802.64
12	4	2	5.89	1400.35	5.38	1803.53
12	6	2	5.12	1944.94	14.53	1801.81
13	4	2	6.80	1948.26	8.89	1802.09
13	6	2	6.86	1725.51	17.69	1802.15
7	4	3	4.36	691.05	1.45	1801.73
8	4	3	4.99	1517.04	4.87	1801.31
8	6	3	4.58	1385.26	11.78	1805.26
9	4	3	6.69	1415.00	9.44	1808.20
9	6	3	6.02	2195.12	15.86	1802.14
7	4	4	5.43	1641.86	7.91	1801.75
8	4	4	7.48	1815.61	11.56	1802.62

Table 7.2: Detailed Results for Upper Bounds

n	m	p.node	τ	Best-Worst	Opt. Gap	UB Time
10	4	2	3	0.93	2.16	1020.45
10	4	2	4	1.66	2.41	929.08
10	6	2	4	1.51	2.59	990.04
10	6	2	5	3.70	3.32	577.52
11	4	2	4	3.84	4.55	920.47
11	4	2	5	3.17	4.86	471.77
11	6	2	5	2.68	4.09	947.60
11	6	2	6	3.35	3.78	972.54
12	4	2	7	3.41	5.89	999.53
12	4	2	8	1.99	6.69	1037.57
12	6	2	7	2.39	5.12	1046.11
12	6	2	8	1.26	5.70	1188.65
13	4	2	8	2.00	6.80	1093.07
13	4	2	9	1.66	7.03	1157.93
13	6	2	8	0.00	6.86	394.58
13	6	2	9	0.00	6.37	922.76
7	4	3	4	4.45	4.36	502.52
7	4	3	5	1.27	5.21	224.54
8	4	3	4	3.20	4.99	955.46
8	4	3	5	2.75	5.29	996.81
8	6	3	4	3.40	4.58	970.73
8	6	3	5	3.00	5.54	1036.83
9	4	3	4	2.50	6.69	1004.48
9	4	3	5	2.40	6.22	1083.39
9	6	3	4	1.85	6.30	634.01
9	6	3	5	1.64	6.02	726.23
7	4	4	4	3.65	5.43	1641.86
7	4	4	5	2.29	6.33	1784.54
8	4	4	4	2.47	7.48	1069.36
8	4	4	5	1.22	8.33	889.12

Table 7.3: Detailed Results for Lower Bound

n	m	p.node	τ	Opt. Gap	LB Time
10	4	2	5	2.16	233.48
10	4	2	6	2.58	86.19
10	6	2	6	2.59	195.02
10	6	2	7	3.01	133.02
11	4	2	6	4.55	505.41
11	4	2	7	4.90	186.27
11	6	2	7	3.78	400.73
11	6	2	8	4.16	271.17
12	4	2	8	5.89	400.82
12	4	2	9	6.25	372.54
12	6	2	8	5.12	898.83
12	6	2	9	5.44	598.04
13	4	2	9	6.80	855.19
13	4	2	10	7.13	795.40
13	6	2	9	5.68	1935.79
13	6	2	10	6.86	1330.93
7	4	3	4	4.36	182.06
7	4	3	6	5.83	54.33
8	4	3	5	4.99	561.58
8	4	3	6	5.92	245.98
8	6	3	6	4.58	414.53
8	6	3	7	6.21	417.96
9	4	3	7	6.69	410.53
9	4	3	8	6.79	954.73
9	6	3	7	6.02	1468.89
9	6	3	8	6.86	1862.53
7	4	4	5	5.43	625.34
7	4	4	6	6.26	357.19
8	4	4	7	7.48	746.25

Chapter 8

Conclusion

In this thesis, we study the uncapacitated lot sizing problem with perishable items. We first assume that demand is deterministic and evaluate different formulations for the problem. Computational results show that the facility location formulation is very effective in solving the basic problem. However, it fails to model age related constraints that can be faced in practice.

In order to be able to work on those extensions of our problem, we strengthen the LP relaxation of the natural formulation by proposing strong valid inequalities. We implement a branch and cut algorithm where we use the valid inequalities to solve both the basic problem and the one with composition constraints. We observe that the valid inequalities are very effective for both problems and branch and cut algorithm is capable of solving instances that cannot be solved with a commercial solver.

In the second part of the study, we take into account demand uncertainty and formulate the problem as a multistage stochastic program. We test the performance of the valid inequalities on the stochastic problem and present numerical results. Although the inequalities are not as strong as they are for the deterministic problem, computational results show that the branch and cut algorithm improves the optimality gaps for the stochastic problem as well.

Size of the stochastic problem grows exponentially as the number of planning horizon increases. In order to deal with large problems, we decompose the scenario tree into group subproblems and by solving these group subproblems we compute lower and upper bounds for the stochastic problems.

There can be several future research directions on this problem. It is possible to analyze the properties of the extreme points and come up with a dynamic programming algorithm or study cases where the valid inequalities are sufficient to describe the convex hull. The valid inequalities can be extended for the stochastic problem similar to work of Guan et al. [22] on extending the (l, S) inequalities for the stochastic problem.

In decomposing the stochastic problem, we assume that each scenario is added only to one subproblem. Sandıkçı and Özaltın [27] state that a scenario can be included in more than one subproblem. It can be another further research area to investigate the results of adding a scenario in multiple subproblems.

Lastly, we assume that products have fixed lifetime. The problem can be extended by considering random lifetime.

Bibliography

[1] F. W. Harris, "How many parts to make at once," *The Magazine of Management*, vol. 10, no. 2, pp. 135–136, 1913.

[2] R. Jans and Z. Degraeve, "Modeling industrial lot sizing problems: a review," *International Journal of Production Research*, vol. 46, no. 6, pp. 1619–1643, 2008.

[3] S. K. Goyal and B. C. Giri, "Recent trends in modeling of deteriorating inventory," *European Journal of Operational Research*, vol. 134, pp. 1–16, 2001.

[4] A. Dhamodharan and R. A. Proano, "Determining the optimal vaccine vial size in developing countries: a monte carlo simulation approach," *Health Care Management Science*, vol. 15, pp. 188–196, Sep 2012.

[5] "Blood components." `https://www.redcrossblood.org/donate-blood/how-to-donate/types-of-blood-donations/blood-components.html`. Accessed: 2019-07-10.

[6] S. Günpınar and G. Centeno, "Stochastic integer programming models for reducing wastages and shortages of blood products at hospitals," *Computers & Operations Research*, vol. 54, pp. 129 – 141, 2015.

[7] R. Haijema, J. van der Wal, and N. M. Dijk, "Blood platelet production: Optimization by dynamic programming and simulation," *Computers & Operations Research*, vol. 34, no. 3, pp. 760 – 779, 2007. Logistics of Health Care Management.

[8] H. M. Wagner and T. M. Whitin, "Dynamic version of the economic lot size model," *Management Science*, no. 12, p. 1770, 2004.

[9] J. Krarup and O. Bilde, *Plant location, Set Covering and Economic Lot Size: An 0 (mn)-Algorithm for Structured Problems*, vol. 36, pp. 155–180. 01 1977.

[10] I. Barany, T. Van Roy, and L. A. Wolsey, *Uncapacitated lot-sizing: The convex hull of solutions*, pp. 32–43. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984.

[11] Y. Pochet and L. Wolsey, "Polyhedra for lot-sizing with wagner-whitin costs," *Mathematical Programming*, vol. 67, pp. 297–323, 1994.

[12] Y. Pochet and L. A. Wolsey, *Production planning by mixed integer programming.* Springer Series in Operations Research and Financial Engineering., Springer, New York, 2006.

[13] N. Brahimi, N. Absi, S. Dauzere-Peres, and A. Nordli, "Single-item dynamic lot-sizing problems: An updated survey," *European Journal of Operational Research*, vol. 263, no. 3, pp. 838 – 863, 2017.

[14] S. Nahmias, "Perishable inventory theory: A review," *Mathematical Programming*, vol. 4, pp. 601–796, 1982.

[15] V. N. Hsu, "Dynamic economic lot size model with perishable inventory," *Management Science*, vol. 46, no. 8, pp. 1159–1169, 2000.

[16] F. Z. Sargut and G. Işık, "Dynamic economic lot size model with perishable inventory and capacity constraints," *Applied Mathematical Modelling*, vol. 48, pp. 806 – 820, 2017.

[17] M. Önal, H. Romeijn, A. Sapra, and W. van den Heuvel, "The economic lot-sizing problem with perishable items and consumption order preference," *European Journal of Operational Research*, vol. 244, no. 3, pp. 881 – 891, 2015.

[18] M. Önal, "The two-level economic lot sizing problem with perishable items," *Operations Research Letters*, vol. 44, no. 3, pp. 403 – 408, 2016.

[19] L. Janssen, T. Claus, and J. Sauer, "Literature review of deteriorating inventory models by key topics from 2012 to 2015," *International Journal of Production Economics*, vol. 182, pp. 86 – 112, 2016.

[20] S. Nahmias, *Perishable Inventory Systems. [electronic resource].* International Series in Operations Research & Management Science: 160, Springer US, 2011.

[21] M. A. Aloulou, A. Dolgui, and M. Y. Kovalyov, "A bibliography of non-deterministic lot-sizing models.," *International Journal of Production Research*, vol. 52, no. 8, pp. 2293 – 2310, 2014.

[22] Y. Guan, S. Ahmed, G. L. Nemhauser, and A. J. Miller, "A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem.," *Mathematical Programming*, vol. 105, no. 1, pp. 55 – 84, 2006.

[23] Y. Guan, S. Ahmed, A. J. Miller, and G. L. Nemhauser, "On formulations of the stochastic uncapacitated lot-sizing problem," *Operations Research Letters*, vol. 34, no. 3, pp. 241 – 250, 2006.

[24] D. M. Summa and L. A. Wolsey, "Lot-sizing on a tree," *Operations Research Letters*, vol. 36, no. 1, pp. 7 – 13, 2008.

[25] Z. Zhou and Y. Guan, "Stochastic lot-sizing problem with deterministic demands and wagner–whitin costs," *Operations Research Letters*, vol. 38, no. 5, pp. 414 – 419, 2010.

[26] B. Sandıkçı, N. Kong, and A. J. Schaefer, "A hierarchy of bounds for stochastic mixed-integer programs.," *Mathematical Programming*, no. 1-2, p. 253, 2013.

[27] B. Sandıkçı and O. Özaltin, "A scalable bounding method for multi-stage stochastic programs," *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1772–1800, 2017.

[28] G. Zenarosa, O. Prokoyev, and A. Schaefer, "A scenario tree decomposition: bounds for multistage stochastic mixed-integer programs," 2014.

Appendix A

Computational Results for the Deterministic Problem

Table A.1: Branch and Cut Results for Data Set 1 with $n = 50$

n	m	seed	IP	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
50	15	46	5481.70	0.00	25.99	0.03	187693	7	20.23	10.69
50	15	96	5231.05	0.00	28.97	0.00	187549	0	11.88	6.56
50	15	146	4398.85	0.00	24.42	0.03	207169	3	16.91	8.45
50	15	196	4920.50	0.00	21.96	0.00	201728	0	18.35	8.10
50	20	46	5475.30	0.00	26.40	0.06	512430	8	63.79	45.93
50	20	96	5229.75	0.00	29.43	0.00	459108	0	55.76	41.24
50	20	146	4395.55	0.00	24.86	0.10	1040080	7	928.68	716.58
50	20	196	4898.40	0.00	22.19	0.02	533956	6	76.59	45.06
50	25	46	5474.00	0.00	26.65	0.00	1038120	0	155.85	119.39
50	25	96	5229.10	0.00	29.67	0.07	1005365	6	200.49	158.89
50	25	146	4394.40	0.00	25.10	0.08	1025861	7	225.33	185.42
50	25	196	4890.40	0.00	22.36	0.00	1178531	0	184.22	124.57

Table A.2: Branch and Cut Results for Data Set 1

n	m	seed	IP	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
100	15	46	10983.05	0.00	26.02	0.04	448316	51	118.31	83.24
100	15	96	10110.75	0.00	29.26	0.07	401972	15	59.00	42.09
100	15	146	8881.95	0.08	24.53	0.08	648024	0	1800.10	1289.74
100	15	196	10004.20	0.00	22.00	0.04	526953	13	68.45	32.88
100	20	46	10970.10	0.00	26.43	0.08	1125124	29	283.06	203.15
100	20	96	10107.75	0.06	29.73	0.05	1741373	0	1800.58	1469.26
100	20	146	8874.05	0.00	24.94	0.07	1160605	28	404.21	324.92
100	20	196	9957.20	0.00	22.21	0.04	1208594	12	308.11	169.71
100	25	46	10968.00	0.00	26.66	0.00	2416047	0	981.45	772.38
100	25	96	10107.10	0.00	29.97	0.04	2515945	15	1678.84	1408.58
100	25	146	8871.20	0.00	25.15	0.11	2449344	33	1203.73	1017.32
100	25	196	9939.10	0.00	22.36	0.02	2441876	5	592.30	360.42
150	15	46	16488.45	0.00	26.00	0.11	675686	74	297.92	231.08
150	15	96	14847.95	0.00	29.42	0.07	642305	59	164.47	124.37
150	15	146	13496.95	0.00	24.37	0.06	1016922	40	1009.21	741.29
150	15	196	14451.80	0.00	22.33	0.06	865955	15	198.12	98.56
150	20	46	16463.70	0.00	26.38	0.07	1845086	34	871.84	650.22
150	20	96	14839.40	0.00	29.88	0.03	1552905	15	388.64	299.78
150	20	146	13487.25	0.00	24.78	0.09	1570073	42	579.26	473.37
150	20	196	14386.65	0.00	22.58	0.06	2164468	29	933.75	463.76
150	25	46	16459.40	0.00	26.60	0.03	3680506	18	1621.71	1271.18
150	25	96	14838.00	0.00	30.11	0.02	3443222	3	1014.33	809.14
150	25	146	13484.40	0.05	24.99	0.05	3275829	23	1800.92	1565.37
150	25	196	14356.90	0.00	22.71	0.03	4426029	3	1700.40	788.69
200	15	46	21583.80	0.00	26.27	0.10	920453	189	591.08	471.55
200	15	96	20498.25	0.00	29.02	0.02	826610	43	251.02	187.15
200	15	146	18290.55	0.00	24.18	0.05	831645	36	240.74	181.73
200	15	196	19155.75	0.00	22.42	0.05	1147208	30	398.45	219.68
200	20	46	21554.30	0.00	26.67	0.07	2360092	64	1462.95	1180.37
200	20	96	20486.50	0.00	29.47	0.01	2132647	6	726.63	562.35
200	20	146	18276.45	0.00	24.58	0.03	2073349	18	498.87	371.60
200	20	196	19077.55	0.09	22.71	0.08	3164418	15	1801.18	905.84
200	25	46	21550.80	0.03	26.90	0.05	4808491	0	1801.86	1313.81
200	25	96	20489.70	0.04	29.72	0.05	4499832	0	1801.36	1469.19
200	25	146	18273.60	0.04	24.78	0.04	4284869	0	1802.04	1410.74
200	25	196	19171.40	0.79	23.37	0.81	5082320	0	1800.23	736.69

Table A.3: Branch and Cut Results for Data Set 2

n	m	seed	IP	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
50	15	46	6307.30	0.00	34.39	0.07	275857	10	40.9	19.78
50	15	96	5939.05	0.00	36.53	0.05	215521	7	24.0	12.90
50	15	146	5221.55	0.00	34.74	0.02	261402	0	27.1	11.48
50	15	196	5552.70	0.00	29.44	0.32	471084	36	116.7	34.23
50	20	46	6279.60	0.00	34.94	0.09	701039	7	100.2	52.50
50	20	96	5931.05	0.00	37.15	0.01	749398	0	81.1	46.53
50	20	146	5203.45	0.00	35.41	0.18	916033	6	191.5	114.01
50	20	196	5514.20	0.00	29.89	0.33	1404983	13	334.5	101.18
50	25	46	6275.40	0.00	35.33	0.03	1690686	3	380.8	257.87
50	25	96	5927.40	0.00	37.47	0.03	1611776	5	248.1	173.22
50	25	146	5195.90	0.00	35.79	0.08	2276721	11	556.5	314.70
50	25	196	5490.10	0.00	30.06	0.27	2263826	16	1082.0	287.71
100	15	46	12685.80	0.00	34.75	0.04	640347	10	146.0	69.33
100	15	96	11475.65	0.00	36.81	0.14	426445	35	125.0	87.11
100	15	146	10580.55	0.00	35.19	0.09	759131	24	165.2	82.57
100	15	196	11261.60	0.20	29.40	0.49	1183007	148	1800.2	293.92
100	20	46	12680.20	0.41	35.60	0.41	3807745	0	1800.0	988.88
100	20	96	11455.35	0.00	37.45	0.01	2579674	3	984.2	603.61
100	20	146	10542.75	0.00	35.89	0.22	1973994	42	1435.6	639.25
100	20	196	11201.25	0.48	30.02	0.57	2656949	16	1800.1	424.90
100	25	46	12628.45	0.00	35.76	0.06	4345646	9	1675.9	833.92
100	25	96	11450.60	0.04	37.79	0.05	3987200	0	1801.1	1406.79
100	25	146	10540.85	0.43	36.33	0.42	6127852	1	1800.3	809.46
100	25	196	11169.05	0.59	30.30	0.57	4849408	1	1800.0	654.76
150	15	46	19031.50	0.00	34.72	0.10	954475	28	436.8	235.61
150	15	96	16866.70	0.00	37.02	0.07	660425	40	222.8	148.84
150	15	146	16091.25	0.00	35.18	0.28	1222614	92	839.2	422.78
150	15	196	16312.80	0.57	29.87	0.63	3547707	37	1800.1	477.58
150	20	46	18960.15	0.00	35.36	0.13	2886381	26	1750.1	872.43
150	20	96	16838.15	0.10	37.68	0.10	2615297	0	1800.9	1320.04
150	20	146	16043.80	0.10	35.91	0.22	2713953	25	1800.4	701.61
150	20	196	16359.10	1.39	31.11	1.47	4136455	3	1800.7	487.24
150	25	46	19087.40	0.90	36.21	0.95	4764026	0	1800.4	683.84
150	25	96	16831.90	0.06	38.03	0.07	4917225	0	1800.5	1207.96
150	25	146	16757.05	4.70	39.07	4.78	5288963	0	1800.2	579.45
150	25	196	20964.30	23.56	46.63	23.58	5148694	0	1800	472.02

Table A.4: Branch and Cut Results for Data Set 2 with $n = 200$

n	m	seed	IP	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
200	15	46	24936.25	0.00	35.02	0.13	1313077	64	836.79	436.29
200	15	96	23296.25	0.00	36.71	0.12	1150901	141	930.34	742.73
200	15	146	21755.40	0.00	34.91	0.17	1427006	111	1038.43	561.48
200	15	196	21674.00	0.90	30.11	0.95	3875690	17	1800.10	464.47
200	20	46	25125.35	1.17	36.40	1.19	3501692	0	1801.13	740.23
200	20	96	23260.85	0.06	37.38	0.06	3375790	0	1801.12	1244.58
200	20	146	21983.05	1.55	36.49	1.54	4416280	0	1801.68	727.49
200	20	196	28772.10	25.84	48.15	25.88	4427869	0	1800.12	352.16
200	25	46	31098.95	21.47	48.96	21.47	4484525	0	1800.27	474.30
200	25	96	23343.3	0.55	37.96	0.51	6430862	0	1802.3	997.74
200	25	146	24875.40	14.57	44.26	14.51	4398448	0	1800.14	437.01
200	25	196	27776.75	25.39	46.69	25.39	4451876	0	1800.12	385.84

Table A.5: Branch and Cut Results with Strategy 2 for Data Set 2 with $n = 50$

n	m	seed	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
50	15	46	0.00	34.39	1.68	150923	6281	105.38	8.85
50	15	96	0.00	36.53	1.62	151110	8754	71.44	4.86
50	15	146	0.00	34.74	1.85	162388	2268	65.20	4.22
50	15	196	0.00	29.44	2.05	181278	29251	388.97	4.46
50	20	46	0.00	34.94	3.15	350481	59057	1783.17	17.19
50	20	96	0.67	37.15	3.48	405423	145725	1800.03	17.98
50	20	146	0.00	35.41	3.18	341982	34748	1052.77	16.83
50	20	196	2.99	29.89	5.79	278989	69300	1800.13	15.56
50	25	46	1.41	35.33	3.54	801446	26130	1800.04	66.21
50	25	96	1.75	37.47	4.28	795533	49800	1800.09	68.06
50	25	146	1.63	35.79	3.77	639131	22100	1800.12	54.56
50	25	196	1.91	30.06	4.05	688755	36174	1800.02	53.15

Table A.6: Branch and Cut Results with Strategy 2 for Data Set 2

n	m	seed	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
100	15	46	1.66	34.75	2.82	314489	49701	1800.10	9.95
100	15	96	0.98	36.81	2.07	292964	95290	1800.04	9.64
100	15	146	1.33	35.19	2.13	387583	46016	1800.04	11.74
100	15	196	2.63	29.39	3.35	349170	46878	1800.04	9.93
100	20	46	2.43	35.38	3.06	715665	23600	1800.05	40.86
100	20	96	2.21	37.45	3.16	782488	37250	1800.04	43.15
100	20	146	3.14	35.93	3.97	734946	15600	1800.04	41.71
100	20	196	3.78	29.99	4.38	783703	17055	1800.03	41.74
100	25	46	3.83	36.07	4.30	1905522	8642	1800.05	181.03
100	25	96	3.36	37.79	4.21	1622907	22400	1800.10	176.28
100	25	146	3.77	36.32	4.44	1720035	8663	1800.06	140.23
100	25	196	4.14	30.48	4.58	1912021	5900	1800.08	142.91
150	15	46	2.19	34.76	2.80	486021	25300	1800.06	16.74
150	15	96	0.75	37.02	1.24	481703	52000	1800.08	17.92
150	15	146	2.43	35.21	3.03	520603	27200	1800.31	17.45
150	15	196	3.46	29.82	3.77	525599	28659	1800.04	15.91
150	20	46	2.97	35.43	3.27	1103670	13932	1800.13	68.55
150	20	96	2.74	37.68	3.31	1187360	28773	1800.07	70.66
150	20	146	2.90	35.92	3.21	1179503	11750	1800.06	72.01
150	20	196	4.67	30.59	4.93	1219798	8133	1800.08	68.09
150	25	46	4.18	35.81	4.47	2867921	4407	1800.07	297.10
150	25	96	3.79	38.05	4.29	2771757	11631	1800.07	298.91
150	25	146	4.35	36.46	4.59	2633518	4531	1800.05	255.39
150	25	196	4.69	30.79	4.89	2753396	2800	1800.07	240.78
200	15	46	2.51	35.09	2.76	694327	19600	1800.07	25.98
200	15	96	1.77	36.73	2.09	609645	41144	1800.06	24.82
200	15	146	2.31	34.99	2.70	704263	23100	1800.33	25.77
200	15	196	4.29	30.04	4.53	706354	18764	1800.07	23.12
200	20	46	3.31	35.89	3.50	1515766	9226	1800.08	100.16
200	20	96	3.18	37.38	3.40	1620132	12435	1800.04	100.98
200	20	146	3.33	35.77	3.55	1562505	8500	1800.66	100.09
200	20	196	4.92	30.77	5.12	1639543	6200	1800.06	96.45
200	25	46	4.85	36.52	5.17	3761460	1764	1800.09	399.61
200	25	96	4.14	37.75	4.37	3698513	5800	1800.10	427.63
200	25	146	4.57	36.35	4.86	3706038	1241	1800.10	387.17
200	25	196	5.28	31.29	5.44	3855334	107	1800.12	356.68

Table A.7: Branch and Cut Results for the Problem with Composition Constraints for $C = 0.5$

n	m	seed	IP	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
50	15	46	6355.47	0.00	33.90	0.14	209374	8	35.63	16.84
50	15	96	6003.13	0.00	36.11	0.07	174580	3	29.12	13.34
50	15	146	5262.00	0.00	34.34	0.03	528516	3	56.52	26.91
50	20	46	6354.28	0.11	34.96	0.11	2701509	0	1800.15	1053.60
50	20	96	6003.13	0.04	37.11	0.04	2081619	0	1800.11	1074.64
50	20	146	5261.20	0.00	35.43	0.04	1651092	3	259.22	129.41
50	25	46	6354.28	0.08	35.43	0.08	7915609	0	1800.26	1021.25
50	25	96	6004.03	0.29	37.52	0.29	5424689	0	1800.36	1098.60
50	25	146	5291.40	0.77	36.31	0.77	7325858	0	1800.55	985.73
100	15	46	12789.72	0.00	34.22	0.00	457915	5	103.86	36.85
100	15	96	11596.72	0.00	36.31	0.00	370799	2	86.72	41.15
100	15	146	10656.95	0.00	34.68	0.09	578819	16	195.80	107.34
100	20	46	12811.93	0.21	35.43	0.26	4073727	3	1800.51	955.74
100	20	96	11596.72	0.00	37.35	0.05	3589135	10	1516.04	910.22
100	20	146	10668.03	0.45	35.86	0.47	3113471	6	1800.09	585.39
100	25	46	13183.65	3.43	37.67	3.43	5566555	0	1800.55	679.54
100	25	96	11597.63	0.21	37.73	0.21	5710109	0	1800.70	1055.90
100	25	146	11383.15	7.00	40.33	7.00	6620914	0	1800.07	627.79
150	15	46	19185.95	0.00	34.15	0.08	711678	5	261.92	99.78
150	15	96	17045.60	0.00	36.52	0.05	584711	10	168.77	85.71
150	15	146	16217.38	0.00	34.66	0.09	1033992	37	461.27	218.40
150	20	46	19576.13	2.16	36.55	2.16	4227018	0	1801.40	694.49
150	20	96	17045.60	0.10	37.57	0.10	4835087	0	1801.05	1012.15
150	20	146	16639.80	2.81	37.40	2.81	5399311	0	1800.15	617.98
150	25	46	23723.72	20.20	47.99	20.20	3692975	0	1800.09	497.13
150	25	96	17088.20	0.68	38.09	0.68	6352428	0	1800.26	887.07
150	25	146	17581.65	8.98	41.17	8.98	4326068	0	1800.08	402.36
200	15	46	25131.78	0.00	34.44	0.07	1306181	34	758.54	356.99
200	15	96	23552.48	0.09	36.20	0.09	1415530	0	1800.24	1110.07
200	15	146	21922.45	0.00	34.34	0.11	1389234	29	906.81	538.63
200	20	46	26920.65	6.92	39.84	6.92	3710633	0	1801.07	445.93
200	20	96	23614.18	0.46	37.42	0.46	4446911	0	1800.08	932.16
200	20	146	22181.30	1.46	36.22	1.47	4794355	0	1801.14	559.33
200	25	46	31069.27	21.98	48.21	21.98	3182651	0	1800.17	504.91
200	25	96	24324.03	3.90	39.60	3.90	5701297	0	1800.72	689.62
200	25	146	27407.35	22.50	48.73	22.50	3499123	0	1800.48	335.37

Table A.8: Branch and Cut Results for the Problem with Composition Constraints for $C = 0.3$

n	m	seed	IP	Opt.Gap	LP Gap	Root Gap	User Cuts	Nodes	Time	Sep. Time
50	20	46	6483.02	0.49	34.68	0.49	11770820	0	1800.03	931.02
50	20	96	6133.14	0.60	36.89	0.60	7851602	0	1800.17	1040.15
50	20	146	5376.32	0.63	35.30	0.63	13946831	0	1800.22	866.32
50	25	46	6470.02	0.58	35.10	0.58	14419788	0	1800.26	967.70
50	25	96	6109.14	0.45	37.12	0.45	10145605	0	1800.23	1028.33
50	25	146	5357.32	0.53	35.66	0.53	12498570	0	1800.17	997.87
100	20	46	13532.16	4.00	37.29	4.00	8418915	0	1800.04	715.99
100	20	96	11801.30	0.11	36.83	0.11	7122461	0	1800.17	1080.27
100	20	146	11310.94	4.44	37.96	4.44	8718822	0	1800.13	752.20
100	25	46	13392.22	3.48	37.13	3.48	7513190	0	1800.13	697.09
100	25	96	11803.70	0.52	37.28	0.52	7963462	0	1800.11	1077.04
100	25	146	11596.90	7.12	40.00	7.12	8366958	0	1800.07	778.35
150	20	46	20061.24	3.02	36.46	3.02	6084636	0	1800.06	528.35
150	20	96	17364.32	0.32	37.09	0.32	5022788	0	1800.26	1032.97
150	20	146	16968.20	3.15	37.02	3.15	7793781	0	1800.73	643.16
150	25	46	20299.24	4.51	37.68	4.51	6378007	0	1801.24	754.25
150	25	96	17396.72	1.05	37.64	1.05	8144158	0	1800.06	1003.78
150	25	146	17304.34	5.44	38.73	5.44	7221673	0	1801.61	767.31
200	20	46	27608.64	7.87	39.79	7.87	5045853	0	1800.07	477.13
200	20	96	24038.66	0.50	36.89	0.50	5513277	0	1800.21	905.44
200	20	146	22799.46	2.74	36.32	2.74	5652797	0	1800.31	516.23
200	25	46	28457.76	11.11	42.03	11.11	5455791	0	1800.13	734.70
200	25	96	24186.46	1.94	37.69	1.94	6463392	0	1800.15	970.68
200	25	146	22880.16	3.52	37.03	3.52	6368587	0	1800.43	831.64

Appendix B

Computational Results for the Stochastic Problem

Table B.1: Branch and Cut Results for the Stochastic Problem where $t' = n$ and $n = \{9, 10\}$

n	m	seed	IP	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
9	4	40	922.08	1.37	20.52	2.74	25777	561500	1800.19	2.09
9	4	46	1050.37	1.05	25.73	2.62	25489	855300	1800.07	5.19
9	4	96	945.34	1.39	28.92	2.34	25165	573707	1800.35	7.77
9	6	40	908.17	1.52	21	2.03	70801	441682	1800.79	4.99
9	6	46	1045.53	1.42	27.12	2.99	70892	385800	1800.29	9.93
9	6	96	940.41	1.66	30.34	2.27	64555	418900	1800.33	15.39
10	4	40	1019.08	1.86	20.73	2.96	55113	245300	1800.28	8.09
10	4	46	1156.08	1.46	26.05	2.38	54794	393647	1800.86	22.18
10	4	96	1051.79	1.70	29.06	2.43	53628	261400	1800.80	33.45
10	6	40	1003.02	2.57	21.18	2.87	156812	101408	1800.85	19.11
10	6	46	1149.52	1.95	27.37	3.03	158676	171907	1800.72	26.70
10	6	96	1047.34	2.06	30.55	2.72	135598	223200	1800.31	45.86

Table B.2: Branch and Cut Results for the Stochastic Problem where $t' = n$ and $n = \{11, 12, 13\}$

n	m	seed	IP	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
11	4	40	1118.96	2.97	21.06	3.3	111823	123500	1800.37	31.47
11	4	46	1265.36	2.68	26.42	2.98	117593	136639	1800.60	103.85
11	4	96	1162.77	2.23	29.28	2.65	104093	109600	1800.70	42.33
11	6	40	1099.83	3.02	21.43	3.08	220090	42500	1800.51	61.49
11	6	46	1256.37	2.36	27.64	3.12	186868	80600	1800.46	62.55
11	6	96	1156.81	2.69	30.7	2.63	151562	86800	1800.48	88.04
12	4	40	1217.65	3.69	21.45	3.73	176097	39700	1800.85	124.22
12	4	46	1373.01	2.29	26.64	3.16	171017	43000	1800.34	393.16
12	4	96	1276.02	2.59	29.65	3.15	129103	65660	1800.35	171.75
12	6	40	1204.95	4.21	22.35	4.23	500875	10572	1800.64	235.10
12	6	46	1371.56	3.35	28.3	3.93	405006	14675	1800.64	233.61
12	6	96	1275.31	3.63	31.37	3.96	312583	33308	1800.39	344.82
13	4	40	1326.03	4.72	22.23	4.75	563330	11500	1814.36	760.47
13	4	46	1487.33	3.56	27.11	4.1	416298	19700	1800.51	1052.47
13	4	96	1395.70	3.77	30.27	4.17	342856	20100	1800.74	667.55
13	6	40	1521	18.45	33.71	18.5	1655715	0	1944	1135
13	6	46	1494.48	5.26	29.18	5.42	1295641	7976	1801.14	897.44
13	6	96	1403.72	5.56	32.4	5.57	944296	6739	1800.65	1466.78

Table B.3: Branch and Cut Results for the Stochastic Problem

n	m	t'	seed	IP	Opt. Gap	LP Gap	Root Gap	User cuts	Nodes	Time	Sep. Time
15	6	7	40	1426.66	0.34	19.49	1.96	59350	213947	1800.42	7.19
15	6	7	46	1634.10	0.40	26.57	1.43	43779	184300	1800.27	9.02
15	6	7	96	1546.10	0.34	4.75	1.53	42889	227400	1800.40	7.97
15	9	7	40	1416.99	2.00	18.18	2.66	120779	14510	1800.29	17.31
15	9	7	46	1628.72	1.46	27.20	2.76	97578	44500	1800.31	20.96
15	9	7	96	1538.82	1.79	4.51	2.37	96787	84510	1800.38	24.35
15	6	8	40	1443.75	0.68	19.50	2.11	118971	54400	1800.35	31.73
15	6	8	46	1619.93	0.63	26.41	1.58	86534	96600	1800.49	27.57
15	6	8	96	1535.61	0.80	4.02	1.23	83767	84200	1800.32	36.41
15	9	8	40	1441.52	2.89	15.54	3.33	232382	5914	1800.65	56.97
15	9	8	46	1620.80	2.77	32.99	2.32	181572	18919	1800.70	70.86
15	9	8	96	1535.32	2.40	2.24	2.56	185586	28600	1800.59	81.11
15	6	9	40	1454.64	1.70	17.30	2.51	143436	16711	1800.43	77.10
15	6	9	46	1637.84	1.43	34.12	2.01	83454	54700	1800.65	46.75
15	6	9	96	1551.00	1.19	2.53	1.67	76010	79040	1801.26	44.27
15	9	9	40	1456.35	4.26	16.96	4.26	244624	4620	1800.53	105.74
15	9	9	46	1643.06	4.40	33.92	4.86	226521	11408	1800.98	109.43
15	9	9	96	1550.62	2.61	2.87	2.66	325836	19832	1801.43	248.53
20	6	7	40	1869.72	0.35	41.05	0.89	96154	79600	1800.48	22.47
20	6	7	46	2116.64	0.34	28.16	1.12	62438	122400	1800.34	19.12
20	6	7	96	2053.81	0.51	24.38	0.93	59315	131117	1800.36	19.03
20	9	7	40	1881.76	2.57	41.82	2.93	196825	11701	1800.35	42.48
20	9	7	46	2121.17	2.24	27.80	2.96	161082	25300	1800.95	53.79
20	9	7	96	2055.26	1.79	24.17	1.72	157201	48447	1801.37	59.55
20	6	8	40	1903.12	1.03	42.14	1.96	104864	24932	1800.42	64.21
20	6	8	46	2126.35	0.68	28.27	1.31	62033	79309	1801.02	34.46
20	6	8	96	2053.74	1.07	24.05	1.28	58172	64401	1800.59	33.98
20	9	8	40	1900.99	2.72	40.30	3.23	404917	2300	1800.51	165.25
20	9	8	46	2142.59	3.60	26.37	3.62	316041	6152	1800.23	176.89
20	9	8	96	2054.67	1.60	29.44	1.73	374716	21006	1801.82	315.18
20	6	9	40	1910.63	1.79	40.04	2.66	223977	12017	1800.60	251.00
20	6	9	46	2137.54	0.94	25.95	1.49	136146	29500	1800.46	198.37
20	6	9	96	2067.65	1.34	30.02	1.50	116209	44340	1801.11	137.17
20	9	9	40	2048.88	10.79	43.94	11.12	748160	0	1800.78	488.15
20	9	9	46	2435.73	15.07	34.97	15.08	660907	0	1801.84	655.63
20	9	9	96	2088.73	3.18	30.50	3.32	698795	3022	1838.35	937.70

Table B.4: Decomposition Results

n	m	p.node	seed	UB	LB	Time	Gap
10	4	2	40	1025.51	1000.47	1368.22	2.44
10	4	2	46	1161.36	1133.69	1225.20	2.38
10	4	2	96	1058.07	1040.39	1168.37	1.67
10	6	2	40	1011.25	986.36	1249.96	2.46
10	6	2	46	1163.37	1122.95	1181.51	3.47
10	6	2	96	1052.29	1032.86	1123.69	1.85
11	4	2	40	1149.48	1093.29	1607.53	4.89
11	4	2	46	1297.04	1235.40	1367.97	4.75
11	4	2	96	1193.80	1145.78	1302.13	4.02
11	6	2	40	1111.63	1077.10	1470.64	3.11
11	6	2	46	1286.24	1223.28	1360.50	4.89
11	6	2	96	1175.09	1135.84	1288.66	3.34
12	4	2	40	1239.06	1178.55	1466.17	4.88
12	4	2	46	1423.63	1331.27	1408.67	6.49
12	4	2	96	1325.63	1242.03	1326.21	6.31
12	6	2	40	1220.63	1165.10	2243.32	4.55
12	6	2	46	1405.04	1323.01	1889.03	5.84
12	6	2	96	1302.37	1237.48	1702.47	4.98
13	4	2	40	1361.00	1268.84	2080.89	6.77
13	4	2	46	1534.33	1432.52	1961.22	6.64
13	4	2	96	1446.79	1345.72	1802.67	6.99
13	6	2	40	1357.81	1251.19	2038.52	7.85
13	6	2	46	1511.29	1419.12	1824.36	6.10
13	6	2	96	1430.86	1336.08	1313.65	6.62
7	4	3	40	752.08	708.93	788.41	5.74
7	4	3	46	868.00	824.14	668.27	4.26
7	4	3	96	784.76	746.88	616.47	3.87
8	4	3	40	844.20	796.66	1289.89	5.63
8	4	3	46	983.50	919.50	1222.46	6.51
8	4	3	96	893.41	843.23	1216.02	5.62
8	6	3	40	838.38	788.54	1485.34	5.94
8	6	3	46	965.53	914.53	1361.40	5.28
8	6	3	96	888.17	840.19	1309.03	5.40
9	4	3	40	945.20	886.64	1319.28	6.20
9	4	3	46	1084.57	1020.15	1563.67	5.94
9	4	3	96	1012.16	946.20	1362.06	6.52
9	6	3	40	929.36	877.96	2598.58	5.53
9	6	3	46	1080.73	1014.90	2092.75	6.09
9	6	3	96	1007.67	942.74	1894.03	6.44

Table B.5: Decomposition Results for Problems with four possible outcomes

n	m	p.node	seed	UB	LB	Time	Gap
7	4	4	40	751.19	708.16	1763.89	5.73
7	4	4	46	850.58	810.31	1655.02	4.74
7	4	4	96	790.35	744.37	1506.67	5.82
8	4	4	40	859.50	794.78	1386.87	7.53
8	4	4	46	985.68	903.12	1498.14	8.38
8	4	4	96	923.13	839.24	1209.58	9.09