Probabilistic Choice Models for Product Pricing Using Reservation Prices

by

Betsy Y. Hui

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.
I understand that my thesis may be made electronically available to the public.

Betsy Y. Hui

Abstract

The problem of pricing a product line to maximize profits is an important challenge faced by many companies. To address this problem, we discuss four different probabilistic choice models that are based on reservation prices: the Uniform Distribution Model, the Weighted Uniform Model, the Share-of-Surplus Model, and the Price Sensitive Model. They are formulated as convex mixed-integer mathematical programs. We explore the properties and additional valid inequalities of these formulations. We also compare their optimal solutions on a set of inputs. In general, the Uniform Distribution, Weighted Uniform, and Price Sensitive Models have the same optimal solution while the Share-of-Surplus Model gives a different solution in many cases.

We develop a few heuristics for finding good feasible solutions. These simple and efficient heuristics perform well and help to improve the solution time. Computational results of solving problem instances of various sizes are shown.

Acknowledgements

I would like to thank my supervisor, Professor Romy Shioda, for her guidance, support, and patience that made this thesis possible. I am also grateful for Professor Levent Tuncel's many helpful comments and suggestions throughout my research. In addition, I would like to thank my readers, Dean Tom Coleman and Professor Levent Tuncel, for their invaluable feedback.

Contents

List of Tables

Chapter 1

Introduction

One of the key revenue management challenges for a company is to determine the "right" price for each of their product line. Generally speaking, a company wants to set the prices to maximize their total profit. The challenge arises from the complex relationship between the product prices and the total profit. For example, how do the prices affect the demand for each product? In cases where multiple products are offered by the company, the price and demand for a product cannot be considered in isolation from the other products. Clearly, a customer's decision to purchase a product can be swayed by the relative prices of similar products offered by the same company. Thus, prices need to be set to avoid "cannibalizing" their product line. For example, if there are high margin and low margin products, setting the price of the latter too low may decrease demand for the high margin product, thus resulting in lower profit.

In this thesis, we study several different models for product pricing from a mathematical programming perspective. The models differ from one another according to different assumptions on customer purchasing behavior. Before we discuss the details of our approach, let us first give a brief overview of relevant work in this area.

1.1 Background

Revenue management in general is the practice of maximizing a company's revenue by optimally choosing the price of products and which customers to serve at any given time.

It has been used extensively in the airline, hotel, and manufacturing industry (see [7] and [9] for a comprehensive overview of the history of revenue management).

Product pricing is a sub-problem of revenue management, focused on determining the optimal prices of a product line. There are many variations on product pricing framework depending on the setting. For example, there is the single-product, multi-customer setting, which is primarily concerned with what price to offer to different customer segments. Airline revenue management is one of the most popular examples of this context, where business travelers, leisure travelers and budget travelers are offered different prices for the same flight, depending on the lead time of purchase and additional options (e.g., partially refundable tickets). An alternative framework is the multi-product, multi-customer setting where every customer is offered the same price for a given product, but different customer segments have varying preferences. This is more of a combinatorial problem where given the customer preference information, the prices need to be set to maximize total revenue. We will focus on the second type of problem in this thesis.

In general, suppose a company has m different product lines and market analysis tells them that there are n distinct customer segments, where customers of the same segment behave the "same". A key revenue management problem is to determine optimal prices for each product to maximize total revenue, given the customer choice behavior. There are multitudes of models for customer choice behavior [9], but this paper focuses solely on those based on reservation prices.

Let R_{ij} denote the *reservation price* of segment i for product j, $i = 1, \ldots, n$, $j = 1, \ldots, m$, which reflects how much customers of segment i are *willing* and *able* to spend on product j. R_{ij} is not only the dollar amount that product j is worth to customers in segment i, but it also reflects how much they are able to pay for it. For example, if a customer segment believes that a 7 day vacation to St. Lucia is worth \$2,000, but they can only afford \$1,000 for a vacation, then their reservation price for St. Lucia is \$1,000. Without loss of generality, we make the following assumption:

Assumption 1.1.1. R_{ij} is a nonnegative integer for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

If the price of product j is set to $\$\pi_j$, $\pi_j \geq 0$, then the *utility* or *surplus* (we will use these terms interchangeably throughout the thesis) of segment i for product j is the difference between the reservation price and the price, i.e., $R_{ij} - \pi_j$.

Finally, we assume that reservation prices are the same for every customer in a given segment and each segment pays the same price for each product. Customer choice models based on reservation prices assume that customer purchasing behavior can be fully determined by their reservation price and the price of products.

Even in a reservation price framework, there are several different models for customer choice behavior in the literature. In [2, 3], the authors proposed a pricing model that maximizes profits with the assumption that each customer segment only buys the product with the maximum surplus if the surplus is nonnegative. This model is often referred to as the *maximum utility* or *envy free pricing* model. In this model, each segment buys at most one product. The authors modeled the problem as a nonconvex, nonlinear mixed-integer programming problem and solved the problem using a variety of heuristic approaches.

In [6], the authors examined a Share-of-Surplus Choice Model in which the probability that a segment will choose a product is the ratio of its surplus versus the total surplus for the segment across all products with nonnegative surplus. They proposed a heuristic which involves decomposing the problem into hypercubes and use a simulated annealing algorithm to find the best hypercube. Solutions found by the heuristic for problems with sizes up to 5 products and 10 segments were shown to be near-optimal.

Another approach of pricing multiple products is to consider the problem of bundle pricing [4]. It is the problem of determining whether it is more profitable to offer some of the products together as a package or individually, and what prices should be assigned to the bundles or individual products to maximize profit. The authors formulated the bundle pricing problem as a mixed integer linear problem using a disjunctive programming technique [1].

Some research has been done on partitioning customers into segments by the probability that they would buy each product. In [5], the authors proposed a segmentation approach that groups the customers according to their reservation prices and price sensitivity. The probability of a segment choosing a product j is modeled as a multinomial logit model with the segment's reservation price, price sensitivity, and the price of the product j as parameters. Unlike their model, we do not consider price sensitivity in this paper as a criterion when we partition customers into segments and we assume that all segments react to price changes in the same way.

In this thesis, we assume that the reservation prices for each customer segment and product are given. Given different models of customer purchasing behavior, we aim to formulate and solve the corresponding revenue maximization problem as a mixed-integer programming problem. In the Appendix, we will discuss how we performed the customer segmentation and estimated the reservation prices from real purchase orders of a Canadian travel company.

1.2 Probabilistic Choice Models

In this section, we will introduce the general framework of probabilistic customer choice models that determines the probability that customer segment i will purchase product j, $i = 1, \ldots, n$, $j = 1, \ldots, m$.

It is often assumed that a segment will only consider purchasing a product with non-negative utility, i.e.,

Assumption 1.2.1. If segment i buys product j, then $R_{ij} - \pi_j \geq 0$, $i = 1, \ldots, m$, $j = 1, \ldots, m$.

Let β_{ij} be binary decision variables:

$$\beta_{ij} := \begin{cases} 1, & \text{if and only if the surplus of product } j \text{ is nonnegative for segment } i, j, \\ 0, & \text{otherwise.} \end{cases}$$

i.e., $\beta_{ij} = 1$ if and only if $R_{ij} - \pi_j \geq 0$ and $\beta_{ij} = 0$ if and only if $R_{ij} - \pi_j < 0$, where, again, π_j is the decision variable for the price of product j. This relationship can be naively modelled by:

$$(R_{ij} - \pi_j)\beta_{ij} \geq 0,$$

$$(R_{ij} - \pi_j)(1 - \beta_{ij}) \leq 0,$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. To linearize the above inequalities, we can use a disjunctive programming trick. Let p_{ij} be an auxiliary variable where $p_{ij} = \pi_j \beta_{ij}$, i.e,

$$p_{ij} := \begin{cases} \pi_j, & \text{if } \beta_{ij} = 1, \\ 0, & \text{otherwise.} \end{cases}$$

This relationship can be modeled by the following set of linear inequalities:

$$
\begin{aligned}
p_{ij} &\geq 0, \\
p_{ij} &\leq \pi_j, \\
p_{ij} &\leq R_{ij}\beta_{ij}, \\
p_{ij} &\geq \pi_j - (\max_{i=1,\ldots,n} R_{ij} + 1)(1 - \beta_{ij}),
\end{aligned}
$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. The first two inequalities set $p_{ij} = 0$ when $\beta_{ij} = 0$, and the last two inequalities set $p_{ij} = \pi_j$ when $\beta_{ij} = 0$. R_{ij} is a valid upperbound for p_{ij} since if $p_{ij} > R_{ij}$, then $\beta_{ij} = 0$ and thus $p_{ij} = 0$. Also, $\max_{i=1,\ldots,n} R_{ij} + 1$ is a valid upperbound for π_j since no segment will buy product j if $\pi_j > R_{ij}$ for all $i = 1, \ldots, n$. We also need the following constraint to force β_{ij} to be 1 when R_{ij} equals to π_j for all j:

$$(R_{ij} - \pi_j + 1) \leq (R_{ij} - \min_i R_{ij} + 1)\beta_{ij} \quad \forall i, \forall j,$$

which is valid under Assumption 1.1.1.

Where $\boldsymbol{\pi}, \boldsymbol{\beta}$, and $\boldsymbol{p}$ are vectors of π_j, β_{ij} and p_{ij}, respectively, let P be the following polyhedron:

$$
\begin{aligned}
P = \{(\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{p}) : \quad & R_{ij}\beta_{ij} - p_{ij} \geq 0, && i = 1, \ldots, n, j = 1, \ldots, m, \\
& R_{ij}(1 - \beta_{ij}) - \pi_j \leq 0, && i = 1, \ldots, n, j = 1, \ldots, m, \\
& p_{ij} \leq \pi_j, && i = 1, \ldots, n, j = 1, \ldots, m, \\
& R_{ij} - \pi_j + 1 \leq (R_{ij} - \min_i R_{ij} + 1)\beta_{ij} && i = 1, \ldots, n, j = 1, \ldots, m, \\
& p_{ij} \geq 0, \pi_j \geq 0, && i = 1, \ldots, n, j = 1, \ldots, m\}.
\end{aligned}
$$

Thus, to model the condition in Assumption 1.2.1, we need to set prices π_j and β_{ij} such that $\beta \in \{0, 1\}$ and $(\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{p}) \in P$.

There are ambiguities regarding the choices between multiple products with nonnegative utility. Given all the products with nonnegative surplus, which products would the customer buy? Are there some products they are more likely to buy than others? In a probabilistic choice framework, we need to determine the probability Pr_{ij} that segment i

buys product j. Then the *expected* revenue for the company is

$$\sum_{i=1}^{n} N_i E[\text{revenue earned from segment } i] = \sum_{i=1}^{n} N_i \sum_{j=1}^{m} \pi_j Pr_{ij}.$$

In our revenue management problem, we can interpret Pr_{ij} as the fraction of customers of segment i that buys product j, i.e., the expected revenue is

$$\sum_{j=1}^{m} \pi_j E[\text{number of customers in segment } i \text{ that buys product } j] = \sum_{j=1}^{m} \pi_j \sum_{i=1}^{n} N_i Pr_{ij}.$$

Furthermore, Pr_{ij} is positive if and only if the surplus of product j is nonnegative for segment i.

Thus, the expected revenue maximization problem is:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} N_i \pi_j Pr_{ij}, && (1.1)\\
\text{s.t.} \quad & Pr_{ij} > 0 \Leftrightarrow \beta_{ij} = 1, \quad i = 1,\ldots,n; j = 1,\ldots,m,\\
& Pr_{ij} = 0 \Leftrightarrow \beta_{ij} = 0, \quad i = 1,\ldots,n; j = 1,\ldots,m,\\
& (\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{p}) \in P,\\
& \beta_{ij} \in \{0,1\}, \qquad i = 1,\ldots,n; j = 1,\ldots,m.
\end{aligned}$$

All the probabilistic choice models explored in this thesis is based on the optimization problem (1.1). What differentiates the different model is how Pr_{ij} is defined.

One of the most popular probabilistic choice model in the marketing literature may be the *multinomial logit* (MNL) model,

$$Pr_{ij} = \frac{e^{v_{ij}}}{\sum_k e^{v_{ik}}},$$

where v_{ij} represent the utility or desirability of the product j to segment i. Clearly, there are wide varieties to how this v_{ij} is modeled as well. The main motive for the exponential is to allow v_{ij} to take on any real value. For example, an alternative model is to have

$$Pr_{ij} = \frac{v_{ij}}{\sum_k v_{ik}},$$

but we would then require $v_{ij} \geq 0$ and $\sum_k v_{ik} > 0$, which could be easily addressed in many cases.

In this thesis, we examine several probabilistic choice model from a mathematical programming perspective. Depending on how Pr_{ij} is modeled, we can formulate the optimization problem (1.1) as a convex mixed-integer programming problem (MIP). In Chapter 2, we assume that Pr_{ij} is uniform across all products with nonnegative surplus. We call this model the *Uniform Distribution Model*. In Chapter 3, we modify the Uniform Distribution Model so that customers are more likely to purchase products with higher reservation prices. We call this model the *Weighted Uniform Model*. In Chapter 4, we explore mathematical programming formulations of the Share-of-Surplus Model proposed in [6], including an MIP formulation for the case with restricted prices. Chapter 5 explores the *Price Sensitive Model* where Pr_{ij} decreases as the price of product j increases. In Chapter 6, we discuss special properties of the optimal solutions and compare the optimal prices π_j and variables β_{ij} of the different models. We also consider enhancements to the models (Chapter 7), including heuristics to determine good feasible solutions quickly and valid inequalities to speed up the solution time of the MIP. In Chapter 8, we show how we can incorporate product capacity limits and product costs into the models . We illustrate some computational results of our models in Chapter 9 and conclude and discuss future work in Chapter 10.

Chapter 2

Uniform Distribution Model

2.1 The Model

A very simple model of customer choice behavior is to assume that each segment chooses products with a uniform distribution across all products with nonnegative surplus. We call this model the *Uniform Distribution Model*. Again, let β_{ij} be binary decision variables where

$$\beta_{ij} := \begin{cases} 1, & \text{if the surplus of product } j \text{ is nonnegative for segment } i,\ j, \\ 0, & \text{otherwise.} \end{cases}$$

i.e., $\beta_{ij} = 1$ if and only if $R_{ij} - \pi_j \geq 0$.
Then in the Uniform Distribution Model, the probability that the customer segment i buys product j would be

$$Pr_{ij} := \begin{cases} 0, & \text{if } \sum_{j=1}^{m} \beta_{ij} = 0, \\ \dfrac{\beta_{ij}}{\sum_{k=1}^{m} \beta_{ik}}, & \text{otherwise.} \end{cases}$$

Under this assumption, the expected revenue is

$$\sum_{i=1}^{n} N_i t_i$$

where

$$t_i := \begin{cases} \sum_{j=1}^{m} \pi_j \frac{\beta_{ij}}{\sum_{k=1}^{m} \beta_{ik}} = \frac{\sum_{j=1}^{m} p_{ij}}{\sum_{k=1}^{m} \beta_{ik}}, & \text{if } \sum_{j=1}^{m} \beta_{ij} \neq 0, \\ 0, & \text{otherwise,} \end{cases}$$

where p_{ij} is the auxiliary variable in Section 1.2 such that $p_{ij} := \pi_j \beta_{ij}$. Thus, t_i corresponds to the average price that segment i pays.

We further reformulate the problem to

$$\begin{array}{lrl} \max & \sum_{i=1}^{n} N_i t_i & \\ \text{s.t.} & \sum_{j=1}^{m} \beta_{ij} t_i \leq \sum_{j=0}^{m} p_{ij}, & \forall i, \\ & t_i \leq (\max_k R_{ik}) \sum_j \beta_{ij}, & \forall i, \\ & (\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\beta}) \in P, & \\ & \beta_{ij} \in \{0, 1\}, & \forall i, j. \end{array}$$

where P is the polyhedron defined in Section 1.2.

Let us introduce yet another auxiliary variable a_{ij} such that $a_{ij} = t_i \beta_{ij}$, i.e., $a_{ij} = t_i$ if $\beta_{ij} = 1$ and $a_{ij} = 0$ otherwise. Then the above formulation can be converted to a linear mixed-integer programming problem

$$\begin{array}{lrl} \max & \sum_{i=1}^{n} N_i t_i, & \\ \text{s.t.} & \sum_{j=1}^{m} a_{ij} \leq \sum_{j=1}^{m} p_{ij}, & \forall i, \\ & t_i \leq (\max_k R_{ik}) \sum_j \beta_{ij}, & \forall i, \\ & a_{ij} \leq (\max_k R_{ik}) \beta_{ij}, & \forall i, \forall j, \\ & a_{ij} \leq t_i, & \forall i, \forall j, \\ & a_{ij} \geq t_i - (\max_k R_{ik})(1 - \beta_{ij}), & \forall i, \forall j, \\ & (\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\beta}) \in P, & \\ & \beta_{ij} \in \{0, 1\}, & \forall i, j. \end{array} \tag{2.1}$$

To make this Uniform Distribution Model more realistic, we may want to bound the absolute difference of the positive surpluses by adding the constraints

$$|(R_{ij} - \pi_j) - (R_{ik} - \pi_k)| \leq \Delta \quad \forall i, \forall j \neq k \text{ where } \beta_{ij} = \beta_{ik} = 1$$

where Δ is some user-defined constant. The linearized form of this constraint is

$$(R_{ij}\beta_{ij} - p_{ij}) - (R_{ik}\beta_{ik} - p_{ik}) \leq \Delta + R_{ij}(1 - \beta_{ik}), \quad \forall i, \forall j \neq k$$

$$(R_{ik}\beta_{ik} - p_{ik}) - (R_{ij}\beta_{ij} - p_{ij}) \leq \Delta + R_{ik}(1 - \beta_{ij}), \quad \forall i, \forall j \neq k.$$

2.2 Alternative Formulation

Note that if β_{ij}'s are given in Problem (2.1), then

$$\pi_j = \min_{i:\beta_{ij}=1} R_{ij}$$

in the optimal solution if $\beta_{ij} = 1$ for some i, $i = 1, \ldots, n$. If $\beta_{ij} = 0$ for all i (i.e., product j is not bought by any customer), then $\pi_j > \max\limits_{i=1,\ldots,n} R_{ij}$.

Let us introduce a dummy customer segment, segment 0, where $R_{0j} > \max_{i=1,\ldots,n} R_{ij}$ and $N_0 = 0$, and a binary decision variable x_{ij} where:

$$x_{ij} := \begin{cases} 1, & \text{if segment } i \text{ has the smallest reservation price out of all} \\ & \text{segments with nonnegative surplus for product } j, \\ 0, & \text{otherwise.} \end{cases}$$

With the constraint $\sum_{i=0}^{n} x_{ij} = 1$ for all products j, we get

$$\pi_j = \sum_{i=0}^{n} R_{ij}x_{ij}, \qquad \beta_{ij} = \sum_{l:R_{lj} \leq R_{ij}} x_{lj}.$$

Thus the continuous variables p_{ij} and π_j can be eliminated. Using the x_{ij} variables, the objective function of the Uniform Distribution Model is

$$\sum_{i=0}^{n} N_i \sum_{j=1}^{m} \left(\sum_{i=0}^{n} R_{ij}x_{ij} \right) \left(\frac{\sum_{l:R_{lj} \leq R_{ij}} x_{lj}}{\sum_{k=1}^{m} \sum_{l:R_{lk} \leq R_{ik}} x_{lk}} \right) = \sum_{i=0}^{n} N_i \left(\frac{\sum_{j=1}^{m} \sum_{l:R_{lj} \leq R_{ij}} R_{lj}x_{lj}}{\sum_{k=1}^{m} \sum_{l:R_{lk} \leq R_{ik}} x_{lk}} \right)$$

where the equality follows from $\sum_{i=0}^{n} x_{ij} = 1, \forall j$, $x_{ij}^2 = x_{ij}$, and $x_{ij}x_{lj} = 0$ for $i \neq l$.

Then:

$$t_i := \begin{cases} \frac{\sum_j \sum_{l:R_{lj} \le R_{ij}} R_{lj} x_{lj}}{\sum_k \sum_{l:R_{lk} \le R_{ik}} x_{lk}}, & \text{if } \sum_k \sum_{l:R_{lk} \le R_{ik}} x_{lk} \neq 0, \\ 0, & \text{otherwise.} \end{cases}$$

Then Model (2.1) is equivalent to

$$\begin{array}{llr} \max & \sum_{i=1}^{n} N_i t_i, & (2.2) \\ \text{s.t.} & \sum_{i=0}^{n} x_{ij} = 1, \quad \forall j, & \\ & \sum_{j=1}^{m} a_{ij} \le \sum_{j=1}^{m} \sum_{l:R_{lj} \le R_{ij}} R_{lj} x_{lj}, \quad \forall i, & \\ & a_{ij} \le t_i, \quad \forall i, j, & \\ & a_{ij} \le (\max_k R_{ik}) \sum_{l:R_{lj} \le R_{ij}} x_{lj}, \quad \forall i, j, & \\ & a_{ij} \ge t_i - (\max_k R_{ik}) \sum_{l:R_{lj} > R_{ij}} x_{lj}, \quad \forall i, j, & \\ & t_i \le (\max_k R_{ik}) \sum_{j=1}^{m} \sum_{l:R_{lj} \le R_{ij}} x_{lj}, \quad \forall i, & \\ & t_i \ge 0, \quad \forall l, i, & \\ & a_{ij} \ge 0, \quad \forall i, j, & \\ & x_{ij} \in \{0, 1\}, \quad \forall i, j. & \end{array}$$

2.3 Strength of the Two Formulations

Aside from computational experimentation, we wish to compare the relative “strength” of the original and alternative mixed-integer programming formulations of the Uniform Distribution Model. Namely, let us compare the strength of the LP relaxation of formulations (2.1) and (2.2).

Let F_1 be the feasible region of the LP relaxation of (2.1) and let F_2 be that of (2.2). We compare both formulations on the same data n (the number of customers), m (number of product) and R_{ij} (reservation price), $i = 1, \ldots, n$, $j = 1, \ldots, m$. However, note that we add a dummy customer segment 0 in the alternate formulation (2.2). Thus, (2.2) has customer segments $i = 0, 1, \ldots, n$, and $R_{0j} = \max_{i=1,\ldots,n} R_{ij} + 1$.

Let $\Pi_t(F_k)$ be the projection of the set F_k, $k = 1, 2$, onto the variables t_i, $i = 1, \ldots, n$, i.e.,

$$\Pi_t(F_1) := \{\boldsymbol{t} : \exists(\boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{p}, \boldsymbol{a}) \text{ such that } (\boldsymbol{t}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{p}, \boldsymbol{a}) \in F_1\}$$

and

$$\Pi_t(F_2) := \{\boldsymbol{t} : \exists(t_0, \boldsymbol{x}, \boldsymbol{a}) \text{ such that } (t_0, \boldsymbol{t}, \boldsymbol{x}, \boldsymbol{a}) \in F_2\}$$

where $\boldsymbol{t}$ is the vector of t_i's , $i = 1, \ldots, n$, $\boldsymbol{a}$ is the vector of a_{ij}'s, $i = 1, \ldots, n$, $j = 1, \ldots, m$, $\boldsymbol{\beta}$ is the vector of β_{ij}'s, $i = 1, \ldots, n$, $j = 1, \ldots, m$, $\boldsymbol{\pi}$ is the vector of π_j's, $j = 1, \ldots, m$, $\boldsymbol{p}$ is the vector of p_{ij}'s, $i = 1, \ldots, n$, $j = 1, \ldots, m$, and $\boldsymbol{x}$ is the vector of x_{ij}'s, $i = 1, \ldots, n$, $j = 1, \ldots, m$. The following lemma shows that $\Pi_t(F_2)$ is strictly contained inside $\Pi_t(F_1)$, implying that the optimal objective value of the LP relaxation of (2.2) is less than or equal to that of of (2.1).

Lemma 2.3.1. $\Pi_t(F_2) \subset \Pi_t(F_1)$ *and the inclusion is strict.*

Proof. To show the inclusion, suppose $(\hat{t}_0, \hat{\boldsymbol{t}}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{a}}) \in F_2$. Let $\bar{\beta}_{ij} = \sum_{l:R_{lj} \leq R_{ij}} \hat{x}_{lj}$, $\bar{\pi}_j = \sum_{i=1}^{n} R_{ij}\hat{x}_{ij}$, $\bar{p}_{ij} = \sum_{l:R_{lj} \leq R_{ij}} R_{lj}\hat{x}_{lj}$, $\bar{a}_{ij} = \hat{a}_{ij}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$. It is easy to see that $(\hat{\boldsymbol{t}}, \bar{\boldsymbol{\beta}}, \bar{\boldsymbol{\pi}}, \bar{\boldsymbol{p}}, \bar{\boldsymbol{a}}) \in F_1$. Thus, $\Pi_t(F_2) \subseteq \Pi_t(F_1)$

To show that the inclusion is strict, let $n = 2$, $m = 2$, $N_1 = N_2 = 1$, and $\boldsymbol{R} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$ be the matrix of reservation prices where rows are the segments and products are the columns. F_1 contains the following point:

$$\boldsymbol{t} = \begin{bmatrix} 2 \\ 0.5 \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} 0 & 1 \\ 0 & 0.5 \end{bmatrix}, \quad \boldsymbol{\pi} = \begin{bmatrix} 2 & 2 \end{bmatrix}, \quad \boldsymbol{p} = \begin{bmatrix} 0 & 2 \\ 0 & 0.5 \end{bmatrix}, \quad \boldsymbol{a} = \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}$$

where again, the rows correspond to segments and the columns correspond to products. The dummy segment in (2.2) has reservation prices $R_{01} = 2$ and $R_{02} = 3$. We will show that $\boldsymbol{t} = \begin{bmatrix} 2 \\ 0.5 \end{bmatrix} \notin \Pi_t(F_2)$.

Given $t_1 = 2$ and $t_2 = 0.5$, the fifth set of constraints in (2.2) are $a_{11} \geq 2 - 2x_{01}$ and $a_{12} \geq 2 - 2x_{02}$. The second set of constraints yield $a_{11} + a_{12} \leq x_{11} + 2x_{12} + x_{22}$. Combining it with the above two inequalities and the first set of constraints gives us

$x_{01}+x_{12}+x_{02}-x_{21} \geq 2$, implying $x_{01}+x_{12}+x_{02} \geq 2$. With the first set of constraints, this yields $x_{01} \geq 1$ and $x_{12}+x_{02} \geq 1$, implying $x_{11}=x_{21}=0$ and $x_{22}=0$. The sixth set of constraints $t_2 \leq x_{21}+x_{22}=0$ contradicts $t_2=0.5$. Thus, $\boldsymbol{t}=\begin{bmatrix} 2 \\ 0.5 \end{bmatrix} \notin \Pi_t(F_2)$.

□

Thus, to test the empirical running time of the Uniform Distribution Model, we will use the alternative MIP formulation (2.2) instead of (2.1). Chapter 9 illustrates the running time of the Uniform Distribution Model on various problem sizes.

2.4 A Pure 0-1 Formulation

It turns out that the Uniform Distribution Model can also be formulated as a pure 0-1 optimization problem. For $k=0,\ldots,m$, let

$$y_{ik} := \begin{cases} 1, & \text{if segment } i \text{ has exactly } k \text{ products with nonnegative surplus,} \\ 0, & \text{otherwise.} \end{cases}$$

Then, the probability that segment i buys product j is

$$\sum_{k=1}^{m} \frac{1}{k}\beta_{ij} y_{ik},$$

Thus, with variables β_{ij}, π_j and x_{ij} defined earlier, the objective function is:

$$\sum_{i=1}^{n} N_i \sum_{j=1}^{m} \pi_j Pr_{ij},$$

$$= \sum_{i=1}^{n} N_i \sum_{j=1}^{m} \pi_j \left(\sum_{k=1}^{m} \frac{1}{k}\beta_{ij} y_{ik}\right),$$

$$= \sum_{i=1}^{n} N_i \sum_{j=1}^{m} \sum_{l:R_{lj} \leq R_{ij}} R_{lj} x_{lj} \left(\sum_{k=1}^{m} \frac{1}{k} y_{ik}\right).$$

Let $z_{ljik} := x_{lj} y_{ik}, \forall i, l : R_{lj} \leq R_{ij}, \forall j, k=1,\ldots,m$. Then the Uniform Distribution Model can be modeled by the the following 0-1 programming problem:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} N_i \sum_{j=1}^{m} \sum_{l:R_{lj} \leq R_{ij}} R_{lj} \sum_{k=1}^{m} \frac{1}{k} z_{ljik}, && \quad (2.3)\\
\text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} = 1, && j = 1, \ldots, m,\\
& \sum_{k=0}^{m} y_{ik} = 1, && i = 1, \ldots, n,\\
& \sum_{k=1}^{m} \sum_{j=1}^{m} \sum_{l:R_{l,j} \leq R_{i,j}} \frac{1}{k} z_{l,j,i,k} = 1 - y_{i,0}, && i = 1, \ldots, n,\\
& \sum_{j=1}^{m} \sum_{l:R_{l,j} \leq R_{ij}} x_{lj} = \sum_{k=0}^{m} k y_{ik} && i, \ldots, m,\\
& z_{l,j,i,k} \leq x_{l,j}, && \forall i, \forall j, k = 1, \ldots, m; l : R_{l,j} \leq R_{i,j},\\
& z_{l,j,i,k} \leq y_{i,k}, && \forall i, \forall j, k = 1, \ldots, m; l : R_{l,j} \leq R_{i,j},\\
& z_{l,j,i,k} \geq x_{l,j} + y_{i,k} - 1, && \forall i, \forall j, k = 1, \ldots, m; l : R_{l,j} \leq R_{i,j},\\
& x_{i,j} \in \{0, 1\}, && \forall i, \forall j,\\
& y_{i,k} \in \{0, 1\}, && \forall i, k = 0, \ldots, m,\\
& 0 \leq z_{l,j,i,k} \leq 1, && \forall i, \forall j, k = 1, \ldots, m; l : R_{l,j} \leq R_{i,j},
\end{aligned}
$$

2.4.1 Preliminary Computational Results

To compare the empirical performance of the pure 0-1 formulation (2.3) and the previous mixed 0-1 formulation (2.2), we randomly generated multiple instance of reservations prices R_{ij} for various n (number of customer segments) and m (number of products). For each (n, m), five random instances were generated. Both models were run with default parameter settings of CPLEX 9.1. The results are shown in Table 2.1.

			Uniform Alternate Formulation (2.2)				Uniform Pure 0-1 Model (2.3)			
n	m	v	LP optval	SM itn	nodes	time	LP optval	SM itn	nodes	time
4	4	1	2304.79	58	3	0.018	2564.71	206	0	1.320
		2	3447.79	17	0	0.007	**3404.00**	78	0	0.070
		3	333.60	60	0	0.008	**333.00**	62	0	0.050
		4	3005.67	25	0	0.002	3060.92	64	0	0.060
		5	3294.81	31	0	0.007	3360.95	103	0	0.090
4	10	1	382.54	157	42	0.065	406.42	4132	83	16.460
		2	381.85	142	3	0.059	398.19	1109	26	365.750
		3	358.60	107	13	0.056	397.36	1859	40	11.860
		4	355.97	105	0	0.037	389.98	496	0	5.230
		5	394.23	90	0	0.029	402.74	267	0	0.630
10	4	1	744.71	196	12	0.106	802.93	4106	52	28.620
		2	845.80	266	35	0.110	856.12	1195	17	803.000
		3	799.50	259	31	0.117	850.95	5320	68	31.880
		4	809.58	159	0	0.033	856.85	972	3	15.600
		5	883.05	99	0	0.031	925.44	1111	21	15.070
10	10	1	985.58	359	36	0.424	997.40	6105	57	240.610
		2	991.44	253	8	0.150	1008.53	5123	45	183.780
		3	1016.35	269	0	0.137	1021.94	1016	0	84.340
		4	825.48	18666	2630	2.762	872.92	139656	1138	1849.010
		5	1014.14	357	19	0.161	1021.50	1309	12	121.720

Table 2.1: Comparison of Uniform Model formulations (2.2) and (2.3).

Table 2.1 compares formulations (2.2) and (2.3) in terms of the objective value of their linear programming relaxation ("LP optval"), total number of dual simplex iterations ("SM itn"), total number of branch-and-bound nodes ("nodes"), and total CPU seconds required to find a provable optimal solution ("time"). n is the number of customer segments, m is the number of products, and v is a label of the problem instance. Bolded LP optval correspond to the integer optimal value.

The above results clearly show that the mixed-integer formulation (2.2) is far superior to the pure 0-1 formulation (2.3) in terms of total running time. This is not surprising since the latter formulation involves significantly more variables, thus the per node computation time is expected to be longer. However, it may be surprising that in almost all cases, the pure 0-1 formulation has a weaker LP relaxation than the mixed-integer formulation and requires more branch-and-bound nodes.

These preliminary computational results may indicate that there is no merit in studying the pure 0-1 formulation. However, since the constraints for (2.3) can be represented by 0-1 knapsack constraints, there may be strong cover inequalities that can be generated from them. Furthermore, these inequalities can be projected down to the space of x_{ij} variables in the alternate mixed-integer formulation (2.2). We further explore this idea in the Section 7.2.

Chapter 3

Weighted Uniform Model

3.1 The Model

In this chapter, we modify the Uniform Distribution Model of Chapter 2 so that customers are more likely to purchase a product with higher reservation price. This model, which we call the *Weighted Uniform Model*, is inspired by the multinomial-logit (MNL) model discussed in Section 1.2. Let $v_{ij} = R_{ij}$, but only consider products with nonnegative surplus. Let

$$Pr_{ij} := \begin{cases} 0, & \text{if } \sum_{j=1}^{m} R_{ij}\beta_{ij} = 0, \\ \frac{u(R_{ij})\beta_{ij}}{\sum_{k=1}^{m} u(R_{ik})\beta_{ik}}, & \text{otherwise.} \end{cases}$$

where $u(\cdot)$ is a monotonically increasing function of R_{ij}. Thus, with this definition of Pr_{ij}, out of all products with nonnegative surplus, a customer is more likely to buy a product with higher reservation price. In the marketing literature, $u(x) = \exp(x)$ is a common function for the MNL model since $u(x) > 0$ for all $x \in \mathbb{R}, x < \infty$. However, since from Assumption 1.1.1 $R_{ij} \geq 0, \forall i, j$, we define $u(x) = x$, i.e.,

$$Pr_{ij} = \frac{R_{ij}\beta_{ij}}{\sum_{k=1}^{m} R_{ik}\beta_{ik}}, \quad \text{if } \sum_{j=1}^{m} R_{ij}\beta_{ij} \geq 1$$

.

Analogous to Model (2.1), the corresponding expected revenue maximizing problem is

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} N_i t_i, && && (3.1)\\
\text{s.t.} \quad & \textstyle\sum_{j=1}^{m} R_{ij} a_{ij} \le \sum_{j=1}^{m} R_{ij} p_{ij}, && \forall i,\\
& t_i \le (\max_k R_{ik}) \textstyle\sum_{j=1}^{m} R_{ij} \beta_{ij}, && \forall i,\\
& a_{ij} \le (\max_k R_{ik}) \beta_{ij}, && \forall i, \forall j,\\
& a_{ij} \le t_i, && \forall i, \forall j,\\
& a_{ij} \ge t_i - (\max_k R_{ik})(1 - \beta_{ij}), && \forall i, \forall j,\\
& (\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\beta}) \in P,\\
& \beta_{ij} \in \{0, 1\}, && \forall i, j.
\end{aligned}
$$

3.2 Alternative Formulation

Analogous to the alternate formulation of the Uniform Distribution Model in Section 2.2, the Weighted Uniform Model has an alternate formulation using the variables x_{ij}:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} N_i t_i, && && (3.2)\\
\text{s.t.} \quad & \textstyle\sum_{i=0}^{n} x_{ij} = 1, && \forall j,\\
& \textstyle\sum_{j=1}^{m} R_{ij} a_{ij} \le \sum_{j=1}^{m} \sum_{l: R_{lj} \le R_{ij}} R_{ij} R_{lj} x_{lj}, && \forall i,\\
& a_{ij} \le t_i, && \forall i, j,\\
& a_{ij} \le (\max_k R_{ik}) \textstyle\sum_{l: R_{lj} \le R_{ij}} x_{lj}, && \forall i, j,\\
& a_{ij} \ge t_i - (\max_k R_{ik}) \textstyle\sum_{l: R_{lj} > R_{ij}} x_{lj}, && \forall i, j,\\
& t_i \le (\max_k R_{ik}) \textstyle\sum_{j=1}^{m} \sum_{l: R_{lj} \le R_{ij}} x_{lj}, && \forall i,\\
& t_i \ge 0, && \forall l, i,\\
& a_{ij} \ge 0, && \forall i, j,\\
& x_{ij} \in \{0, 1\}, && \forall i, j.
\end{aligned}
$$

Similar to the Uniform Distribution Model, this alternate formulation results in a stronger integer programming formulation. Chapter 9 illustrates the running time of the Weighted Uniform Model on various problem sizes.

Chapter 4

Share-of-Surplus Model

4.1 The Model

It seems realistic to assume that the probability of a customer buying a product is related to the surplus. A similar scenario is when a customer prefers buying the product that has the most discount at the moment, rather than picking a product randomly or preferring the product with the highest reservation price. We want a model such that a larger surplus means that a larger fraction of a customer segment buys that product. That is, the probability that a customer buys a product depends on the customer's reservation price as well as the price of the product. A monotonically increasing function is needed to describe the relationship between the probability and the surplus.

The *Share-of-Surplus Choice Model* [6] is a form of a probabilistic choice model where the probability that a segment will choose a product is the ratio of its surplus versus the total surplus for the segment across all products with nonnegative surplus. Again, let

$$\beta_{ij} := \begin{cases} 1, & \text{if the surplus of product } j \text{ is nonnegative for segment } i,\ j, \\ 0, & \text{otherwise.} \end{cases}$$

In this model, the probability that segment i will buy product j is given by:

$$Pr_{ij} := \frac{(R_{ij} - \pi_j)\beta_{ij}}{\sum_k (R_{ik} - \pi_k)\beta_{ik}}.$$

For the moment, let us assume that $\sum_k (R_{ik} - \pi_k)\beta_{ik} > 0$ for all $i = 1, \dots, n$ for notational simplicity. We will relax this assumption in Section 4.4.

With the above definition, $Pr_{ij} = 0$ if $R_{ij} = \pi_j$, which may not be desirable. To ensure that the probability Pr_{ij} is strictly positive when $R_{ij} = \pi_j$, we may define the probability as follows:

$$Pr^*_{ij} := \frac{(R_{ij} - \pi_j + c)\beta_{ij}}{\sum_k (R_{ik} - \pi_k + c)\beta_{ik}}. \tag{4.1}$$

where c is a small positive constant. For simplicity's sake, we will use the first definition of the probability throughout the rest of this chapter. Note that this differs from the standard MNL model since we do not consider negative surplus products.

The expected revenue given by this model is

$$\sum_{i=1}^{n} \sum_{j=1}^{m} N_i \pi_j \left(\frac{(R_{ij} - \pi_j)\beta_{ij}}{\sum_k (R_{ik} - \pi_k)\beta_{ik}} \right).$$

We can model this Share-of-Surplus Choice Model as the following nonlinear mixed-integer programming model:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} N_i \pi_j \left(\frac{(R_{ij} - \pi_j)\beta_{ij}}{\sum_k (R_{ik} - \pi_k)\beta_{ik}} \right) & \qquad (4.2) \\ \text{s.t.} \quad & (\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{p}) \in P, & \\ & \beta_{ij} \in \{0, 1\}, \qquad i = 1, \dots, n; j = 1, \dots, m, & \end{aligned}$$

where P is the polyhedron defined in Section 1.2.

The objective function can further be reformulated to a sum of ratios, where the numerator is a concave quadratic and the denominator is linear:

$$\max \sum_{i=1}^{n} \sum_{j=1}^{m} N_i \pi_j \left(\frac{(R_{ij} - \pi_j)\beta_{ij}}{\sum_k (R_{ik} - \pi_k)\beta_{ik}} \right) \quad \Leftrightarrow \quad \max \sum_{i=1}^{n} \sum_{j=1}^{m} N_i \left(\frac{R_{ij} p_{ij} - p_{ij}^2}{\sum_k R_{ik}\beta_{ik} - p_{ik}} \right).$$

Thus, Model (4.2) can be formulated as the following mixed-integer fractional programming problem with linear constraints:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} N_i \left(\frac{R_{ij} p_{ij} - p_{ij}^2}{\sum_k R_{ik} \beta_{ik} - p_{ik}} \right), \\ \text{s.t.} \quad & (\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\beta}) \in P, \\ & \beta_{ij} \in \{0, 1\}, \qquad \forall i, j. \end{aligned} \tag{4.3}$$

An advantage of this model is the fact that it allocates some customers to each product whose price leaves a positive utility for the customer segment in a way that higher the positive utility of the product, higher the market share of the product.

In the following sections, we explore ways to solve Problem (4.3) to global optimality by finding convex relaxations or approximations of the model.

4.2 Quadratically Constrained Optimization Problem

Let us bring the objective function of Model (4.3) into the constraints and convert the problem to a minimization problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} N_i t_i \\ \text{s.t.} \quad & \sum_{j=1}^{m} \left(p_{ij}^2 - R_{ij} p_{ij} \right) \leq t_i \left(\sum_k (R_{ik} \beta_{ik} - p_{ik}) \right), \quad \forall i, \\ & (\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\beta}) \in P, \\ & \beta_{ij} \in \{0, 1\}, \qquad \forall i, j. \end{aligned}$$

Let us further introduce auxiliary variables x_i and y_i such that

$$\begin{aligned} x_i - y_i &:= t_i, \\ x_i + y_i &:= \sum_k (R_{ik} \beta_{ik} - p_{ik}). \end{aligned}$$

Then Model (4.3) can be written as

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} N_i t_i && (4.4)\\
\text{s.t.} \quad & \sqrt{\textstyle\sum_{j=1}^{m} \left(p_{ij}^2 - R_{ij}p_{ij}\right) + y_i^2} \le x_i, && \forall i,\\
& x_i - y_i = t_i, && \forall i,\\
& x_i + y_i = \textstyle\sum_k (R_{ik}\beta_{ik} - p_{ik}), && \forall i,\\
& (\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\beta}) \in P, &&\\
& \beta_{ij} \in \{0,1\}, && \forall i, j,
\end{aligned}
$$

whose continuous relaxation strongly resembles a second-order cone formulation. Unfortunately, the continuous relaxation of Model (4.4) does not result in a convex optimization problem.

Let us elaborate. Let a symmetric $n \times n$ matrix $\boldsymbol{A}$, an n-vector $\boldsymbol{a}$ and a constant α be given. Then the quadratic inequality

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} + 2\boldsymbol{a}^T \boldsymbol{x} + \alpha \le 0$$

is convex if and only if $\boldsymbol{A} \succeq \boldsymbol{O}$. We can represent this inequality as

$$\left[1, \boldsymbol{x}^T\right] \begin{bmatrix} \alpha & \boldsymbol{a}^T \\ \boldsymbol{a} & \boldsymbol{A} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix} \le 0.$$

To represent this inequality as a second order cone constraint, we need to find an $(n+1) \times (n+1)$ nonsingular matrix $\boldsymbol{L}$ such that

$$\boldsymbol{L}^T \begin{bmatrix} \alpha & \boldsymbol{a}^T \\ \boldsymbol{a} & \boldsymbol{A} \end{bmatrix} \boldsymbol{L} = \begin{bmatrix} -1 & \boldsymbol{0}^T \\ \boldsymbol{0} & I \end{bmatrix}.$$

Therefore, we conclude that the quadratic constraint is representable by a single Second Order Cone constraint if and only if the matrix

$$\begin{bmatrix} \alpha & \boldsymbol{a}^T \\ \boldsymbol{a} & \boldsymbol{A} \end{bmatrix}$$

has at most one negative eigenvalue. However, note that in our model (4.4) the corresponding matrix (ordered with respect to $1, t, \boldsymbol{\beta}, \boldsymbol{p}$):

$$\begin{bmatrix} 0 & 0 & \mathbf{0}^T & -\boldsymbol{R}^T \\ 0 & 0 & -\tilde{\boldsymbol{R}}^T & \boldsymbol{e}^T \\ \mathbf{0} & -\tilde{\boldsymbol{R}} & \boldsymbol{O} & \mathbf{0}^T \\ -\boldsymbol{R} & \boldsymbol{e} & \mathbf{0} & I \end{bmatrix}$$

has exactly two negative eigenvalues.

4.3 Second-Order Cone Approximation

As we showed above, we cannot expect convexity from this quadratic inequality as it stands. So let us consider some convex relaxations or approximations to Model (4.4).

Adding the term

$$\frac{\frac{1}{4}R_{ij}^2}{\sum_k (R_{ik} - \pi_k)\beta_{ik}}$$

to the objective function of Model (4.3) yields the following convex formulation:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} N_i t_i && (4.5) \\ \text{s.t.} \quad & \sqrt{\textstyle\sum_{j=1}^{m} \left(p_{ij} - \frac{1}{2}R_{ij}\right)^2 + y_i^2} \le x_i, && \forall i, \\ & x_i - y_i = t_i, && \forall i, \\ & x_i + y_i = \textstyle\sum_k (R_{ik}\beta_{ik} - p_{ik}), && \forall i, \\ & (\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\beta}) \in P, && \\ & \beta_{ij} \in \{0, 1\}, && \forall i, j, \end{aligned}$$

whose relaxation is a second-order cone programming problem. Unfortunately, empirical experiments showed that this approximation resulted in unrealistic prices. It seems that the price of each product are set to be much lower than the reservation prices of the customers who buy that product. However, with the β's fixed, we do not want the prices to be too low in practice since that generally result in lower revenue.

4.4 Restricted Prices

Unlike the Uniform and Weighted Uniform Models, the optimal price given β_{ij}'s is not immediate for the Share-of-Surplus model. Define $B_i = \{j|\beta_{ij} = 1\}$. Then the optimal prices is the solution to

$$\begin{array}{lll} \max & \displaystyle\sum_{i=1}^{n}\sum_{j\in B_i} N_i\pi_j\left(\frac{(R_{ij}-\pi_j)}{\sum_{k\in B_i}(R_{ik}-\pi_k)}\right) & \\ \text{s.t.} & R_{ij}-\pi_j \geq 0, & \forall i, j \in B_i, \\ & R_{ij}-\pi_j < 0, & \forall i, j \notin B_i, \\ & \pi_j \geq 0, & \forall j. \end{array} \tag{4.6}$$

If β_{ij} equals one for at least one segment, then we know that

$$\pi_j \in (\max_{i:\beta_{ij}=0} R_{ij}, \min_{i:\beta_{ij}=1} R_{ij}].$$

Suppose product l is bought by at least one segment and its price is increased by $\epsilon > 0$ such that β_{ij}'s do not change. Define $S_j = \{i|\beta_{ij} = 1\}$. Then the change in the objective value is:

$$\sum_{i\in S_l} N_i\left(\frac{\sum_{j\in B_i\setminus\{l\}}\pi_j(R_{ij}-\pi_j)+(\pi_l+\epsilon)(R_{il}-(\pi_l+\epsilon))}{(\sum_{k\in B_i\setminus\{l\}}(R_{ik}-\pi_k))+(R_{il}-(\pi_l+\epsilon))} - \frac{\sum_{j\in B_i}\pi_j(R_{ij}-\pi_j)}{\sum_{k\in B_i}(R_{ik}-\pi_k)}\right)$$

$$= \sum_{i\in S_l} N_i\left(\frac{\sum_{j\in B_i}\pi_j(R_{ij}-\pi_j)+\epsilon(R_{il}-2\pi_l-\epsilon)}{\sum_{k\in B_i}(R_{ik}-\pi_k)-\epsilon} - \frac{\sum_{j\in B_i}\pi_j(R_{ij}-\pi_j)}{\sum_{k\in B_i}(R_{ik}-\pi_k)}\right)$$

$$= \sum_{i\in S_l} N_i\left(\frac{\epsilon(R_{il}-2\pi_l-\epsilon)(\sum_{k\in B_i}(R_{ik}-\pi_k))+\epsilon\sum_{j\in B_i}\pi_j(R_{ij}-\pi_j)}{(\sum_{k\in B_i}(R_{ik}-\pi_k))(\sum_{k\in B_i}(R_{ik}-\pi_k)-\epsilon)}\right)$$

$$= \sum_{i\in S_l} \epsilon N_i\left(\frac{(R_{il}-(\pi_l+\epsilon))\sum_{j\in B_i}(R_{ij}-\pi_j)+\sum_{j\in B_i}(\pi_j-\pi_l)(R_{ij}-\pi_j)}{(\sum_{k\in B_i}(R_{ik}-\pi_k))(\sum_{k\in B_i}(R_{ik}-\pi_k)-\epsilon)}\right) \tag{4.7}$$

Increasing the price of product l by ϵ would result in an increased objective value if (4.7) is positive. The β_{ij}'s do not change after the price increase, which implies that $R_{il} \geq \pi_l + \epsilon$.

Therefore, all the terms in (4.7) are nonnegative except perhaps $(\pi_j - \pi_l)$. Thus, we can expect (4.7) to be positive if π_l is relatively low compared to other prices. Intuitively, this means that if π_l is low enough relative to other prices, then we want to raise π_l so that the surplus of product j decreases, hence decreasing the probability that the customers will buy this low-priced product. On the other hand, if π_l is high enough relative to other prices, we want to decrease π_l so that the probability that the customers will buy this expensive product increases, thus generating more revenue.

Suppose we restrict π_j to be equal to $\min_{i:\beta_{ij}=1} R_{ij}$, just as in the Uniform Distribution and Weighted Uniform Models. Then the Share-of-Surplus Model can be modeled as a mixed-integer linear programming model.

Again, let x_{ij} equal 1 if segment i has the smallest reservation price out of all segments with positive surplus for product j, 0 otherwise. Again, we introduce a dummy segment 0 with $R_{0j} > \max_{i=1,\ldots,n} R_{ij}, \forall j$, $N_0 = 0$ and add the constraint $\sum_{i=0}^{n} x_{ij} = 1$. As before, $\beta_{ij} = \sum_{l:R_{lj} \leq R_{ij}} x_{ij}$ and let us restrict π_j to equal $\sum_i R_{ij}x_{ij}$. Then the objective function of the Share-of-Surplus Model is:

$$
\begin{aligned}
\sum_{i=0}^{n}\sum_{j=1}^{m} N_i\pi_j\left(\frac{(R_{ij}-\pi_j)\beta_{ij}}{\sum_k (R_{ik}-\pi_k)\beta_{ik}}\right) &= \sum_{i=0}^{n} N_i \sum_{j=1}^{m}\pi_j\left(\frac{(R_{ij}-\pi_j)\sum_{l:R_{lj}\leq R_{ij}} x_{lj}}{\sum_{k=1}^{m}((R_{ik}-\pi_k)\sum_{l:R_{lk}\leq R_{ik}} x_{lk})}\right) \\
&= \sum_i N_i \sum_j \left(\sum_i R_{ij}x_{ij}\right)\left(\frac{(R_{ij}-\sum_s R_{sj}x_{sj})\sum_{l:R_{lj}\leq R_{ij}} x_{lj}}{\sum_k((R_{ik}-\sum_r R_{rj}x_{rj})\sum_{l:R_{lk}\leq R_{ik}} x_{lk})}\right) \\
&= \sum_i N_i \sum_j \left(\sum_i R_{ij}x_{ij}\right)\left(\frac{R_{ij}\sum_{l:R_{lj}\leq R_{ij}} x_{lj} - \sum_{l:R_{lj}\leq R_{ij}} R_{lj}x_{lj}}{\sum_k(R_{ik}\sum_{l:R_{lk}\leq R_{ik}} x_{lk} - \sum_{l:R_{lk}\leq R_{ik}} R_{lk}x_{lk})}\right) \\
&= \sum_i N_i \left(\frac{\sum_j (R_{ij}\sum_{l:R_{lj}\leq R_{ij}} R_{lj}x_{lj} - \sum_{l:R_{lj}\leq R_{ij}} R_{lj}^2 x_{lj})}{\sum_k(R_{ik}\sum_{l:R_{lk}\leq R_{ik}} x_{lk} - \sum_{l:R_{lk}\leq R_{ik}} R_{lk}x_{lk})}\right) \\
&= \sum_i N_i \left(\frac{\sum_j \sum_{l:R_{lj}\leq R_{ij}} R_{lj}(R_{ij}-R_{lj})x_{lj}}{\sum_k \sum_{l:R_{lk}\leq R_{ik}} (R_{ik}-R_{lk})x_{lk}}\right)
\end{aligned}
$$

Let us now relax the assumption that the denominator $\sum_{k=1}^{m}(R_{ik} - \pi_k)\beta_{ik} > 0$ or

$\sum_{k=1}^{m}\sum_{l:R_{lk}\leq R_{ik}}(R_{ik}-R_{lk})x_{lk}>0$ for all i. Define:

$$t_i := \begin{cases} \frac{\sum_j \sum_{l:R_{lj}\leq R_{ij}} R_{lj}(R_{ij}-R_{lj})x_{lj}}{\sum_k \sum_{l:R_{lk}\leq R_{ik}}(R_{ik}-R_{lk})x_{lk}}, & \text{if } \sum_k \sum_{l:R_{lk}\leq R_{ik}}(R_{ik}-R_{lk})x_{lk}\neq 0, \\ 0, & \text{otherwise.} \end{cases}$$

Let us introduce an auxiliary continuous variable u_{lij} where $u_{lij} := t_i x_{lj}$ for all segments l, i and products j where $R_{lj}\leq R_{ij}$. Then we can formulate the problem as a linear mixed-integer programming problem:

$$\begin{array}{lcl}
\max & \displaystyle\sum_{i=1}^{n} N_i t_i, & (4.8) \\
\text{s.t.} & \sum_{i=0}^{n} x_{ij}=1, & \forall j, \\
& \sum_{j=1}^{m}\sum_{l:R_{lj}\leq R_{ij}}(R_{ij}-R_{lj})u_{lij} \leq \sum_{j=1}^{m}\sum_{l:R_{lj}\leq R_{ij}} R_{lj}(R_{ij}-R_{lj})x_{lj}, & \forall i, \\
& u_{lij}\leq t_i, & \forall l,i,j,R_{lj}\leq R_{ij}, \\
& u_{lij}\leq (\max_k R_{ik})x_{lj}, & \forall l,i,j,R_{lj}\leq R_{ij}, \\
& u_{lij}\geq t_i-(\max_k R_{ik})(1-x_{lj}), & \forall l,i,j,R_{lj}\leq R_{ij}, \\
& t_i\leq (\max_k R_{ik})\sum_{j=1}^{m}\sum_{l:R_{lj}\leq R_{ij}}(R_{ij}-R_{lj})x_{lj}, & \forall i, \\
& t_i\geq 0, & \forall l,i, \\
& u_{lij}\geq 0, & \forall l,i,j,R_{lj}\leq R_{ij}, \\
& x_{ij}\in\{0,1\}, & \forall i,j.
\end{array}$$

If we use the probability Pr^*_{ij} (4.1) instead, then the objective function is:

$$\sum_i N_i \left(\frac{\sum_j \sum_{l:R_{lj}\leq R_{ij}} R_{lj}(R_{ij}-R_{lj}+c)x_{lj}}{\sum_k \sum_{l:R_{lk}\leq R_{ik}}(R_{ik}-R_{lk}+c)x_{lk}}\right)$$

Then the problem can be formulated as follows:

$$\max \quad \sum_{i=1}^{n} N_i t_i, \tag{4.9}$$

$$\begin{array}{lll}
\text{s.t.} & \sum_{j=1}^{m} \sum_{l:R_{lj} \le R_{ij}} (R_{ij} - R_{lj} + c) u_{lij} \le \sum_{j=1}^{m} \sum_{l:R_{lj} \le R_{ij}} R_{lj}(R_{ij} - R_{lj} + c) x_{lj}, & \forall i, \\
& \sum_{i=0}^{n} x_{ij} = 1, & \forall j, \\
& u_{lij} \le t_i, & \forall l, i, j, R_{lj} \le R_{ij}, \\
& u_{lij} \le (\max_k R_{ik}) x_{lj}, & \forall l, i, j, R_{lj} \le R_{ij}, \\
& u_{lij} \ge t_i - (\max_k R_{ik})(1 - x_{lj}), & \forall l, i, j, R_{lj} \le R_{ij}, \\
& t_i \le (\max_k R_{ik}) \sum_{j=1}^{m} \sum_{l:R_{lj} \le R_{ij}} (R_{ij} - R_{lj} + c) x_{lj}, & \forall i, \\
& t_i \ge 0, & \forall l, i, \\
& u_{lij} \ge 0, & \forall l, i, j, R_{lj} \le R_{ij}, \\
& x_{ij} \in \{0, 1\}, & \forall i, j.
\end{array}$$

If $c < 1$, then we need to replace the constraint

$$t_i \le (\max_k R_{ik}) \sum_{j=1}^{m} \sum_{l:R_{lj} \le R_{ij}} (R_{ij} - R_{lj} + c) x_{lj}, \forall i$$

by

$$t_i \le \frac{1}{c} (\max_k R_{ik}) \sum_{j=1}^{m} \sum_{l:R_{lj} \le R_{ij}} (R_{ij} - R_{lj} + c) x_{lj}, \forall i$$

so that the right-hand-side is $\ge (\max_k R_{ik})$ whenever the summation is non-zero.

The constant c used in the formulation is assumed to be small enough such that the difference between Pr^*_{ij} and Pr_{ij} is almost negligible but that the probability is positive when the surplus is nonnegative. In future work, we may like to examine the effect of the value of c on the problem and determine the ideal value for the constant.

From experiments, we found that the Share-of-Surplus Model is hard to solve. We would like to explore other ways to formulate it or perhaps find different cuts in order to decrease the solution time. We may also want to investigate other monotonically increasing functions to describe the probability which would perhaps lead to formulations that are easier to solve. The experimental results are discussed further in Chapter 9.

Chapter 5

Price Sensitive Model

A common economic assumption is that as the price of a product decreases, the demand increases. In this chapter, we discuss a probabilistic choice model where the probability of a customer buying a particular product with nonnegative surplus is inversely proportional to the price of the product. Unlike the models discussed in the previous chapters, the probability Pr_{ij} depends only on π_j.

5.1 The Model

Again, let p_{ij} be the auxiliary variable where $p_{ij} := \pi_j \beta_{ij}$. Consider the probability of customer segment i buying product j as defined below:

$$Pr_{ij} := \begin{cases} 0, & \text{if } \beta_{ij} = 0 \text{ (Case 0)}, \\ 1, & \text{if } \beta_{ij} = 1, \sum_k \beta_{ik} = 1 \text{ (Case 1)}, \\ \frac{1}{\sum_k \beta_{ik} - 1} \left(\beta_{ij} - \frac{p_{ij}}{\sum_k p_{ik}} \right), & \text{otherwise (Case 2)}. \end{cases}$$

In this model, $Pr_{ij} = 0$ if product j has a negative surplus for segment i (Case 0), $Pr_{ij} = 1$ if product j is the only product with nonnegative surplus (Case 1), and if there are multiple products with nonnegative surplus (Case 2), Pr_{ij} is inversely proportional to the price of those products. Thus, we call this model the *Price Sensitive Model*. With some reformulation, the expected revenue maximization problem corresponding to this model can be formulated as a second-order cone problem with integer variables.

If Rev_{ij} is the expected revenue earned from segment i for product j, we want $Rev_{ij} = 0$ in Case 0, $Rev_{ij} = \pi_j$ in Case 1, and $Rev_{ij} = \frac{\pi_j}{\sum_k \beta_{ik} - 1}\left(\beta_{ij} - \frac{p_{ij}}{\sum_k p_{ik}}\right)$ in Case 2. Let z_i be a binary decision variable where $z_i = 1$ if segment i buys exactly one product and $z_i = 0$ otherwise. The expected revenue Rev_{ij} earned from segment i buying product j can be modeled as:

$$Rev_{ij} := \begin{cases} 0, & \text{if } \sum_j p_{ij} = 0, \\ \frac{\pi_j}{\sum_k \beta_{ik} - 1 + z_i}\left(\beta_{ij} - \frac{p_{ij}}{\sum_k p_{ik}}\right) + \pi_j z_i, & \text{otherwise.} \end{cases}$$

Then the expected revenue from segment i, Rev_i, is $Rev_i = \sum_{j=1}^m Rev_{ij}$ or

$$Rev_i := \begin{cases} 0, & \text{if } \sum_j p_{ij} = 0, \\ \left(\frac{\sum_j p_{ij}}{\sum_j \beta_{ij} - 1 + z_i} - \frac{\sum_j p_{ij}^2}{(\sum_j \beta_{ij} - 1 + z_i)(\sum_k p_{ik})}\right) + (\sum_j p_{ij}) z_i, & \text{otherwise.} \end{cases}$$

Let s_i be an auxiliary variable where $s_i := (\sum_j p_{ij}) z_i$, which we know is a relationship that can be modeled by linear constraints. Also let

$$t_i := \begin{cases} 0, & \text{if } \sum_j p_{ij} = 0, \\ \frac{\sum_j p_{ij}}{\sum_j \beta_{ij} - 1 + z_i} - \frac{\sum_j p_{ij}^2}{(\sum_j \beta_{ij} - 1 + z_i)(\sum_k p_{ik})}, & \text{otherwise.} \end{cases}$$

Then the expected revenue maximization problem corresponding to the Price Sensitive Model is:

$$\begin{aligned}
\max \quad & \sum_{i=1}^n N_i t_i + \sum_{i=1}^n N_i s_i, && (5.1) \\
\text{s.t.} \quad & \textstyle\sum_j p_{ij}^2 \le (\sum_j p_{ij})^2 - t_i(\sum_j \beta_{ij} - 1 + z_i)(\sum_j p_{ij}), && \forall i, \\
& t_i \le \textstyle\sum_j p_{ij}, && \forall i, \\
& s_i \le \textstyle\sum_j p_{ij}, && \forall i, \\
& s_i \le \textstyle\sum_j R_{ij} z_i, && \forall i, \\
& \textstyle\sum_j \beta_{ij} \le z_i + m(1 - z_i), && \forall i, \\
& z_i \ge \beta_{ij} - \textstyle\sum_{k \ne j} \beta_{ik}, && \forall i, \forall j, \\
& (\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\beta}) \in P, \\
& \beta_{ij} \in \{0, 1\}, && \forall i, j, \\
& z_i \in \{0, 1\}, s_i \ge 0, && \forall i, j,
\end{aligned}$$

where P is the polyhedron defined in Section 1.2.

We need to reformulate the first set of constraints to make the continuous relaxation of (5.1) a convex programming problem. Let us look at the first set of constraints:

$$\sum_j p_{ij}^2 \leq (\sum_j p_{ij})^2 - t_i(\sum_j \beta_{ij} - 1 + z_i)(\sum_j p_{ij}), \quad \forall i. \tag{5.2}$$

However, if $t_i > 0$ then $z_i = 0$ and if $z_i = 1$ then $t_i = 0$. Thus, we can eliminate the z_i term from the above inequality if we include the constraint

$$t_i \leq \max_k R_{ik}(1 - z_i).$$

Also, let b_{ij} be auxiliary variables where $b_{ij} := t_i\beta_{ij}$. Again, such relations can be modeled by linear constraints. Then, (5.2) becomes

$$\sum_j p_{ij}^2 \leq (\sum_j p_{ij})(\sum_j p_{ij} - \sum_j b_{ij} + t_i), \quad \forall i.$$

Let us further introduce auxiliary variables x_i and y_i such that:

$$x_i + y_i = \sum_j p_{ij} - \sum_j b_{ij} + t_i, \qquad \forall i,$$

$$x_i - y_i = \sum_j p_{ij}, \qquad \forall i.$$

Thus, the constraint becomes

$$\sum_j p_{ij}^2 \leq (x_i + y_i)(x_i - y_i) = x_i^2 - y_i^2$$

Then (5.2) can be represented by the second-order cone and linear constraints shown below:

$$\begin{aligned}
\sqrt{\textstyle\sum_j p_{ij}^2 + y_i^2} &\leq x_i, && \forall i, \\
x_i + y_i &= \textstyle\sum_j p_{ij} - \sum_j b_{ij} + t_i, && \forall i, \\
x_i - y_i &= \textstyle\sum_j p_{ij}, && \forall i, \\
t_i &\leq \tilde{R}_i(1 - z_i), && \forall i, \forall j, \\
b_{ij} &\leq \tilde{R}_i\beta_{ij}, && \forall i, \\
b_{ij} &\leq t_i, && \forall i, \forall j, \\
b_{ij} &\geq t_i - \tilde{R}_i(1 - \beta_{ij}), && \forall i, \forall j,
\end{aligned} \tag{5.3}$$

where $\tilde{R}_i := \max_j R_{ij}$.

The formulation of the Price Sensitive Model becomes:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} N_i t_i + \sum_{i=1}^{n} N_i s_i, && \\
\text{s.t.} \quad & \sqrt{\textstyle\sum_j p_{ij}^2 + y_i^2} \leq x_i, && \forall i, \\
& x_i + y_i = \textstyle\sum_j p_{ij} - \sum_j b_{ij} + t_i, && \forall i, \\
& x_i - y_i = \textstyle\sum_j p_{ij}, && \forall i, \\
& t_i \leq \tilde{R}_i(1 - z_i), && \forall i, \forall j, \\
& b_{ij} \leq \tilde{R}_i \beta_{ij}, && \forall i, \\
& b_{ij} \leq t_i, && \forall i, \forall j, \\
& b_{ij} \geq t_i - \tilde{R}_i(1 - \beta_{ij}), && \forall i, \forall j, \\
& t_i \leq \textstyle\sum_j p_{ij}, && \forall i, \\
& s_i \leq \textstyle\sum_j p_{ij}, && \forall i, \\
& s_i \leq \textstyle\sum_j R_{ij} z_i, && \forall i, \\
& \textstyle\sum_j \beta_{ij} \leq z_i + m(1 - z_i), && \forall i, \\
& z_i \geq \beta_{ij} - \textstyle\sum_{k \neq j} \beta_{ik}, && \forall i, \forall j, \\
& (\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\beta}) \in P, && \\
& \beta_{ij} \in \{0, 1\}, && \forall i, j, \\
& z_i \in \{0, 1\}, s_i \geq 0, b_{ij} \geq 0, && \forall i, j.
\end{aligned}
\tag{5.4}
$$

5.2 Computational Results

Table 5.1 shows some computational results of running small problem instances with the Price Sensitive Model formulation (5.4). The first ten cases (t*) each has 3 products and 3 segments. The next six cases (rand*) each has 5 products and 5 segments and the reservation prices are random numbers that range from 500 to 1200. The rest of the cases are subsets of real data and the file name ($n \times m$) indicates the number of segments and the number of products, respectively, in the inputs. The model was run with default parameter settings of CPLEX 9.1 and a time limit of two hours (7200 CPU seconds).

For each case, Table 5.1 shows the objective value ("Objective Value"), total CPU seconds required to find a provable optimal solution ("Time"), total number of dual simplex iterations ("Number of Iterations"), total number of branch-and-bound nodes ("Number of Nodes"), total number of branch-and-bound nodes unvisited ("Number of Nodes Left"), and the optimality gap when CPLEX was terminated ("Gap").

Since the formulation has a second-order cone constraint, only small problems can be solved quickly. The smaller cases can be solved to optimality fairly quickly, but the solutions for the last two cases ("10×10" and "10×20") found by CPLEX after 2 hours have large optimality gaps.

File	CPLEX Status	Objective Value	Time	Number of Iterations	Number of Nodes	Number of Nodes Left	Gap
t1	Optimal	14.47	0.61	861	106	0	0
t2	Optimal	18.46	0.74	616	54	0	0
t3	Optimal	16.00	0.53	453	48	0	0
t4	Optimal	12.95	1.00	980	86	0	0
t5	Optimal	14.11	0.69	716	84	0	0
t6	Optimal	90.00	0.76	1023	98	0	0
t7	Optimal	9.09	0.67	579	46	0	0
t8	Optimal	12.95	1.01	980	86	0	0
t9	Optimal	92.00	0.78	940	86	0	0
t10	Optimal	21.00	0.61	507	52	0	0
rand1	Optimal	4277.49	21.30	20625	1682	0	0
rand2	Optimal	113443274.07	14.47	12025	735	3	0
rand3	Optimal	100887782.67	9.67	7254	498	42	0
rand4	Optimal	113332142.77	6.34	4899	316	0	0
rand5	Optimal	110221028.53	10.08	7443	474	5	0
rand6	Optimal	101665597.25	12.85	11705	873	13	0
2x2	Optimal	2656.00	0.01	8	0	1	0
2x5	Optimal	121519.94	0.59	768	56	0	0
5x2	Optimal	200078.66	0.54	785	81	0	0
5x5	Optimal	163817.58	23.62	25662	2151	0	0
5x10	Optimal	217195.42	4232.98	2685098	182506	15	0.01
10x5	Optimal	324163.32	1331.47	970831	79289	1	0.01
10x10	Feasible	381825.37	7247.06	1655571	105230	58508	71.96
10x20	Feasible	553040.97	7248.08	168276	7984	5627	85.77

Table 5.1: Price Sensitive Model

Chapter 6

Properties of the Models

In this chapter, we discuss properties of the optimal solutions of our models for specialized data sets. We also compare the optimal prices π_j and variables β_{ij} for all the models on different sets of reservation prices.

6.1 Special Properties

Lemma 6.1.1. *Suppose $n \leq m$, and for every segment i, we can find a unique product $p(i)$ such that $R_{ip(i)} = \max_j R_{ij}$. Further suppose that for each of such product $p(i)$, segment i is the unique segment such that $R_{ip(i)} = max_k R_{kp(i)}$.*

Let $J := \{j | j = p(i) \text{ for some segment } i \neq 0\}$.

Then in the optimal solution,

$$\beta_{ij} := \begin{cases} 1, & \text{if } j = p(i), \\ 0, & \text{otherwise.} \end{cases}$$

In the alternative formulations, the optimal solution is

$$x_{ij} := \begin{cases} 1, & \text{if } j = p(i), \\ 1, & \text{if } i = 0 \text{ and } j \notin J, \\ 0, & \text{otherwise.} \end{cases}$$

where segment 0 is the dummy segment.

Proof. The maximum revenue we can get from segment i is $N_i(\max_j R_{ij}) = N_i R_{ip(i)}$. This happens when segment i only buys product $p(i)$. Hence, the objective value of any feasible solution is at most $\sum_{i=1}^n N_i R_{ip(i)}$.

Consider the solution with the x variables assigned as in the lemma and

$$\pi_j := \begin{cases} \max_{k=1,\dots,n} R_{kj}, & \text{if } j \in J, \\ R_{0j}, & \text{otherwise.} \end{cases}$$

Because of the assumptions in the Lemma, the solution is feasible with exactly one segment with nonnegative surplus for each product $j \in J$ and no segment buying any products $j \notin J$. That implies every segment only buys the product with the maximum reservation price. The corresponding objective value is $\sum_{i=1}^n N_i R_{ip(i)}$, and thus the solution is optimal. □

The following lemmas apply to the Uniform Distribution Model (Chapter 2), the Weighted Uniform Model (Chapter 3), and the Share-of-Surplus Model with restriced prices (Section 4.4).

Lemma 6.1.2. *If the optimal values for the x (or β) variables are known, then the optimal prices can be determined. Furthermore, if the optimal prices are known, then the optimal values for the x (or β) variables can be determined.*

Proof. If the x variables are known, then $\pi_j = R_{ij}$ where i is the segment such that $x_{ij} = 1$. If the β variables are known, then $\pi_j = \min_{i:\beta_{ij}=1} R_{ij}$.

If the optimal prices are known, we know that each π_j equals the reservation price of some segment. Then in the optimal solution,

$$x_{ij} := \begin{cases} 1, & \text{if } R_{ij} = \pi_j, \\ 0, & \text{otherwise.} \end{cases}$$

and

$$\beta_{ij} := \begin{cases} 1, & \text{if } R_{ij} \geq \pi_j, \\ 0, & \text{otherwise.} \end{cases}$$

These are the only values that would make the solution feasible.

□

Lemma 6.1.3. *Suppose R_{st} is the maximum reservation price over all segments and products and only one pair of segment and product has that reservation price. Then in any optimal solution, segment s buys product t.*

Proof. Suppose in an optimal solution, $\beta_{st} = 0$. We know that $\beta_{it} = 0$ for all segments i in all three models. Let v be the optimal value. Consider the objective value v' if β_{st} is set to 1. We will have $\pi_t = R_{st}$.

In the Uniform Distribution Model, if $\sum_j \beta_{sj} = 0$, then clearly the objective value increases by $N_s R_{st}$. If $\sum_j \beta_{sj} \geq 1$, then

$$v' - v = N_s \left(\frac{R_{st} + \sum_j p_{sj}}{1 + \sum_j \beta_{sj}} - \frac{\sum_j p_{sj}}{\sum_j \beta_{sj}} \right)$$

$$= N_s \left(\frac{R_{st} \sum_j \beta_{sj} - \sum_j p_{sj}}{(\sum_j \beta_{sj})(1 + \sum_j \beta_{sj})} \right)$$

$$= N_s \left(\frac{R_{st} \sum_j \beta_{sj} - \sum_j \pi_j \beta_{sj}}{(\sum_j \beta_{sj})(1 + \sum_j \beta_{sj})} \right)$$

$$= N_s \left(\frac{\sum_j (R_{st} - \pi_j) \beta_{sj}}{(\sum_j \beta_{sj})(1 + \sum_j \beta_{sj})} \right) \tag{6.1}$$

R_{st} is the maximum reservation price and each of the π_j's equals to a reservation price, so $R_{st} \geq \pi_j \quad \forall j$. The condition $\sum_j \beta_{sj} \geq 1$ implies that $\beta_{sk} = 1$ for some product $k \neq t$, and we know that $R_{st} > \pi_k$. So the expression (6.1) is strictly positive. This contradicts the fact that it is an optimal solution. Therefore, $\beta_{st} \geq 1$ in an optimal solution.

Similarly, in the Weighted Uniform Model, if $\sum_j \beta_{sj} = 0$, then clearly the objective value increases by $N_s R_{st}$. If $\sum_j \beta_{sj} = 1$, then

$$v' - v = N_s \left(\frac{R_{st}\pi_t + \sum_j R_{sj}\pi_j\beta_{sj}}{R_{st} + \sum_j R_{sj}\beta_{sj}} - \frac{\sum_j R_{sj}\pi_j\beta_{sj}}{\sum_j R_{sj}\beta_{sj}} \right)$$

$$= N_s R_{st} \left(\frac{\pi_t \sum_j R_{sj}\beta_{sj} - \sum_j R_{sj}\pi_j\beta_{sj}}{(\sum_j R_{sj}\beta_{sj})(R_{st} + \sum_j R_{sj}\beta_{sj})} \right)$$

$$= N_s R_{st} \left(\frac{\sum_j (\pi_t - \pi_j) R_{sj} \beta_{sj}}{(\sum_j R_{sj}\beta_{sj})(R_{st} + \sum_j R_{sj}\beta_{sj})} \right)$$
$$> 0$$

where $\pi_t = R_{st} > \pi_j \quad \forall j$.

In the Share-of-Surplus Model with restricted prices, if $\sum_j \beta_{sj} = 0$, then the objective value increases by $N_s R_{st}$. Otherwise,

$$v' - v = N_s \left(\frac{\pi_t(R_{st} - \pi_t + c) + \sum_j \pi_j (R_{sj} - \pi_j + c)\beta_{sj}}{(R_{st} - \pi_t + c) + \sum_j (R_{sj} - \pi_j + c)\beta_{sj}} - \frac{\sum_j \pi_j (R_{sj} - \pi_j + c)\beta_{sj}}{\sum_j (R_{sj} - \pi_j + c)\beta_{sj}} \right)$$
$$= N_s (R_{st} - \pi_t + c) \left(\frac{\sum_j (\pi_t - \pi_j)(R_{sj} - \pi_j + c)\beta_{sj}}{(\sum_j (R_{sj} - \pi_j + c)\beta_{sj})((R_{st} - \pi_t + c) + \sum_j (R_{sj} - \pi_j + c)\beta_{sj})} \right)$$
$$> 0$$

where we let $c > 0$ to avoid singularity.

In all three models, we showed that the solution is not optimal if $\beta_{st} = 0$. So in any optimal solution, segment s buys product t. □

6.2 Comparisons

In this section, we compare the optimal solution, in terms of the prices π_j's and β_{ij}'s, of the different models. We notice in most examples, the four models have the same optimal solutions. Of the ones where they have different optimal solutions, usually the Uniform Distribution Model, the Weighted Uniform model, and the Price Sensitive model have the same optimal solution, while the Share-of-Surplus Model has a different optimal solution.

The optimal solutions of the models for four small test cases are shown to illustrate the differences in the models (Tables 6.1, 6.2, 6.3, and 6.4). Each table corresponds to a different set of reservations prices. The matrix $\boldsymbol{R}$ corresponds to the reservation prices where the rows correspond to the customer segments and the columns correspond to the products. The column "Price π" corresponds to the optimal prices, the "β" corresponds to the optimal β_{ij} and "Obj Value" corresponds to the optimal objective value.

The only difference between the inputs of Test 1 (Table 6.1) and Test 2 (Table 6.2) is R_{21}. All the models have the same optimal solution for Test 1, but the Share-of-Surplus Model has a different optimal solution from the other models in Test 2.

Let us consider why the Share-of-Surplus Model has a different optimal solution in Test 2. Clearly, $\beta_{11} = 1$ in an optimal solution in all the models (by Lemma 6.1.3). If we have $\beta_{12} = 1$ and $\beta_{21} = 1$, then we get more revenue from segment 2. In the Uniform, Weighted Uniform, and Price Sensitive models, this solution gives a higher objective value since $\pi_2 = R_{12}$ is quite high and the probability of segment 1 buying product 2, Pr_{12}, is high enough so that the decrease in revenue from segment 1 is small compared to the revenue from segment 2. Pr_{12} is approximately 0.5, 0.44, and 0.36 in the Uniform, Weighted Uniform, and Price Sensitive Models respectively. It is different with the Share-of-Surplus Model, however, because the surplus of segment 1 for product 1 ($R_{11}-R_{21} = 5$) is relatively high. The probability of segment 1 buying the lower priced product, Pr_{11}, is quite high at 0.86, so the decrease in revenue from segment 1 is greater than the gain in revenue from segment 2. Therefore, the optimal solution in the Share-of-Surplus Model is simply $\beta_{11} = 1$ and all other β's are zero.

Compared to Test 2, the surplus ($R_{11} - R_{21}$) is smaller in Test 1 and also $\pi_1 = R_{21}$ is higher. Therefore, with $\beta_{12} = 1$ and $\beta_{21} = 1$, the decrease in revenue from segment 1 (\$1.75) is smaller than the gain in revenue from segment 2 (\$7) in the Share-of-Surplus Model.

In Test 3 (Table 6.3), the Weighted Uniform Model has a different optimal solution than the other models. In the other three models, segments 1 and 2 only buy product 1, and segment 3 does not buy any products. This is because the reservation prices of segment 3 are relatively low. If segment 3 buys any product, the revenue from segment 1 and 2 will decrease significantly because of the lower prices and the decrease in revenue cannot be compensated by the extra revenue from segment 3. However, this is not the case in the Weighted Uniform Model. Recall that in the Weighted Uniform Model, the probability of segment i buying product j is proportional to R_{ij}. For both segments 1 and 2, the reservation prices for product 1 are much greater than the reservation prices for product 2. R_{11} and R_{21} are almost double R_{12} and R_{22}, respectively. Therefore, when the price of product 2 is 22, the probability of segments 1 and 2 buying product 1 at a

Test 1	$\boldsymbol{R} = \begin{bmatrix} 9 & 8 & 3 \\ 7 & 3 & 2 \\ 1 & 1 & 1 \end{bmatrix}$		
	Price π	β	Obj Value
Uniform	7 8 4	1 1 0 1 0 0 0 0 0	14.50
Share of Surplus	7 8 4	1 1 0 1 0 0 0 0 0	14.25
Weighted Uniform	7 8 4	1 1 0 1 0 0 0 0 0	14.47
Price Sensitive	7 8 4	1 1 0 1 0 0 0 0 0	14.47

Table 6.1: Optimal Prices π_j and β_{ij} of all four models.

high price is much greater than the probability of those segments buying product 2. The extra revenue from segment 3 overcompensates the small loss in revenue from the other segments.

Test 4 (Table 6.4) is another example in which the Share-of-Surplus Model has a different optimal solution as the other models.

We also compare the models' optimal solutions on random data with 5 segments and 5 products in which the reservation prices are uniformly generated from a specified range. The difference in the optimal prices are shown in Tables 6.5 and 6.6. We let 'U', 'W', 'S', and 'P' represent the Uniform, Weighted Uniform, Share-of-Surplus, and Price Sensitve Models respectively. For example, the column "U - W" shows the difference in the optimal prices of the Uniform Model and the Weighted Uniform Model. Suppose π_j^1 are the optimal prices of one model and π_j^2 are those of another model. Then the entry in the table is

Test 2	$\boldsymbol{R} = \begin{bmatrix} 9 & 8 & 3 \\ 4 & 3 & 2 \\ 1 & 1 & 1 \end{bmatrix}$		
	Price π	β	Obj Value
Uniform	4 8 4	1 1 0 1 0 0 0 0 0	10
Share of Surplus	9 9 4	1 0 0 0 0 0 0 0 0	9
Weighted Uniform	4 8 4	1 1 0 1 0 0 0 0 0	9.8824
Price Sensitive	4 8 4	1 1 0 1 0 0 0 0 0	9.3333

Table 6.2: Optimal Prices π_j and β_{ij} of all four models.

Test 3	$\boldsymbol{R} = \begin{bmatrix} 49 & 28 & 27 \\ 46 & 25 & 23 \\ 24 & 22 & 21 \end{bmatrix}$		
	Price π	β	Obj Value
Uniform	46 29 28	1 0 0 1 0 0 0 0 0	92
Share-of-Surplus	46 29 28	1 0 0 1 0 0 0 0 0	92
Weighted Uniform	46 22 28	1 1 0 1 1 0 0 1 0	96.822
Price Sensitive	46 29 28	1 0 0 1 0 0 0 0 0	92

Table 6.3: Optimal Prices π_j and β_{ij} of all four models.

Test 4	$\boldsymbol{R} = \begin{bmatrix} 889 & 1241 & 1015 & 1284 \\ 779 & 594 & 823 & 625 \\ 1425 & 1053 & 1018 & 1283 \\ 1112 & 652 & 1195 & 608 \end{bmatrix}$			
	Price π	β		Obj Value
Uniform	1112 1241 823 1283	0 1 1 1 0 0 1 0 1 0 1 1 1 0 1 0		3978.83
Share-of-Surplus	1425 1242 1195 1284	0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0		3904.00
Weighted Uniform	1112 1241 823 1283	0 1 1 1 0 0 1 0 1 0 1 1 1 0 1 0		4013.61
Price Sensitive[1]	1112 1241 823 1283	0 1 1 1 0 0 1 0 1 0 1 1 1 0 1 0		3921.13

Table 6.4: Optimal Prices π_j and β_{ij} of all four models.

$\sum_{j=1}^{m} |\pi_j^1 - \pi_j^2|$.

The Uniform and the Weighted Uniform Models have the same optimal prices for all these problem instances, probably because it is unlikely in the random data to have the reservation prices for one product to be much larger than those of another product as in Test 3 (Table 6.3). These two models have the same optimal prices as the Price Sensitive Model except in only 2 of the problem instances. The same optimal prices (hence, same optimal β's) imply that the Uniform Model may not be as naive as it seems since in most cases, it gives the same solutions as the two other more realistic models. However, the Share-of-Surplus Model appears to behave in a special way with results different from the other three models in more cases.

Tables 6.7 and 6.8 show the differences in the optimal values of each pair of the models. For example, the column "U - W" is the optimal value of the Uniform Model minus the optimal value of the Weighted Uniform Model. The differences in the optimal values of the Uniform, the Weighted Uniform, and the Price Sensitive Models are quite small in many problem instances, but the Share-of-Surplus Model gives smaller optimal values than the other three models in most cases (the columns "U - S," "W - S," and "P - S" have positive and relatively large entries). It is most likely because the probability for a segment to buy a lower-priced product is usually higher in the Share-of-Surplus Model than in the other three models.

[1]The actual optimal prices found by CPLEX are $(1111.999934, 1240.999815, 823.000164, 1282.999916)$

		Difference in Prices			
Range	Test #	U - W	U - S	U - P	S - P
1000-1100	1	0	0	0	0
	2	0	0	0	0
	3	0	0	0	0
	4	0	0	0	0
	5	0	1	0	1
	6	0	0	0	0
	7	0	4	1	3
	8	0	0	0	0
	9	0	0	0	0
	10	0	0	0	0
1000-1500	1	0	0	0	0
	2	0	126	0	126
	3	0	0	0	0
	4	0	35	0	35
	5	0	73	0	73
	6	0	356	0	356
	7	0	0	0	0
	8	0	45	0	45
	9	0	0	0	0
	10	0	0	0	0
1000-1700	1	0	0	0	0
	2	0	146	0	146
	3	0	11	0	11
	4	0	0	0	0
	5	0	389	389	0
	6	0	0	0	0
	7	0	0	0	0
	8	0	0	0	0
	9	0	0	0	0
	10	0	105	0	105

Table 6.5: Comparison of the Prices in the Models' Solutions for Random Tests (1)

		Difference in Prices			
Range	Test #	U - W	U - S	U - P	S - P
1000-2000	1	0	199	0	199
	2	0	257	0	257
	3	0	0	0	0
	4	0	0	0	0
	5	0	0	0	0
	6	0	0	0	0
	7	0	0	0	0
	8	0	768	0	768
	9	0	156	0	156
	10	0	0	0	0
1000-3000	1	0	10	0	10
	2	0	0	0	0
	3	0	1	1	0
	4	0	716	0	716
	5	0	0	0	0
	6	0	0	0	0
	7	0	137	0	137
	8	0	0	0	0
	9	0	0	0	0
	10	0	0	0	0

Table 6.6: Comparison of the Prices in the Models' Solutions for Random Tests (2)

		Difference in Objective Value					
Range	Test #	U - W	U - S	U - P	W - S	W - P	P - S
1000-1100	1	-0.01	1.00	0.01	1.01	0.02	0.99
	2	0	0	0	0	0	0
	3	0.07	45.09	0.49	45.02	0.42	44.60
	4	-0.04	4.66	0.02	4.70	0.06	4.64
	5	-0.05	12.18	0.24	12.23	0.29	11.93
	6	-0.38	21.90	0.28	22.28	0.66	21.62
	7	-0.10	32.63	0.66	32.73	0.76	31.97
	8	0	0	0	0	0	0
	9	0.07	9.39	0.14	9.32	0.07	9.25
	10	-0.10	24.17	0.18	24.27	0.28	23.99
1000-1500	1	-1.36	-20.63	0.62	-19.27	1.98	-21.26
	2	-5.17	132.60	10.96	137.77	16.13	121.65
	3	-0.31	44.17	1.65	44.48	1.96	42.53
	4	-0.22	9.00	1.42	9.22	1.64	7.58
	5	0.22	14.00	0.23	13.78	0.01	13.77
	6	-1.30	117.19	5.10	118.49	6.40	112.09
	7	0	0	0	0	0	0
	8	-7.00	112.90	8.77	119.90	15.77	104.13
	9	0	0	0	0	0	0
	10	-0.01	8.44	0.08	8.45	0.09	8.36
1000-1700	1	-2.05	73.47	3.61	75.52	5.66	69.86
	2	0.27	55.86	1.40	55.59	1.13	54.46
	3	0.22	15.50	0.44	15.28	0.22	15.06
	4	-0.12	-5.75	0.03	-5.63	0.15	-5.78
	5	-8.93	83.78	2.76	92.71	11.69	81.03
	6	0	0	0	0	0	0
	7	0	0	0	0	0	0
	8	-3.33	136.89	3.78	140.22	7.11	133.11
	9	0.50	30.60	0.65	30.10	0.15	29.94
	10	-4.74	113.11	7.19	117.85	11.93	105.92

Table 6.7: Comparison of the Objective Values for Random Tests (1)

		Difference in Objective Value					
Range	Test #	U - W	U - S	U - P	W - S	W - P	P - S
1000-2000	1	-9.94	201.44	19.48	211.38	29.42	181.96
	2	-19.04	487.80	83.00	506.84	102.04	404.80
	3	0	0	0.01	0	0.01	-0.01
	4	0	0	0	0	0	0
	5	0.52	18.24	0.20	17.72	-0.32	18.05
	6	0	0	0	0	0	0
	7	-4.44	113.49	6.21	117.93	10.65	107.28
	8	-2.11	185.00	23.07	187.11	25.18	161.93
	9	-1.93	18.43	1.12	20.36	3.05	17.31
	10	0	0	0.01	0	0.01	-0.01
1000-3000	1	-0.90	81.00	3.43	81.90	4.33	77.57
	2	-0.20	33.77	0.37	33.97	0.57	3.39
	3	-0.20	208.21	8.75	208.41	8.95	199.47
	4	-35.80	203.74	21.62	239.54	57.42	182.13
	5	-2.50	177.71	11.55	180.21	14.05	166.16
	6	0	0	0.01	0	0.01	-0.01
	7	12.20	326.37	30.57	314.17	18.37	295.79
	8	3.80	54.62	11.83	50.82	8.03	42.80
	9	0	0	0.01	0	0.01	-0.01
	10	-17.60	180.03	10.80	197.63	28.40	169.23

Table 6.8: Comparison of the Objective Values for Random Tests (2)

Chapter 7

Enhancements

In this chapter, we explore ways to improve the solution time for the mixed-integer programming problems. First, we develop heuristics to efficiently find "good" feasible solutions. Second, we study two sets of valid inequalities in hopes to find effective cutting planes.

7.1 Heuristics

As we will see in Chapter 9, CPLEX takes significant time just to find a feasible solution for larger problems. Fortunately, we can easily find a feasible mixed-integer solution for the formulations of all our models. For example, in the alternative formulations of the models, the solution

$$x_{1j} = 1, \qquad \forall j,$$

$$x_{ij} = 0, \qquad \forall i \neq 1,$$

$$\pi_j = R_{1j}, \qquad \forall j$$

is a feasible solution (all other variables can be easily determined after the x variables are fixed). Thus, we may provide the solver a "good" starting feasible solution in hopes of decreasing the solution time.

7.1.1 Heuristic 0

One possible strategy, which we call *Heuristic 0*, is to set $\beta_{ij*} = 1$ for each segment i where $R_{ij*} = \max_j R_{ij}$. The other β variables are set accordingly to ensure feasibility. The pseudo-code is presented in Algorithm 1. For special data sets, this heuristic can result in the optimal solution.

Lemma 7.1.1. *Suppose the conditions are the same as stated in Lemma 6.1.1. That is, for every segment i, we can find a unique product $p(i)$ such that $R_{ip(i)} = max_j R_{ij}$, and for each of such product $p(i)$, segment i is the unique segment such that $R_{ip(i)} = max_k R_{kp(i)}$.*

Then Heuristic 0 gives the optimal solution.

Proof. In the first step when we set $\beta_{ij} = 1$ if $R_{ij} = \max_k R_{ik}$, we have

$$\beta_{ij} := \begin{cases} 1, & \text{if } j = p(i), \\ 0, & \text{otherwise.} \end{cases}$$

which is the optimal solution by Lemma 6.1.1. This is the solution found by Heuristic 0, thus, it provides the optimal solution for this particular data set. □

However, Heuristic 0 may not yield a strong solution in general. For the rest of this section, we discuss a few simple techniques for improving on the feasible solution found by Heuristic 0.

Algorithm 1 Heuristic 0

---

heuristic0(numSegments, numProducts, N, R, beta, pi)

1: for each segment i, set $\beta_{ij} = 1$ where $j = argmax_j R_{ij}$
2: for each product j, set $\pi_j = min_i\{R_{ij} : \beta_{ij} = 1\}$
3: return β

7.1.2 Heuristic 1

After running Heuristic 0, we select a product k that is bought by at least one customer segment, and let l be the segment with the lowest reservation price that buys product k. We

Algorithm 2 Make one swap if possible

swap(j, k, β, π)

for seg $i = argmin_i\{R_{ij} : \beta_{ij} = 1\}$, find the greatest increase in objective value if segment i buys another product k (or does not buy any product) instead of j.
if increase in obj val ≤ 0 **then**
    return 0
end if
$\beta_{ij} := 0$
if product k is found **then**
    $\beta_{ik} := 1$
end if
make the solution feasible and set the prices π to the appropriate values
return the increase in objective value after ONE swap

Algorithm 3 Heuristic 1

heuristic1(numSegments, numProducts, N, R, β, π)

β := heuristic0(numSegments, numProducts, N, R, β, π)
k := -1;
repeat
    increase := 0;
    make a heap H where the elements are products and the comparator compares the product prices
    while *increase* ≤ 0 and H is not empty **do**
        j := H.pop()
        increase := swap(j, k, β, π)
    end while
until *increase* ≤ 0

consider the change in the objective value if the segment does not buy product k anymore and perhaps buys another product q that it does not currently buy (i.e., β_{lq} currently equals to 0). This can be thought of as swapping β_{lk} with β_{lq}. We select the option that increases the objective value the most and modify the β variables accordingly. That is, segment l either does not buy product k anymore, or it buys another product instead of product k. If none of the options increases the objective value, we make no changes. We repeat until no swaps can be made to increase the objective value. This algorithm terminates because the objective function is bounded and the objective value strictly increases after each swap.

The order in which we select the products to be examined affects the final solution that will be given by the heuristic. The goal is to use an order that maximizes the total increase in the objective value. In this heuristic, we sort the products by the price and examine the products in the order of the lowest price to the highest price. If we make a change in any iteration, we sort the products again since the prices may change, and start with the lowest-priced product again. The heuristic stops when no changes can be made after examining all the products consecutively from the lowest price to the highest price. The pseudo-code is shown in Algorithm 3 and the *swap* subroutine is shown in Algorithm 2.

This simple heuristic can be used to find a feasible integral solution for any of the models. The only part that needs to be changed is how the objective value is calculated. The version shown here makes use of β, but it can be easily modified to use the x variables as in the alternative formulation.

7.1.3 Heuristic 2

Heuristic 1 (Section 7.1.2) can be modified to have a polynomial runtime if the price of the product that we examine is non-decreasing in each iteration.

From experiments of Heuristic 1, we noticed that if a swap can be made when product k at price π_k is selected, it is very unlikely that a swap can be made for a product at a price lower than π_k in subsequent iterations. Therefore, we would expect the results to be similar if we do not examine products with lower prices again.

Heuristic 2 is the same as Heuristic 1 (Section 7.1.2) but the products are selected in a different order. After a customer is swapped out of product k with price π_k before the swap, only products with prices at least π_k are examined. The price of product k increases after a swap, so it will be examined again if there are still customers buying product k. If a new product s is bought and if its new price π_s^{new} is less than π_k, then product s will never be examined. If a product cannot be swapped to increase the objective value, then

Algorithm 4 Heuristic 2

heuristic2(numSegments, numProducts, N, R, β, π)

β := heuristic0(numSegments, numProducts, N, R, β, π)
increase := 0;
make a heap H where the elements are products and the comparator compares the product prices
while H is not empty **do**
    k := -1;
    $\pi_{temp} := \pi_j$
    j := H.pop()
    increase := swap(j, k, β, π)
    if $increase > 0$ **then**
        if product j is still bought by some segment **then**
            H.push(j)
        end if
        if $k \geq 0$ and $\pi_k \geq \pi_{temp}$ **then**
            H.push(k)
        end if
    end if
end while

it will not be examined again. The pseudo-code is presented in Algorithm 4.

Let $O(f(n,m))$ be the runtime to calculate the increase in objective value if segment l does not buy product k anymore or if segment l buys product s instead of product k, where n is the number of customer segments and m is the number of products. Clearly, $f(n,m)$ is polynomial in n and m, since the runtime to calculate the objective value is polynomial.

Lemma 7.1.2. *The runtime of Heuristic 2 is polynomial.*

Proof. The time it takes to examine a product k is $O(mf(n,m))$ since we consider up to m products that product k can swap with. A product is examined multiple times only if its price increases after a swap. Since a product's price always equals to a segment's reservation price, it can only increase at most n times. So there are at most $O(nm)$ iterations to examine a product, and each iteration has a runtime of $O(mf(n,m))$.

Therefore, the runtime of Heuristic 2 is $O(nm^2 f(n,m))$. $\square$

7.1.4 Heuristic 3

Heuristic 3 is a hybrid between Heuristic 1 (Section 7.1.2) and Heuristic 2 (Section 7.1.3). It examines the products in the same way, but after a swap in which segment l buys product s instead of product k and $\pi_s^{new} < \pi_k$ (equivalently, $R_{ls} < R_{lk}$), it would examine all the products with prices $\geq \pi_s^{new}$. That is, the price of the products that it examines decreases only if a product has a lower price after a swap. The pseudo-code is presented in Algorithm 5.

It is not yet clear if this heuristic has an exponential worst case runtime. However, experimental results shows that it has a similar runtime as Heuristic 2 and the resulting objective value is usually better (Tables 7.1, 7.2, and 7.3).

7.1.5 Comparison of the Heuristics

Tables 7.1, 7.2, 7.3, and 7.4 show the results of the three heuristics with problem instances of different sizes as inputs. The column "n" is the number of segments and "m" is the number of products in the problem instance.

Tables 7.1 and 7.2 show the initial objective value found before any swaps (i.e., Heuristic 0), and the number of swaps performed, the number of CPU seconds required and the final objective value found by each heuristic. The objective values are rounded to the nearest integer. Tables 7.3 and 7.4 show the difference in time required and the final objective value for each pair of the heuristics. For example, the "Heur. 1 – Heur. 2" columns show the time and objective value of Heuristc 2 subtracted from the time and objective value of Heurstic 1, respectively.

All of the heuristics terminates in a very short time. The time required for Heuristic 1 to terminate increases significantly as the problem size increases. The objective values found are better than or at least as good as the ones found by the other two heuristics, except in one problem instance (when n = 60, m = 20) where Heuristic 2 has a better

Algorithm 5 Heuristic 3

heuristic3(numSegments, numProducts, N, R, β, π)

β := heuristic0(numSegments, numProducts, N, R, β, π)
increase := 0;
make a heap H where the elements are products and the comparator compares the product prices
while H is not empty **do**
  k := -1;
  $\pi_{temp} := \pi_j$
  j := H.pop()
  increase := swap(j, k, β, π)
  if $increase > 0$ **then**
    if product j is still bought by some segment **then**
      H.push(j)
    end if
    if $k \geq 0$ **then**
      if $\pi_k \geq \pi_{temp}$ **then**
        H.push(k)
      else
        for products l where $\pi_k \leq \pi_l \leq \pi_{temp}$ **do**
          H.push(l)
        end for
      end if
    end if
  end if
end while

solution. Experimental results shows that Heuristic 3 has a similar runtime as Heuristic 2 and the resulting objective value is usually better. We can see from Tables 7.3 and 7.4 that Heuristic 3 found a lower objective value than Heuristic 2 in one problem instance only (when n = 60, m = 20).

The effect of using a starting solution found by the heuristics for the Uniform Distribution Model is explored in Chapter 9.

7.2 Valid Inequalities

To further improve the solution time for the mixed-integer programming models, we considered several mixed-integer cuts for the various choice models.

7.2.1 Convex Quadratic Cut

In the original Uniform Distribution Model (Section 2.1), the variable a_{ij} were introduced to convexify the bilinear inequalities:

$$\sum_{j=1}^{m} t_i \beta_{ij} \leq \sum_{j=1}^{m} p_{ij}, \quad \forall i.$$

We wish to include a convex constraint in the mixed-integer programming formulation that is implied by the above inequalities and some valid convex inequalities.

Let M_i be a positive number (as small as possible) such that $t_i^2 \leq M_i$, for every feasible solution $(t_1, \ldots, t_n, \beta_{11}, \ldots, \beta_{nm}, p_{11}, \ldots, p_{nm})$ of the mixed integer programming problem. Also, note that $\beta_{ij}^2 \leq \beta_{ij}$. Combining these relations together yields the following set of valid inequalities:

$$a_i t_i^2 + b_i \sum_{j=1}^{m} (\beta_{ij}^2 - \beta_{ij}) + \sum_{j=1}^{m} t_i \beta_{ij} - \sum_{j=1}^{m} p_{ij} \leq a_i M_i, \quad i = 1, \ldots, n, \tag{7.1}$$

where a_i and b_i are nonnegative constants. With appropriate values of a_i and b_i, the above set of quadratic inequalities would represent a convex region.

Lemma 7.2.1. *The function* $f(t, \beta_1, \ldots, \beta_m, p_1, \ldots, p_m) = at^2 + b \sum_{j=1}^{m} (\beta_j^2 - \beta_j) + \sum_{j=1}^{m} t\beta_j - \sum_{j=1}^{m} p_j$ *is a convex function iff* $a > 0$, $b > 0$ *and* $ab \geq \frac{m}{4}$.

Proof. The Hessian of f is

$$\nabla^2 f = \begin{bmatrix} \boldsymbol{A} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}$$

			Heuristic 1			Heuristic 2			Heuristic 3		
n	m	Initial Obj Val	# Swaps	Time	Final Obj Val	# Swaps	Time	Final Obj Val	# Swaps	Time	Final Obj Val
2	2	2656.00	0	0.003	2656	0	0.003	2656	0	0.003	2656
2	5	121520	0	0.003	121520	0	0.002	121520	0	0.003	121520
2	10	165960	0	0.006	165960	0	0.006	165960	0	0.006	165960
2	20	207680	0	0.010	207680	0	0.010	207680	0	0.010	207680
2	60	66801	0	0.025	66801	0	0.024	66801	0	0.024	66801
2	100	66801	0	0.039	66801	0	0.040	66801	0	0.039	66801
5	2	176584	1	0.004	212238	1	0.004	212238	1	0.004	212238
5	5	131346	5	0.020	164038.67	5	0.014	164038.67	5	0.015	164039
5	10	177403	10	0.079	217832	4	0.020	212348	10	0.048	217832
5	20	124311	0	0.022	124311	0	0.022	124311	0	0.022	124311
5	60	377480	0	0.049	377480	0	0.049	377480	0	0.049	377480
5	100	316906	8	0.473	318770	8	0.276	318770	8	0.277	318770
10	2	543760	0	0.003	543760	0	0.003	543760	0	0.004	543760
10	5	307678	3	0.018	323976	3	0.012	323976	3	0.016	323976
10	10	320574	11	0.099	379851	9	0.033	375924	11	0.044	379851
10	20	448921	21	0.365	555829	21	0.114	555829	21	0.134	555829
10	60	489794	11	0.324	624070	10	0.235	624070	10	0.236	624070
10	100	528288	7	0.807	605906	7	0.305	605906	7	0.371	605906

Table 7.1: Comparison of Heuristics (1)

			Heuristic 1			Heuristic 2			Heuristic 3		
n	m	Initial Obj Val	# Swaps	Time	Final Obj Val	# Swaps	Time	Final Obj Val	# Swaps	Time	Final Obj Val
20	2	503054	8	0.019	544933	8	0.013	544933	8	0.016	544933
20	5	552958	9	0.042	597752	9	0.022	597752	9	0.023	597752
20	10	624238	16	0.128	746544	15	0.048	743491	15	0.055	743491
20	20	698462.78	35	0.951	823206	31	0.148	822773	35	0.209	823206
20	60	934624	48	5.134	1174843	37	0.647	1094627	48	0.958	1174844
20	100	806249	25	4.256	868827	10	0.445	851503	17	0.699	863241
60	2	1136086	6	0.019	1228719	6	0.013	1228719	6	0.014	1228719
60	5	1407568	4	0.028	1526584	4	0.020	1526584	4	0.025	1526584
60	10	1480566	39	0.378	1770829	39	0.115	1770829	39	0.123	1770829
60	20	1697868	49	1.241	1957967	56	0.279	1978848	49	0.286	1957967
60	60	1976530	92	21.621	2476842	88	1.376	2437774	91	1.558	2476277
60	100	2824349	83	52.689	3218884	55	2.385	3171053	76	5.013	3215687
100	2	2201501	9	0.028	2368924	9	0.020	2368924	9	0.020	2368924
100	5	2177005	19	0.103	2493507	19	0.049	2493507	19	0.053	2493507
100	10	2307042	43	0.415	2703764	43	0.146	2703764	43	0.153	2703764
100	20	2384416	45	1.854	2736795	45	0.280	2736795	45	0.285	2736795
100	60	3548047	89	21.130	4128381	88	1.725	4127951	88	1.789	4127951
100	100	3568616	132	108.240	4380587	120	3.938	4338345	126	6.679	4366574

Table 7.2: Comparison of Heuristics (2)

		Heur. 1 – Heur. 2		Heur. 1 – Heur. 3		Heur. 2 – Heur. 3	
n	m	Time	Obj Val	Time	Obj Val	Time	Obj Val
2	2	0	0	0	0	0	0
2	5	0.001	0	0	0	-0.001	0
2	10	0	0	0	0	0	0
2	20	0	0	0	0	0	0
2	60	0.001	0	0.001	0	0	0
2	100	-0.001	0	0	0	0.001	0
5	2	0	0	0	0	0	0
5	5	0.007	0	0.006	0	-0.001	0
5	10	0.060	5484	0.031	0	-0.028	-5484
5	20	0	0	0	0	0	0
5	60	0	0	0	0	0	0
5	100	0.197	0	0.196	0	-0.001	0
10	2	0	0	-0.001	0	-0.001	0
10	5	0.006	0	0.002	0	-0.004	0
10	10	0.065	3927	0.055	0	-0.011	-3927
10	20	0.251	0	0.231	0	-0.020	0
10	60	0.089	0	0.088	0	-0.001	0
10	100	0.503	0	0.436	0	-0.066	0
20	2	0.006	0	0.003	0	-0.003	0
20	5	0.020	0	0.019	0	-0.001	0
20	10	0.080	3053	0.073	3053	-0.007	0
20	20	0.802	433	0.742	0	-0.061	-433
20	60	4.487	80216	4.175	0	-0.311	-80216
20	100	3.811	17324	3.558	5586	-0.254	-11738

Table 7.3: Comparison of Heuristics (3)

		Heur. 1 – Heur. 2		Heur. 1 – Heur. 3		Heur. 2 – Heur. 3	
n	m	Time	Obj Val	Time	Obj Val	Time	Obj Val
60	2	0.006	0	0.005	0	-0.001	0
60	5	0.008	0	0.003	0	-0.005	0
60	10	0.263	0	0.255	0	-0.008	0
60	20	0.962	-20881	0.956	0	-0.007	20881
60	60	20.245	39068	20.064	565	-0.182	-38503
60	100	50.304	47831	47.677	3197	-2.627	-44634
100	2	0.009	0	0.008	0	-0.001	0
100	5	0.055	0	0.051	0	-0.004	0
100	10	0.268	0	0.262	0	-0.007	0
100	20	1.574	0	1.569	0	-0.005	0
100	60	19.406	430	19.341	430	-0.064	0
100	100	104.302	42242	101.562	14013	-2.741	-28229

Table 7.4: Comparison of Heuristics (4)

where $\boldsymbol{A} = \begin{bmatrix} 2a & \boldsymbol{e}^T \\ \boldsymbol{e} & 2\boldsymbol{B} \end{bmatrix}$, $\boldsymbol{B}$ is an $m \times m$ diagonal matrix with $b, b, \ldots, b$ on the diagonal, and $\boldsymbol{e}$ is a vector of ones. Since f is twice continuously differentiable, f is convex iff $\boldsymbol{A}$ is a positive semi-definite matrix.

If $b \leq 0$, then $\boldsymbol{A}$ is not positive semi-definite and f is not convex. So we can assume $b > 0$.

The Schur-complement of $\boldsymbol{B}$ in $\boldsymbol{A}$ is $2a - \frac{1}{2b}(\boldsymbol{e}^T\boldsymbol{e})$, thus

$$\boldsymbol{A} \succeq 0 \quad \Leftrightarrow \quad a - \frac{m}{4b} \geq 0 \quad \Leftrightarrow \quad ab \geq \frac{m}{4}.$$

$\square$

Next, we generalize the above construction to allow different coefficients b_j for the inequalities $\beta_{ij}^2 \leq \beta_{ij}$. Let $\boldsymbol{b}$ denote the vector $(b_1, b_2, \ldots, b_m)^T$ and let $\boldsymbol{B}$ denote the $m \times m$ diagonal matrix with entries $b_1, b_2, \ldots, b_m$ on the diagonal.

Lemma 7.2.2. *The function $F(t, \beta_1, \ldots, \beta_m, p_1, \ldots, p_m) = at^2 + \sum_{j=1}^m b_j(\beta_j^2 - \beta_j) + \sum_{j=1}^m t\beta_j - \sum_{j=1}^m p_j$ is a convex function iff $\boldsymbol{b} > 0$ and $a \geq \sum_{j=1}^m \frac{1}{4b_j}$.*

Proof. As in the proof of the previous lemma, F is twice continuously differentiable. The Hessian of F is

$$\nabla^2 F = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0}^T & \boldsymbol{0} \end{bmatrix},$$

where $\boldsymbol{A} := \begin{bmatrix} 2a & \boldsymbol{e}^T \\ \boldsymbol{e} & 2\boldsymbol{B} \end{bmatrix}$. Therefore, F is convex iff $\boldsymbol{A}$ is positive semidefinite. If for some j, $b_j \leq 0$, then $\boldsymbol{A}$ is not positive semidefinite. Therefore, $\boldsymbol{b} > \boldsymbol{0}$. Let $\bar{\boldsymbol{b}} := (\frac{1}{\sqrt{b_1}}, \frac{1}{\sqrt{b_2}}, \ldots, \frac{1}{\sqrt{b_m}})^T$. Also, if $a \leq 0$ then $\boldsymbol{A}$ is not positive semidefinite, thus $a > 0$. The Schur complement of a in $\boldsymbol{A}$ is $2\boldsymbol{B} - \frac{1}{2a}\boldsymbol{e}\boldsymbol{e}^T$. Thus,

$$\begin{aligned} \boldsymbol{A} \succeq 0 \quad &\Leftrightarrow \quad 2\boldsymbol{B} - \frac{1}{2a}\boldsymbol{e}\boldsymbol{e}^T \succeq 0 \\ &\Leftrightarrow \quad 4I - \frac{1}{a}\bar{\boldsymbol{b}}\bar{\boldsymbol{b}}^T \succeq 0 \\ &\Leftrightarrow \quad 4\bar{\boldsymbol{b}}^T\bar{\boldsymbol{b}} - \frac{1}{a}(\bar{\boldsymbol{b}}^T\bar{\boldsymbol{b}})^2 \geq 0 \end{aligned}$$

$$\Leftrightarrow \quad a \geq \frac{\bar{\boldsymbol{b}}^T \bar{\boldsymbol{b}}}{4} = \sum_{j=1}^{m} \frac{1}{4b_j}.$$

□

Corollary 7.2.1. *Let M_i be as above, and $b_1 > 0, b_2 > 0, \ldots, b_m > 0$, and $a \geq \sum_{j=1}^{m} \frac{1}{4b_j}$ be given. Then the inequality*

$$a_i t_i^2 + \sum_{j=1}^{m} [b_j \beta_{ij}^2 + (t_i - b_j)\beta_{ij} - p_{ij}] \leq aM_i$$

is a valid convex quadratic inequality for the feasible region of the mixed integer programming problem.

Several computational experiments on randomly generated problem instances were performed to test whether these valid inequalities are actually a cut. The inequalities (7.1) were added to the Uniform Distribution Model formulation (2.1) and the resulting mixed-integer programming problem with convex quadratic inequalities (MIQP) were solved by CPLEX 9.1. However, it was difficult to test whether (7.1) cut off any regions of the LP feasible region of (2.1) because of the numerical inaccuracies in the solutions provided by CPLEX. For example, in a random instance with $n = 10$ and $m = 10$, CPLEX returned an optimal objective value of 9218.698757 for the continuous relaxation of the MIQP. For the LP relaxation of (2.1), CPLEX returned an optimal value of 9219.085299. However, we verified that this latter optimal solution did not violate any of the inequalities (7.1). Thus, the difference in the optimal value was due to numerical inaccuracies of the CPLEX quadratic programming solver. Clearly, more testing needs to be done, perhaps with a different optimization software.

7.2.2 Knapsack Covers

The pure 0-1 formulation (2.3) shown in Section 2.4 may not be as strong as the mixed-integer formulation (2.2). However, we may be able to exploit the vast amount of work done in developing strong valid inequalities for pure 0-1 programming problems for formulation (2.3). These inequalities can not only improve the solution time for the 0-1 problem, but we may be able to project them to the space of Formulation (2.4) to strengthen the mixed-integer programming formulation as well.

One obvious family of valid inequalities are the knapsack covers [8]. From (2.3), we have the constraints

$$\sum_{j=1}^{m} \beta_{ij} = \sum_{k=0}^{m} k y_{ik}, \quad i, \ldots, m,$$

(where we substituted $\beta_{ij} := \sum_{l:R_{l,j} \leq R_{ij}} x_{lj}$ purely for notational ease) and

$$\sum_{k=0}^{m} y_{ik} = 1, \quad i = 1, \ldots, n.$$

From these, for a given i and k, we get:

$$\sum_{j=1}^{m} \beta_{ij} \leq \sum_{l=0}^{k} k y_{il} + \sum_{l=k+1}^{m} m y_{il}$$

$$\Rightarrow \sum_{j=1}^{m} \beta_{ij} \leq \sum_{l=0}^{k} k y_{il} + \sum_{l=k+1}^{m} m y_{il} + m - m \sum_{k=0}^{m} y_{ik}$$

$$\Rightarrow \sum_{j=1}^{m} \beta_{ij} \leq -\sum_{l=0}^{k} (m-k) y_{il} + m$$

$$\Rightarrow \sum_{j=1}^{m} \beta_{ij} + (m-k) \sum_{l=0}^{k} y_{il} \leq m,$$

where the last inequality is a knapsack constraint (note that $\sum_{l=0}^{k} y_{il} \in \{0,1\}$ in the integer solution so we can treat the term as a 0-1 variable). For a given i and k, let P_{ik} be a subset of $k+1$ products , i.e, $P_{ik} \subseteq \{1, \ldots, m\}$, $|P_{ik}| = k+1$. Thus, the corresponding knapsack cover inequality is

$$\sum_{j \in P_{ik}} \beta_{ij} + \sum_{l=0}^{k} y_{il} \leq k+1. \tag{7.2}$$

Given a fractional solution to (2.3), separating (7.2) can be done in polynomial time. Given x_{ij}'s , and thus β_{ij}'s, we rank β_{ij} for each i, $i = 1, \ldots, n$. For each k, let $P^*_{ik} = \{j :$

β_{ij} is one of the k^{th} largest β_{ij}'s, $j = 1, \ldots, m\}$. Thus, for each i and k, the corresponding cover inequality is violated by the current solution if and only if $\sum_{j\in P^*_{ik}} \beta_{ij} + \sum_{l=0}^{k} y_{il} > k+1$.

We can also incorporate all of the inequalities (7.2) to (2.3) with only polynomial numbers of additional constraints and variables.

Lemma 7.2.3. *Given i and k, there exists β_{ij}, $j = 1, \ldots, m$ and y_{il}, $l = 0, \ldots, k$ satisfying (7.2) for all $P_{ik} \subseteq \{1, \ldots, m\}$, $|P_{ik}| = k+1$ if and only if there exists q and p_j, $j = 1, \ldots, m$ such that*

$$\begin{aligned} (k+1)q + \sum_{j=1}^{m} p_j + \sum_{l=0}^{k} y_{il} &\leq k+1, \\ q + p_j &\geq \beta_{ij}, \quad j = 1, \ldots, m, \\ p_j &\geq 0, \quad j = 1, \ldots, m. \end{aligned}$$

Proof. For given β_{ij}'s, finding the most violated subset P^*_{ik} for (7.2) is equivalent to solving

$$\begin{aligned} \max \quad & \textstyle\sum_{j=1}^{m} \beta_{ij} z_j, \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{m} z_j = k+1, \\ & 0 \leq z_j \leq 1, \qquad j = 1, \ldots, m \end{aligned}$$

Since the feasible region of the above LP is an integral polyhedron, and since the LP is clearly feasible and bounded, it has an optimal 0-1 solution corresponding to the characteristic vector of P^*_{ik}. The Dual of this LP is:

$$\begin{aligned} \min \quad & (k+1)q + \textstyle\sum_{j=1}^{m} p_j, \\ \text{s.t.} \quad & q + p_j \geq \beta_{ij}, \qquad j = 1, \ldots, m, \\ & p_j, \qquad j = 1, \ldots, m. \end{aligned}$$

If there exists β_{ij}'s and y_{il} that satisfies (7.2) for all covers P_{ik}, then it must satisfy (7.2) for P^*_{ik}. Thus, from strong duality, there exists q and p_j satisfying the constraints for the Dual LP and $\sum_{j\in P^*_{ik}} \beta_{ij} = (k+1)q + \sum_{j=1}^{m} p_j$.

Conversely, if there exists q and p_j that satisfies the constraints of the Dual LP and there is a y_{il} such that $(k+1)q + \sum_{j=1}^{m} p_j + \sum_{l=0}^{k} y_{il} \leq k+1$, then from weak duality, $\sum_{j\in P_{ik}} \beta_{ij} \leq (k+1)q + \sum_{j=1}^{m} p_j$ for all P_{ik}'s and thus, $\sum_{j\in P_{ik}} \beta_{ij} + \sum_{l=0}^{k} y_{il} \leq k+1$ for all P_{ik}'s. □

Thus, we can either iteratively separate the knapsack cover inequalities, or from Lemma 7.2.3, add the constraints:

$$\begin{aligned}
(k+1)q_{ik} + \sum_{j=1}^{m} p_{i,j,k} + \sum_{l=0}^{k} y_{il} \leq k+1, &\quad i = 1,\dots,n; k = 0,\dots,m, \\
q_{ik} + p_{ijk} \geq \beta_{ij}, &\quad j = 1,\dots,m; i = 1,\dots,n; k = 0,\dots,m, \\
p_{ijk} \geq 0, &\quad j = 1,\dots,m; i = 1,\dots,n; k = 0,\dots,m,
\end{aligned} \tag{7.3}$$

to (2.3).

Table 7.5 illustrates that these knapsack covers (7.2) are indeed cuts. It compares formulation (2.3) with and without the cover inequalities (7.3) in terms of the objective value of their linear programming relaxation on the same randomly generated instances shown in Section 2.4. Again, n is the number of customer segments, m is the number of products, and v is a label of the problem instance. LP objective values in bold corresponds to the IP optimal value.

These knapsack cover inequalities (7.2) can also be used to generate cuts for the mixed-integer programming formulation (2.2).

Lemma 7.2.4. *Suppose $\bar{x}_{ij}$ is a fractional solution of (2.2) and let $\bar{\beta}_{ij} = \sum_{l:R_{lj} \leq R_{ij}} \bar{x}_{lj}$. For a given i, $i = 1,\dots,n$, if there are no y_{ik}'s that satisfies*

$$\begin{aligned}
\textstyle\sum_{k=0}^{m} y_{ik} = 1, & \\
\textstyle\sum_{k=0}^{m} k y_{ik} = \sum_{j=1}^{m} \bar{\beta}_{ij}, & \\
\textstyle\sum_{l=0}^{k} y_{il} \leq k+1 - \sum_{j \in P^*_{ik}} \bar{\beta}_{ij}, &\quad k = 0,\dots,m
\end{aligned} \tag{7.4}$$

where $P^*_{ik} = \{j | \bar{\beta}_{ij}$ is one of the k largest $\bar{\beta}_{ij}, j = 1,\dots,m\}$, then

$$\sum_{j=1}^{m} v\beta_{ij} + \sum_{j \in P^*_{ik}} w_k \beta_{ij} \leq \sum_{k=0}^{m} (k+1) w_k \tag{7.5}$$

is a valid inequality for (2.2) that cuts of $\bar{x}_{ij}$, where

$$\begin{aligned}
u + kv + \textstyle\sum_{l=0}^{k} w_k \geq 0, &\qquad k = 0,\dots,m, \\
u + \textstyle\sum_{j=1}^{m} \bar{\beta}_{ij} v + \left(k+1 - \sum_{j \in P^*_{ik}} \bar{\beta}_{ij}\right) w_k < 0, & \\
w_k \geq 0, &\qquad k = 0,\dots,m,
\end{aligned}$$

n	m	v	(2.3) without (7.3)	(2.3) with (7.3)
4	4	1	2564.71	2399.63
		2	**3404.00**	**3404.00**
		3	**333.00**	**333.00**
		4	3060.92	**3005.67**
		5	3360.95	3271.48
4	10	1	406.42	390.50
		2	398.19	391.36
		3	397.36	373.89
		4	389.98	365.59
		5	402.74	384.18
10	4	1	802.93	799.31
		2	856.12	853.60
		3	850.95	848.58
		4	856.85	842.16
		5	925.44	911.67
10	10	1	997.40	990.70
		2	1008.53	1003.15
		3	1021.94	1016.75
		4	872.92	864.30
		5	1021.50	1013.01

Table 7.5: Strength of Knapsack Cover inequalities (7.2)

for some u.

Proof. The system (7.4) are valid inequalities for the pure 0-1 formulation (2.3). Thus, $\hat{x}_{ij}$ is a feasible integer solution to (2.2) if and only if $\hat{x}_{ij}$ and $\hat{y}_{ik} = 1$ where $k = \sum_{l:R_{lj} \leq R_{ij}} x_{lj}$ is a feasible integer solution to (2.3).

From Farkas' Lemma, (7.4) is infeasible if and only if there exists u, v, and w_k, $k =$

$0, \ldots, m$, where

$$\begin{aligned} u + kv + \textstyle\sum_{l=0}^{k} w_k &\geq 0, & k = 0, \ldots, m, \\ u + \textstyle\sum_{j=1}^{m} \bar{\beta}_{ij} v + \left(k + 1 - \textstyle\sum_{j \in P^*_{ik}} \bar{\beta}_{ij}\right) w_k &< 0, & \\ w_k &\geq 0, & k = 0, \ldots, m. \end{aligned}$$

Therefore, $\sum_{j=1}^{m} v\beta_{ij} + \sum_{j \in P^*_{ik}} w_k \beta_{ij} \leq \sum_{k=0}^{m} (k+1) w_k$ is a valid inequality for (2.2) and are violated by $\bar{\beta}_{ij}$.

$\square$

Chapter 8

Product Capacity and Cost

In all of our discussions thus far, we have assumed that there are no capacity limits nor costs for our products. Clearly, this is not a realistic assumption in many applications. In this chapter, we discuss how we can incorporate capacity limits and product costs into some of our customer choice models.

8.1 Product Capacity

Product capacity limits are crucial constraints for products such as airline seats and hotel rooms. Certain consumer choice models handle capacity constraints easily, whereas it poses a challenge to others. We present this extension for the Uniform Distribution Model, the Weighted Uniform Model, and the Share-of-Surplus Model with restricted prices. We were not able to incorporate the capacity constraint in the Price Sensitive Model while maintaining the convexity of the continuous relaxation. In all of the following subsections, we assume that the company can sell up to Cap_j units of product j, $Cap_j \geq 0$, $j = 1, \ldots, m$.

8.1.1 Uniform Distribution and Weighted Uniform Model

Capacity constraints can be incorporated to the mixed-integer formulations of the Uniform Distribution Model (Chapter 2) and the Weighted Uniform Model (Chapter 3) with some additional variables. We discuss the formulation for the Uniform Distribution Model only, since it extends easily to the Weighted Uniform Model.

In the Uniform Distribution Model, the expected number of customers that buy product j is $\sum_i N_i \frac{\beta_{ij}}{\sum_k \beta_{ik}}$ if $\sum_k \beta_{ik} \geq 1$ and is 0 if $\sum_k \beta_{ik} = 0$.

Let B_{ij} be an auxiliary variable such that $B_{ij} := \frac{\beta_{ij}}{\sum_k \beta_{ik}}$ if $\sum_k \beta_{ik} \geq 1$ and is 0 if $\sum_k \beta_{ik} = 0$ (i.e., the fraction of customers from segment i buying product j, Pr_{ij}). Thus,

$$\beta_{ij} = B_{ij} \sum_k \beta_{ik}$$

Let $b_{ijk} := B_{ij}\beta_{ik}$. The capacity constraint can be represented by the following set of linear constraints:

$$\begin{aligned}
\textstyle\sum_i N_i B_{ij} \leq Cap_j, &\quad \forall j, \\
\beta_{ij} = \textstyle\sum_k b_{ijk}, &\quad \forall i, \forall j, \\
b_{ijk} \leq \beta_{ik}, &\quad \forall i, \forall j, \forall k, \\
b_{ijk} \geq B_{ij} - (1 - \beta_{ik}), &\quad \forall i, \forall j, \forall k, \\
b_{ijk} \leq B_{ij}, &\quad \forall i, \forall j, \forall k, \\
b_{ijk} \geq 0, &\quad \forall i, \forall j, \forall k.
\end{aligned} \tag{8.1}$$

The above constraints can also be represented by x_{ij} variables of Section 2.2 instead of the β_{ij} variables.

8.1.2 Share-of-Surplus Model

Similar to section (8.1.1), in the Share-of-Surplus Model with restricted prices (4.8), the expected number of customers that buy product j is $\sum_i N_i \left(\frac{\sum_{l:R_{lj} \leq R_{ij}} (R_{ij} - R_{lj}) x_{lj}}{\sum_k (\sum_{l:R_{lk} \leq R_{ik}} (R_{ik} - R_{lk}) x_{lk})} \right)$ if $\sum_k (\sum_{l:R_{lk} \leq R_{ik}} (R_{ik} - R_{lk}) x_{lk}) \neq 0$.

Let B_{ij} be an auxiliary variable such that

$$B_{ij} := \begin{cases} \left(\frac{\sum_{l:R_{lj} \leq R_{ij}} (R_{ij} - R_{lj}) x_{lj}}{\sum_k (\sum_{l:R_{lk} \leq R_{ik}} (R_{ik} - R_{lk}) x_{lk})} \right), & \text{if } \sum_k (\sum_{l:R_{lk} \leq R_{ik}} (R_{ik} - R_{lk}) x_{lk}) \neq 0 \\ 0, & \text{otherwise.} \end{cases}$$

(again, B_{ij} is the fraction of customers from segment i buying product j, or Pr_{ij}). Thus,

$$\sum_{l:R_{lj}\leq R_{ij}} (R_{ij} - R_{lj})x_{lj} = B_{ij} \sum_k (\sum_{l:R_{lk}\leq R_{ik}} (R_{ik} - R_{lk})x_{lk})$$

Let $b_{ijlk} := B_{ij}x_{lk}$. Just as before, the capacity constraint can be represented by the following set of linear constraints:

$$\begin{array}{rl} \sum_i N_i B_{ij} \leq Cap_j, & \forall j, \\ \sum_{l:R_{lj}\leq R_{ij}} (R_{ij} - R_{lj})x_{lj} = \sum_k (\sum_{l:R_{lk}\leq R_{ik}} (R_{ik} - R_{lk})b_{ijlk}) & \forall i, \forall j, \\ b_{ijlk} \leq x_{lk}, & \forall i, \forall j, \forall l, \forall k, \\ b_{ijlk} \geq B_{ij} - (1 - x_{lk}), & \forall i, \forall j, \forall l, \forall k, \\ b_{ijlk} \leq B_{ij}, & \forall i, \forall j, \forall l, \forall k, \\ b_{ijlk} \geq 0, & \forall i, \forall j, \forall l, \forall k. \end{array} \tag{8.2}$$

8.1.3 Risk Products

In some cases, companies may want to penalize against under-shooting a capacity. For example, if there is a large fixed cost or initial investment for product j, the company may sacrifice revenue and decrease its price to ensure that all of the product is sold. We call such products *risk* products. For these products, we may add a penalty for under-shooting in the objective, i.e., given a user-defined penalty coefficient $w_j > 0$ for under-selling product j, we modify the objective to

$$\sum_{i=1}^{n} N_i \sum_{j=1}^{m} \pi_j Pr_{ij} - \sum_{j=1}^{m} w_j (Cap_j - \sum_{i=1}^{n} N_i B_{ij})$$

or

$$\sum_{i=1}^{n} \sum_{j=1}^{m} N_i (\pi_j Pr_{ij} + w_j B_{ij})$$

where B_{ij} is as before.

From a profit optimization point of view, it is sub-optimal to forcibly sell unprofitable products. Such a policy implies that the company is overstocked with these risk products, i.e., Cap_j is too large. In some cases, we may want to treat Cap_j as a variable. For example, in the travel industry, the product procurement division will seek out contracts with hotels to secure certain numbers of rooms for a given time period. However, if that travel destination is not profitable for the company, they may be better off securing very few rooms or not securing any rooms at all. In all of our models, making Cap_j a variable will not affect the linearity of the constraints. Also, there will most likely be an upperbound for Cap_j for all $j = 1, \ldots, m$. If procuring a unit of product j costs v_j, then the objective function can be modified to:

$$\sum_{i=1}^{n} N_i \sum_{j=1}^{m} \pi_j Pr_{ij} - \sum_{j=1}^{m} v_j Cap_j$$

By determining the optimal value for Cap_j, it should no longer be necessary for the company to penalize under-selling of products[1].

8.2 Product Cost

Suppose each product j has a variable cost of c_j per unit. In the objective function, we want to subtract c_j multiplied by the expected number of customers that buy product j. For all the probabilistic choice models discussed in this thesis, the objective function becomes

$$\sum_i N_i \sum_j (\pi_j - c_j) Pr_{ij}$$

where Pr_{ij} is the probability that the customer segment i buys product j. This is equivalent to lowering all the reservation prices of product j by c_j in all of the models except the Price Sensitive Model.

[1] It is possible that a company may procure large quantities of a currently non-profitable product to increase their long-term market share. We will not consider such long-term marketing strategy in this thesis.

Chapter 9

Computational Results

To compare the empirical performances of the Uniform Distribution, Weighted Uniform, and Share-of-Surplus Models, we solve a set of problem instances using the different formulations. The inputs are subsets of reservation prices estimated from actual booking orders of a travel company (our procedure in estimating reservation prices are discussed in the Appendix). The sizes of the inputs used are from 2 segments and 2 products to up to 100 segments and 100 products. Unfortunately, not all reservation prices can be estimated, so the subsets only contain reservation prices that are available. Due to this restriction, we do not have inputs of sizes larger than 100 segments and products. In the future, we would like to find a better way of estimating the reservation prices so that we can test the models with larger real problems.

The models were run with default parameter settings of CPLEX 9.1 and a time limit of two hours (7200 CPU seconds) unless indicated otherwise. They were run on a machine with four 1.3 GHz Itanium 2 processors and 8 GB of RAM, with at most one process running at a time on each processor.

The tables show the number of segments n and the number of products m in the input, whether an optimal solution was found in the time limit ("Status"), total CPU seconds ("Time"), the objective value ("Objective Value"), total number of dual simplex iterations ("Number of Iterations"), total number of branch-and-bound nodes ("Number of Nodes"), total number of branch-and-bound nodes unvisited ("Number of Nodes Left"), and the optimality gap when CPLEX was terminated ("Gap").

9.1 Uniform Distribution Model

Tables 9.1 and 9.2 show the results of the Uniform Distribution Model (2.1). Tables 9.5 and 9.6 show the results of the alternative formulation (2.2).

The Uniform Distribution Model is surprisingly difficult to solve. For both the original and the alternative formulations, about half of the problem instances could not be solved to optimality in two hours. For the problem instances that were solved to optimality in two hours, the alternative formulation is faster than the original formulation. For the other 17 problem instances, the alternative formulation found a better objective value for 9 of them after 2 hours.

The results of the alternative formulation (2.2) using the results of Heuristic 1 (Section 7.1.2) as the starting solutions are shown in Tables 9.7 and 9.8. The column "Heuristic Obj Val" is the objective value found by Heuristic 1 and "Init. Gap" is the percentage difference between the heuristic's objective value and the best objective value found by CPLEX.

The solutions found by the heuristic are fairly good even though the heuristic is so simple. The largest initial gap is 20.11%, and the initial gaps are all under 3% on Table 9.7. Most notably, the heuristic's solution for the last case ($n = 100$, $m = 100$) is better than all the other solutions found by CPLEX in two hours, and its objective value is much higher than the one found without using a starting solution (Table 9.6). For most of the cases, the best objective values found with the heuristic is at least as good as the ones found without the heuristic. However, the heuristic did not help to find an optimal solution for a problem instance that was not solved to optimality without the heuristic.

9.2 Weighted Uniform Model

The results of solving the problem instances using the Weighted Uniform Model alternate formulation (Formulation 3.2) are shown in Tables 9.9 and 9.10. Same as the Uniform Distribution Model, about half of the problem instances could not be solved to optimality in two hours. The optimality gaps are smaller compared with the Uniform Distribution Model alternative formulation (Tables 9.5 and 9.6).

n	m	CPLEX Status	Time	Obj Val	Number of Iterations	Number of Nodes	Number of Nodes Left	Gap (%)
2	2	Optimal	0.02	2656	13	0	0	0
2	5	Optimal	0.01	121520	71	0	0	0
2	10	Optimal	0	165960	81	0	0	0
2	20	Optimal	0.01	207680	138	0	0	0
2	60	Optimal	0.02	66801	197	0	0	0
2	100	Optimal	0.03	66801	245	0	0	0
5	2	Optimal	0.02	212238	162	7	0	0
5	5	Optimal	0.24	164328	1415	105	0	0
5	10	Optimal	3.00	217832	21703	2119	3	0.01
5	20	Optimal	0.02	124311	292	0	0	0
5	60	Optimal	0.07	377480	564	0	0	0
5	100	Optimal	3.19	319142	3460	732	68	0.01
10	2	Optimal	0.07	560232	446	18	0	0
10	5	Optimal	2.85	325489	18235	1695	1	0.01
10	10	Optimal	2789.23	385511	9167097	1076780	790	0.01
10	20	Feasible	7237.72	560367	12727737	474382	376034	8.25
10	60	Feasible	7275.31	631920	8177609	217996	171964	4.48
10	100	Feasible	7457.28	605906	7174112	638891	415703	0.88

Table 9.1: Uniform Model (1)

n	m	CPLEX Status	Time	Obj Val	Number of Iterations	Number of Nodes	Number of Nodes Left	Gap (%)
20	2	Optimal	0.31	546773	1416	16	0	0
20	5	Optimal	24.72	636451	74407	3755	2	0
20	10	Feasible	7252.61	824247	10656382	223336	136694	7.52
20	20	Feasible	7240.35	883241	6136494	79263	69814	18.82
20	60	Feasible	7239.35	1220410	3434397	30294	25958	10.60
20	100	Feasible	7240.15	992662	2489086	14000	12680	9.76
60	2	Optimal	2.73	1358680	6733	117	1	0
60	5	Optimal	3351.05	1910820	3495195	230052	264	0.01
60	10	Feasible	7238.34	1973310	4487693	36960	30922	18.29
60	20	Feasible	7221.81	2266160	2041911	10131	9111	24.76
60	60	Feasible	7223.33	2857320	589082	1655	1499	27.37
60	100	Feasible	7229.98	3389180	201143	420	419	24.83
100	2	Optimal	7.71	2623610	12879	273	3	0.01
100	5	Feasible	7334.14	2978190	5783913	251626	29031	0.71
100	10	Feasible	7224.75	3225400	2475106	9993	8302	21.70
100	20	Feasible	7213.37	3262130	1139453	1965	1691	36.10
100	60	Feasible	7213.87	4452940	289521	1022	927	33.74
100	100	Feasible	7224.99	3560210	122419	417	414	43.91

Table 9.2: Uniform Model (2)

n	m	CPLEX Status	Time	Obj Val	Number of Iterations	Number of Nodes	Number of Nodes Left	Gap
2	2	Optimal	0.06	2656	11	0	0	0
2	5	Optimal	0.07	121520	71	0	0	0
2	10	Optimal	0.06	165960	80	0	0	0
2	20	Optimal	0.07	207680	94	0	0	0
2	60	Optimal	0.08	66801	195	0	0	0
2	100	Optimal	0.08	66801	140	0	0	0
5	2	Optimal	0.08	212238	145	2	0	0
5	5	Optimal	0.27	164328	1361	89	0	0
5	10	Optimal	3.01	217832	21801	2044	1	0.01
5	20	Optimal	0.08	124311	256	0	0	0
5	60	Optimal	0.11	377480	332	0	0	0
5	100	Optimal	2.20	319142	2865	452	35	0.01
10	2	Optimal	0.13	560232	446	18	0	0
10	5	Optimal	2.72	325489	15979	1502	3	0.01
10	10	Optimal	3181.13	385511	9623983	1299396	1112	0.01
10	20	Feasible	7238.98	561337	12451833	535871	434348	8.03
10	60	Feasible	7278.03	631920	8138257	216657	170901	4.48
10	100	Feasible	7472.39	605906	7717636	739322	559107	1.17

Table 9.3: Uniform Model with Heuristic 1 (1)

n	m	CPLEX Status	Time	Obj Val	Number of Iterations	Number of Nodes	Number of Nodes Left	Gap
20	2	Optimal	0.38	546774	1416	16	0	0
20	5	Optimal	24.81	636451	74407	3755	2	0
20	10	Feasible	7261.61	821689	10595897	230660	144129	7.84
20	20	Feasible	7245.32	873254	6042271	84480	76624	19.83
20	60	Feasible	7244.07	1211212	3175678	26667	23652	11.38
20	100	Feasible	7243.18	992662	2347139	12973	11690	9.77
60	2	Optimal	2.73	1358681	6733	117	1	0
60	5	Optimal	3519.19	1910818	3495195	230052	264	0.01
60	10	Feasible	7252.85	1973166	4180683	35371	29337	17.69
60	20	Feasible	7225.01	2266157	2002222	9331	8320	24.72
60	60	Feasible	7225.97	2856577	636216	1698	1528	27.39
60	100	Feasible	7232.24	3578698	353851	1080	939	20.62
100	2	Optimal	7.57	2623610	12879	273	3	0.01
100	5	Feasible	7339.67	2978189	5751906	248851	30153	0.74
100	10	Feasible	7227.60	3225400	2518973	10220	8497	21.61
100	20	Feasible	7217.26	3310931	1192737	2052	1685	35.14
100	60	Feasible	7221.01	4450962	300963	993	917	33.76
100	100	Feasible	7244.49	4758460	202445	1292	1196	25.03

Table 9.4: Uniform Model with Heuristic 1 (2)

n	m	CPLEX Status	Time	Obj Val	Number Iterations	Number of of Nodes	Number of Nodes Left	Gap (%)
2	2	Optimal	0.009	2656	6	0	0	0
2	5	Optimal	0.007	121520	38	0	0	0
2	10	Optimal	0.005	165960	29	0	0	0
2	20	Optimal	0.008	207680	39	0	0	0
2	60	Optimal	0.022	66801	98	0	0	0
2	100	Optimal	0.048	66801	206	0	0	0
5	2	Optimal	0.013	212238	54	5	0	0
5	5	Optimal	0.077	164328	507	99	1	0.002
5	10	Optimal	2.120	217832	21968	4540	2	0.007
5	20	Optimal	0.024	124311	127	0	0	0
5	60	Optimal	0.059	377480	150	0	0	0
5	100	Optimal	0.906	319142	482	30	5	0.009
10	2	Optimal	0.026	560232	123	7	0	0
10	5	Optimal	1.903	325489	16209	4649	6	0.008
10	10	Optimal	738.667	385511	4902316	1438972	2395	0.010
10	20	Feasible	7249.830	561753	35188051	5116542	4307083	6.538
10	60	Feasible	7280.240	631944	15818206	3233966	2401721	2.485
10	100	Feasible	7309.910	605906	17125414	1307050	841663	0.655

Table 9.5: Uniform Model Alternative Formulation (1)

n	m	CPLEX Status	Time	Obj Val	Number Iterations	Number of of Nodes	Number of Nodes Left	Gap (%)
20	2	Optimal	0.050	546773	152	0	0	0
20	5	Optimal	21.309	636451	141298	31251	40	0.010
20	10	Feasible	7295.710	821823	33441844	6281274	3861280	2.830
20	20	Feasible	7215.740	870620	21312071	1383325	1316170	19.024
20	60	Feasible	7249.030	1214110	10223033	345551	293840	10.959
20	100	Feasible	7280.100	1003350	6759345	318931	280624	5.759
60	2	Optimal	2.009	1358680	5757	420	0	0
60	5	Feasible	7290.210	1908940	22301732	3547946	1857893	2.091
60	10	Feasible	7221.630	1963310	15853218	1413813	1374355	17.695
60	20	Feasible	7240.630	2257700	8799107	304290	294812	21.858
60	60	Feasible	7235.580	2928260	2166352	44619	40270	24.676
60	100	Feasible	7222.360	3527260	931556	6303	3723	21.291
100	2	Optimal	6.722	2623610	20970	1360	0	0
100	5	Feasible	7256.840	2975900	14142345	1966351	1787586	6.783
100	10	Feasible	7261.720	3219900	9349927	694326	672535	19.465
100	20	Feasible	7247.980	3359730	3650764	150810	148601	31.501
100	60	Feasible	7223.770	4780230	780897	6693	5461	28.173
100	100	Feasible	7220.100	2179480	358686	377	378	65.570

Table 9.6: Uniform Model Alternative Formulation (2)

n	m	CPLEX Status	Time	Obj Val	Number of Iterations	Number of Nodes	Number of Nodes Left	Gap (%)	Heuristic Obj Val	Init. Gap
2	2	Optimal	0.06	2656	5	0	0	0	2656.00	0
2	5	Optimal	0.06	121520	37	0	0	0	121520.00	0
2	10	Optimal	0.06	165960	28	0	0	0	165960.00	0
2	20	Optimal	0.06	207680	28	0	0	0	207680.00	0
2	60	Optimal	0.08	66801	96	0	0	0	66801.00	0
2	100	Optimal	0.10	66801	125	0	0	0	66801.00	0
5	2	Optimal	0.07	212238	40	1	0	0	212238.00	0
5	5	Optimal	0.15	164328	501	99	1	0.002	164038.67	0.18
5	10	Optimal	1.61	217832	14520	3066	2	0.007	217832.00	0
5	20	Optimal	0.10	124311	124	0	0	0	124311.00	0
5	60	Optimal	0.17	377480	146	0	0	0	377480.00	0
5	100	Optimal	1.17	319142	537	27	3	0.007	318770.00	0.12
10	2	Optimal	0.08	560232	123	7	0	0	543760.00	2.94
10	5	Optimal	1.80	325489	14702	4473	7	0.008	323975.50	0.46
10	10	Optimal	478.43	385511	3099905	976193	1854	0.010	379850.67	1.47
10	20	Feasible	7242.66	562543	36883946	4521011	3838288	7.005	555828.57	1.19
10	60	Feasible	7256.94	631920	16801245	3206741	2375616	2.272	624070.00	1.24
10	100	Feasible	7371.20	605906	14817894	2201874	1495908	0.690	605906.00	0

Table 9.7: Uniform Alternative Formulation with Heuristic (1)

n	m	CPLEX Status	Time	Obj Val	Number of Iterations	Number of Nodes	Number of Nodes Left	Gap (%)	Heuristic Obj Val	Init. Gap
20	2	Optimal	0.13	546773	152	0	0	0	544933.00	0.34
20	5	Optimal	22.45	636451	148607	33304	50	0.010	597751.73	6.08
20	10	Feasible	7277.33	826388	33346360	5977052	2050909	1.178	746544.34	9.66
20	20	Feasible	7218.68	871438	20586030	1288903	1232530	19.083	823205.84	5.53
20	60	Feasible	7263.70	1210300	9038508	435610	404399	11.199	1174843.45	2.93
20	100	Feasible	7263.57	1003460	6982060	250213	214378	5.949	868826.67	13.42
60	2	Optimal	2.10	1358680	5757	420	0	0	1228719.00	9.57
60	5	Feasible	7291.07	1910760	21893848	3472747	1636867	1.772	1526583.90	20.11
60	10	Feasible	7221.03	1942390	15824025	1460216	1417895	18.640	1770828.61	8.83
60	20	Feasible	7237.20	2269720	8570162	290388	280059	21.442	1957967.13	13.74
60	60	Feasible	7254.81	2911460	2180690	43495	36861	25.113	2476842.23	14.93
60	100	Feasible	7274.08	3509440	996247	6782	5251	21.663	3218883.70	8.28
100	2	Optimal	6.78	2623610	20970	1360	0	0	2368923.50	9.71
100	5	Feasible	7251.92	2963210	14311720	2001758	1840599	7.433	2493506.73	15.85
100	10	Feasible	7253.18	3253390	9416746	644540	621411	18.704	2703764.17	16.89
100	20	Feasible	7248.55	3396390	3513959	142581	140405	30.802	2736795.27	19.42
100	60	Feasible	7242.20	4903330	785803	5144	4196	26.295	4128380.75	15.80
100	100	Feasible	7333.07	4380590	375860	659	605	30.798	4380586.63	0

Table 9.8: Uniform Alternative Formulation with Heuristic (2)

n	m	CPLEX Status	Time	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap
2	2	Optimal	0.010	2656	4	0	0	0
2	5	Optimal	0.007	121520	42	0	0	0
2	10	Optimal	0.005	165960	22	0	0	0
2	20	Optimal	0.011	207680	24	0	0	0
2	60	Optimal	0.027	66801	41	0	0	0
2	100	Optimal	0.042	66801	39	0	0	0
5	2	Optimal	0.008	216338	42	0	0	0
5	5	Optimal	0.069	165148	436	115	0	0
5	10	Optimal	3.211	219925	39592	7077	10	0.009
5	20	Optimal	0.025	124311	62	0	0	0
5	60	Optimal	0.067	377480	111	0	0	0
5	100	Optimal	0.547	319192	420	69	8	0.010
10	2	Optimal	0.027	560991	101	9	0	0
10	5	Optimal	1.314	327000	11085	3520	3	0.009
10	10	Optimal	472.095	387000	3165547	968679	1571	0.010
10	20	Feasible	7261.740	565445	37926349	6364811	5439663	6.677
10	60	Feasible	7382.970	639020	18502911	3595605	2263154	1.356
10	100	Feasible	7411.000	605924	15518283	2088360	1123705	0.575

Table 9.9: Weighted Uniform Alternative Formulation (1)

n	m	CPLEX Status	Time	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap
20	2	Optimal	0.064	547088	147	2	0	0
20	5	Optimal	20.426	636940	135886	31914	36	0.010
20	10	Feasible	7279.800	826632	35940853	5420698	3795875	4.246
20	20	Feasible	7221.450	872954	25070512	1744665	1693375	19.076
20	60	Feasible	7347.510	1235430	10506427	990001	945836	8.711
20	100	Feasible	7437.890	1022360	9108317	638577	546293	4.271
60	2	Optimal	1.550	1359510	4894	684	0	0
60	5	Feasible	7298.230	1915680	23198431	3733640	1504625	1.444
60	10	Feasible	7224.970	1994670	15131234	1242080	1163999	15.197
60	20	Feasible	7229.770	2305230	8634225	381480	373301	20.172
60	60	Feasible	7252.620	2988900	2504747	105419	101468	18.636
60	100	Feasible	7259.140	3636920	1250190	40587	39000	14.323
100	2	Optimal	5.692	2623900	17588	2030	0	0
100	5	Feasible	7253.330	2970500	13911585	2424888	2252629	7.967
100	10	Feasible	7273.250	3240780	9749274	689954	670640	18.802
100	20	Feasible	7256.960	3479020	5137320	167759	164638	25.400
100	60	Feasible	7240.230	4909840	1081647	33629	31731	19.097
100	100	Feasible	7244.340	4801770	582742	20574	20184	16.530

Table 9.10: Weighted Uniform Alternative Formulation (2)

9.3 Share-of-Surplus Model

Tables 9.11 and 9.12 show the results of the Share-of-Surplus Model with restricted prices formulation (4.9) where $c = 1$.

It seems that the Share-of-Surplus Model is very difficult to solve. No mixed-integer feasible solutions were found for the larger problem instances because the LP relaxation could not be solved in two hours. From the CPLEX outputs when solving the LP relaxation, we noticed many times that there were unscaled infeasibility and CPLEX takes a long time to try to resolve it. CPLEX's preprocessor scales the rows of the mixed-integer programming formulation before solving it, and unscaled infeasibility occurs if the optimal solution found for the scaled problem is not feasible for the original problem. This seems to imply that our problem is ill-conditioned. Consider the constraints in the formulation (4.9). The reservation prices in the problem instances are generally in the range of 500 to 1500. That is, the coefficients of some of the variables are more than 1500 times the coefficients of other variables, making the problem quite ill-conditioned.

We can attempt to solve this problem by scaling the reservation prices before using them in the model since the optimal solution is the same regardless of the unit the reservation prices are in. We let the parameter R to be $R_{ij} := \frac{r_{ij}}{s}$ where r_{ij} is the original reservation price and s is the scale used.

To use the scaled R's in the formulation, we need to replace the following constraint from formulation (4.9)

$$t_i \leq (\max_k R_{ik}) \sum_{j=1}^{m} \sum_{l:R_{lj} \leq R_{ij}} (R_{ij} - R_{lj} + c)x_{lj}, \forall i$$

with the constraint

$$t_i \leq s(\max_k R_{ik}) \sum_{j=1}^{m} \sum_{l:R_{lj} \leq R_{ij}} (R_{ij} - R_{lj} + \frac{c}{s})x_{lj}, \forall i.$$

If no scaling is done, then $s = 1$ and the two constraint are the same.

Tables 9.13 to 9.20 show the results with scaled reservation prices. They include the scale used ("Scale") and the objective values without scaling ("Obj Val w/o Scale") which is the objective value multiplied by the scale. The reservation prices are scaled by 100 in

n	m	CPLEX Status	Time	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap
2	2	Optimal	0.003	2656.00	10	0	0	0.00
2	5	Optimal	0.008	121520.00	45	0	0	0.00
2	10	Optimal	0.006	165960.00	33	0	0	0.00
2	20	Optimal	0.010	207680.00	55	0	0	0.00
2	60	Optimal	0.033	66801.00	178	0	0	0.00
2	100	Optimal	0.052	66801.00	230	0	0	0.00
5	2	Optimal	0.030	188262.33	174	14	0	0.00
5	5	Optimal	0.233	156227.63	1355	173	0	0.00
5	10	Optimal	12.493	200113.95	91471	15219	6	0.01
5	20	Optimal	0.054	124311.00	186	0	0	0.00
5	60	Optimal	0.226	377480.00	371	0	0	0.00
5	100	Optimal	196.712	314096.48	63338	24343	13332	0.01
10	2	Optimal	0.254	547230.84	1099	44	0	0.00
10	5	Optimal	7.867	318568.75	41285	3391	1	0.01
10	10	Optimal	4974.726	353560.05	18591446	1156595	488	0.01
10	20	Feasible	7238.979	522364.84	11137946	638525	551029	14.17
10	60	Feasible	7421.309	595944.46	2394669	23646	21987	13.74
10	100	Feasible	7398.787	550178.41	1291597	5457	3958	12.28

Table 9.11: Share-of-Surplus (1)

n	m	CPLEX Status	Time	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap
20	2	Optimal	2.869	544852.40	7378	113	0	0.00
20	5	Optimal	1650.083	603613.93	3236589	132607	44	0.01
20	10	Feasible	7234.299	743424.33	5653478	66730	60424	24.54
20	20	Feasible	7216.812	737637.11	1929323	17840	17042	36.68
20	60	Feasible	7319.291	109891.44	118185	164	165	92.28
20	100	Feasible	7244.623	78106.70	104593	0	1	92.98
60	2	Optimal	1167.595	1347710.18	618883	2234	1	0.01
60	5	Feasible	7204.174	1555525.57	472679	84	85	39.07
60	10	Feasible	7202.876	833661.00	184065	0	1	69.98
60	20	No solutions found						
60	60	No solutions found						
60	100	No solutions found						
100	2	Feasible	7207.918	2617144.84	638357	535	419	17.11
100	5	Feasible	7201.978	2701523.77	139184	0	1	34.75
100	10	No solutions found						
100	20	No solutions found						
100	60	No solutions found						
100	100	No solutions found						

Table 9.12: Share-of-Surplus (2)

the Tables 9.13 and 9.14. They are scaled so that the maximum scaled reservation price is 1 in the Tables 9.15 and 9.16, i.e., the scale is the largest original reservation price. In Tables 9.17 and 9.18, the maximum reservation price is 10. The results with the maximum reservation price equals to 10 and rounded to the nearest integer are shown in Tables 9.19 and 9.20.

The scaled instances improve the gap of the best solution by a little for most cases, but not significantly. The larger problems instances still could not be solved in two hours.

To better understand the problem, we ran the LP relaxation of the model with different scaling for 11 of the problem instances. They were run with default parameter settings of CPLEX 9.1 and a time limit of 24 hours. The results are shown in Tables 9.21 and 9.22. The first column indicates the scaling method used: "100" means the scale used is 100, "maxR=1" and "maxR=10" indicates the maximum reservation price is 1 and 10 respectively, and "maxR=10 ro" means the maximum reservation price is 10 and then rounded to the nearest integer. The "CPLEX Status" is *Unknown* indicates that the problem instance could not be solved in 24 hours.

Running the problems without scaling seems to have the worst performance since it results in the worst runtime for 6 of the 8 problem instances that were solved to optimality. On the other hand, scaling such that the maximum reservation price is 1 seems to perform the best. It was the only scaling that could solve the problem instances (60×60) and (100×20) to optimality in 24 hours. It has the fastest runtime for 6 cases and the second fastest runtime for 3 other cases. The improvement is perhaps due to the fact that the scaled reservation prices are closer to 1, hence the coefficients of the x variables are closer to 1, which is the coefficient of many variables in the formulation. Unfortunately, the optimal solution for the ($100x100$) problem instance could not be found with any of the scalings.

These results show that solving the LP relaxation of the Share-of-Surplus Model is a key bottleneck for solving the MIP. It seems like scaling shortens the solution time for the LP relaxation by a little. Clearly, further investigation is required to determine the best scaling strategy to help solve the LP. We also need to study the structure of the LP to determine additional causes of the computational difficulty.

n	m	CPLEX Status	Time	Obj Val w/o Scale	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap
2	2	Optimal	0.003	2656.00	26.56	13	0	0	0.00
2	5	Optimal	0.008	121520.00	1215.20	48	0	0	0.00
2	10	Optimal	0.006	165960.00	1659.60	29	0	0	0.00
2	20	Optimal	0.010	207680.00	2076.80	55	0	0	0.00
2	60	Optimal	0.032	66801.00	668.01	170	0	0	0.00
2	100	Optimal	0.055	66801.00	668.01	266	0	0	0.00
5	2	Optimal	0.026	188262.33	1882.62	178	14	0	0.00
5	5	Optimal	0.180	156227.63	1562.28	957	108	0	0.00
5	10	Optimal	11.190	200113.95	2001.14	78978	14870	6	0.01
5	20	Optimal	0.051	124311.00	1243.11	163	0	0	0.00
5	60	Optimal	0.204	377480.00	3774.80	401	0	0	0.00
5	100	Optimal	74.669	314096.48	3140.96	40180	3226	1217	0.01
10	2	Optimal	0.236	547230.84	5472.31	1089	47	0	0.00
10	5	Optimal	8.763	318568.75	3185.69	41052	3731	1	0.00
10	10	Optimal	4920.005	353560.05	3535.60	16648615	1334858	622	0.01
10	20	Feasible	7247.964	522573.63	5225.74	12330270	821723	714028	13.24
10	60	Feasible	7333.222	595031.83	5950.32	2559793	31151	19144	13.70
10	100	Feasible	7516.007	550176.66	5501.77	1349279	21509	16851	11.68

Table 9.13: Share-of-Surplus, reservation prices scaled by 100 (1)

n	m	CPLEX status	Time	Obj val w/o scale	Obj val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap
20	2	Optimal	2.727	544852.40	5448.52	7435	122	0	0.00
20	5	Optimal	1720.597	603613.93	6036.14	3003469	129654	28	0.01
20	10	Feasible	7234.556	759072.02	7590.72	5786176	71796	63793	23.13
20	20	Feasible	7216.712	741360.40	7413.60	1977868	22423	21479	36.37
20	60	Feasible	7261.751	152840.00	1528.40	88149	10	11	89.26
20	100	Feasible	7223.192	96544.00	965.44	83963	0	1	91.33
60	2	Optimal	981.469	1347710.18	13477.10	586240	2217	0	0.00
60	5	Feasible	7208.297	1799277.94	17992.78	628742	989	915	29.48
60	10	Feasible	7205.185	1145996.94	11459.97	74661	0	1	58.74
60	20	No solutions found							
60	60	No solutions found							
60	100	No solutions found							
100	2	Feasible	7207.671	2537308.99	25373.09	635508	583	462	20.98
100	5	Feasible	7204.616	2701523.77	27015.24	101638	0	1	34.75
100	10	No solutions found							
100	20	No solutions found							
100	60	No solutions found							
100	100	No solutions found							

Table 9.14: Share-of-Surplus, reservation prices scaled by 100 (2)

n	m	CPLEX Status	Time	Obj Val w/o Scale	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap	Scale
2	2	Optimal	0.003	2656.00	2.50	10	0	0	0	1064
2	5	Optimal	0.009	121520.00	77.90	60	0	0	0	1560
2	10	Optimal	0.006	165960.00	63.22	58	0	0	0	2625
2	20	Optimal	0.011	207680.00	61.72	70	0	0	0	3365
2	60	Optimal	0.028	66801.00	28.10	144	0	0	0	2377
2	100	Optimal	0.048	66801.00	28.10	200	0	0	0	2377
5	2	Optimal	0.032	188262.33	80.21	236	16	0	0	2347
5	5	Optimal	0.158	156227.63	94.68	795	101	0	0	1650
5	10	Optimal	12.553	200113.95	104.23	92693	17830	6	0.01	1920
5	20	Optimal	0.072	124311.00	52.30	339	0	0	0	2377
5	60	Optimal	0.181	377480.00	139.76	516	0	0	0	2701
5	100	Optimal	190.210	314096.48	82.20	45447	18300	6858	0.01	3821
10	2	Optimal	0.214	547230.84	210.72	967	52	0	0	2597
10	5	Optimal	8.063	318568.75	174.46	36274	3275	1	0.01	1826
10	10	Optimal	4627.430	353560.05	165.21	16782955	1429172	618	0.01	2140
10	20	Feasible	7283.021	522629.32	193.49	13566471	1002368	894043	13.53	2701
10	60	Feasible	7388.581	596020.80	216.11	2823045	31714	26768	13.46	2758
10	100	Feasible	7637.544	550147.13	143.98	1394015	16724	14012	11.87	3821

Table 9.15: Share-of-Surplus, the maximum reservation price is 1. (1)

n	m	CPLEX Status	Time	Obj Val w/o Scale	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap	Scale
20	2	Optimal	2.041	544852.40	342.03	5823	119	0	0	1593
20	5	Optimal	1584.893	603613.93	329.48	2733059	133057	39	0.01	1832
20	10	Feasible	7235.973	742365.41	319.85	5755205	74990	66480	24.03	2321
20	20	Feasible	7218.123	780306.73	256.85	1786584	17275	16570	33.20	3038
20	60	Feasible	7264.515	231840.00	60.68	124889	270	271	83.71	3821
20	100	Feasible	7249.705	96544.00	25.27	88214	0	1	91.33	3821
60	2	Optimal	836.968	1347710.18	777.23	523024	2237	0	0	1734
60	5	Feasible	7208.292	1844218.81	842.88	586540	920	843	27.74	2188
60	10	Feasible	7204.123	1145996.94	493.75	67089	0	1	58.74	2321
60	20	No solutions found								
60	60	No solutions found								
60	100	No solutions found								
100	2	Feasible	7211.785	2588279.64	1492.66	718432	1602	1273	19.90	1734
100	5	Feasible	7204.581	2701523.77	1022.53	81562	0	1	34.77	2642
100	10	No solutions found								
100	20	No solutions found								
100	60	No solutions found								
100	100	No solutions found								

Table 9.16: Share-of-Surplus, the maximum reservation price is 1. (2)

n	m	CPLEX Status	Time	Obj Val w/o Scale	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap	Scale
2	2	Optimal	0.003	2656.00	24.96	13	0	0	0.00	106.4
2	5	Optimal	0.008	121520.00	778.97	43	0	0	0.00	156.0
2	10	Optimal	0.006	165960.00	632.23	33	0	0	0.00	262.5
2	20	Optimal	0.010	207680.00	617.18	55	0	0	0.00	336.5
2	60	Optimal	0.031	66801.00	281.03	175	0	0	0.00	237.7
2	100	Optimal	0.053	66801.00	281.03	273	0	0	0.00	237.7
5	2	Optimal	0.034	188262.33	802.14	239	18	0	0.00	234.7
5	5	Optimal	0.177	156227.63	946.83	954	106	0	0.00	165.0
5	10	Optimal	11.212	200113.95	1042.26	80609	16181	7	0.01	192.0
5	20	Optimal	0.051	124311.00	522.97	150	0	0	0.00	237.7
5	60	Optimal	0.245	377480.00	1397.56	455	0	0	0.00	270.1
5	100	Optimal	129.728	314096.48	822.03	38668	14017	7323	0.01	382.1
10	2	Optimal	0.235	547230.84	2107.17	1007	49	0	0.00	259.7
10	5	Optimal	8.679	318568.75	1744.63	41322	3585	2	0.01	182.6
10	10	Optimal	5180.500	353560.05	1652.15	17079760	1461167	587	0.01	214.0
10	20	Feasible	7250.343	522179.67	1933.28	10456907	503699	408902	13.19	270.1
10	60	Feasible	7395.238	596044.21	2161.15	2714929	29712	25282	13.40	275.8
10	100	Feasible	7533.797	550177.30	1439.88	1354189	27122	22210	11.58	382.1

Table 9.17: Share-of-Surplus, the maximum reservation price is 10. (1)

n	m	CPLEX Status	Time	Obj Val w/o scale	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap	Scale
20	2	Optimal	2.811	544852.40	3420.29	7748	123	1	0.00	159.3
20	5	Optimal	1773.690	603613.93	3294.84	3048780	134206	37	0.01	183.2
20	10	Feasible	7234.025	743310.20	3202.54	5752233	67864	60306	24.44	232.1
20	20	Feasible	7217.175	760461.33	2503.16	2089323	17030	16149	34.80	303.8
20	60	Feasible	7265.950	152840.00	400.00	119785	182	183	89.26	382.1
20	100	Feasible	7253.102	78106.70	204.41	69197	0	1	92.98	382.1
60	2	Optimal	1061.125	1347710.18	7772.26	609752	2210	2	0.01	173.4
60	5	Feasible	7208.166	1780929.79	8139.53	618179	1000	891	30.22	218.8
60	10	Feasible	7201.985	714525.00	3078.52	160062	0	1	74.27	232.1
60	20		No solutions found							
60	60		No solutions found							
60	100		No solutions found							
100	2	Feasible	7208.812	2558417.99	14754.43	648189	414	350	20.94	173.4
100	5	Feasible	7203.587	2701523.77	10225.30	88266	0	1	34.76	264.2
100	10		No solutions found							
100	20		No solutions found							
100	60		No solutions found							
100	100		No solutions found							

Table 9.18: Share-of-Surplus, the maximum reservation price is 10. (2)

n	m	CPLEX Status	Time	Obj Val w/o Scale	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap	Scale
2	2	Optimal	0.010	2660.00	25.00	13	0	0	0	106.4
2	5	Optimal	0.003	124800.00	800.00	17	0	0	0	156.0
2	10	Optimal	0.005	168000.00	640.00	27	0	0	0	262.5
2	20	Optimal	0.009	201900.00	600.00	54	0	0	0	336.5
2	60	Optimal	0.029	64892.10	273.00	131	0	0	0	237.7
2	100	Optimal	0.064	64892.10	273.00	308	0	0	0	237.7
5	2	Optimal	0.049	197148.00	840.00	340	20	0	0	234.7
5	5	Optimal	0.199	149100.73	903.64	1333	149	0	0	165.0
5	10	Optimal	14.090	204430.82	1064.74	109513	14282	3	0.008	192.0
5	20	Optimal	0.836	117265.33	493.33	1129	202	0	0	237.7
5	60	Optimal	0.183	378140.00	1400.00	367	0	0	0	270.1
5	100	Optimal	873.327	301970.81	790.29	1297787	52288	640	0.010	382.1
10	2	Optimal	0.362	579052.55	2229.70	2168	140	0	0	259.7
10	5	Optimal	13.357	324211.33	1775.53	58199	8404	1	0.004	182.6
10	10	Feasible	7255.469	348881.67	1630.29	26576525	2032571	1049733	6.010	214.0
10	20	Feasible	7239.328	512733.88	1898.31	11793024	620950	534734	13.733	270.1
10	60	Feasible	7712.828	640245.94	2321.41	2983122	12152	6454	7.698	275.8
10	100	Feasible	7672.237	537996.80	1408.00	1889087	19398	15454	13.231	382.1

Table 9.19: Share of Surplus, the maximum reservation price is 10 and rounded. (1)

n	m	CPLEX Status	Time	Obj Val w/o Scale	Obj Val	Number of Iter	Number of Nodes	Number of Nodes Left	Gap	Scale
20	2	Optimal	6.620	535122.67	3359.21	21795	413	0	0	159.3
20	5	Optimal	2113.888	628452.71	3430.42	4985376	359870	11	0.009	183.2
20	10	Feasible	7232.234	793271.00	3417.80	5559267	51842	46660	21.121	232.1
20	20	Feasible	7322.441	825107.48	2715.96	2159065	17868	16962	29.219	303.8
20	60	Feasible	7350.367	239555.72	626.95	68000	7	8	82.803	382.1
20	100	Feasible	7406.689	74763.08	195.66	79617	0	1	93.107	382.1
60	2	Optimal	2070.205	1366263.05	7879.26	1258689	4181	0	0	173.4
60	5	Feasible	7215.075	1880038.45	8592.50	487379	403	376	26.251	218.8
60	10	Feasible	7214.648	785948.72	3386.25	94813	0	1	71.860	232.1
60	20	No solutions found								
60	60	No solutions found								
60	100	No solutions found								
100	2	Feasible	7212.891	2743817.20	15823.63	608674	882	731	16.031	173.4
100	5	No solutions found								
100	10	No solutions found								
100	20	No solutions found								
100	60	No solutions found								
100	100	No solutions found								

Table 9.20: Share of Surplus, the maximum reservation price is 10 and rounded. (2)

n	m	Scaling	CPLEX Status	Time	Obj Val w/o Scale	Obj Val	# Iter.	Scale
		No scale	Optimal	0.37	491649	491649.03	1063	1.0
		100	Optimal	0.29	491649	4916.49	835	100.0
10	10	maxR=1	Optimal	0.29	491649	229.74	821	2140.0
		maxR=10	Optimal	0.26	491649	2297.43	793	214.0
		maxR=10 ro	Optimal	0.35	501474	2343.34	949	214.0
		No scale	Optimal	1.72	674610	674610.01	2239	1.0
		100	Optimal	0.78	674610	6746.10	1797	100.0
10	20	maxR=1	Optimal	1.09	674610	249.76	1614	2701.0
		maxR=10	Optimal	0.88	674610	2497.63	1600	270.1
		maxR=10 ro	Optimal	0.95	678427	2511.76	2156	270.1
		No scale	Optimal	9.20	693731	693730.60	3714	1.0
		100	Optimal	7.79	693731	6937.31	3320	100.0
10	60	maxR=1	Optimal	7.45	693732	251.53	3588	2758.0
		maxR=10	Optimal	10.90	693731	2515.34	4445	275.8
		maxR=10 ro	Optimal	10.49	709322	2571.87	4247	275.8
		No scale	Optimal	5.72	1060889	1060888.61	5332	1.0
		100	Optimal	4.03	1060889	10608.89	3230	100.0
20	10	maxR=1	Optimal	3.92	1060889	457.08	3325	2321.0
		maxR=10	Optimal	3.91	1060889	4570.83	3000	232.1
		maxR=10 ro	Optimal	4.55	1061492	4573.43	3612	232.1
		No scale	Optimal	430.69	1425113	1425113.18	22012	1.0
		100	Optimal	536.07	1425113	14251.13	29008	100.0
20	60	maxR=1	Optimal	248.86	1425113	372.97	15040	3821.0
		maxR=10	Optimal	300.93	1425113	3729.69	17854	382.1
		maxR=10 ro	Optimal	396.77	1394647	3649.95	23526	382.1
		No scale	Optimal	1586.71	1113668	1113668.06	43259	1.0
		100	Optimal	990.64	1113668	11136.68	29600	100.0
20	100	maxR=1	Optimal	979.17	1113668	291.46	31591	3821.0
		maxR=10	Optimal	1301.03	1113668	2914.60	38829	382.1
		maxR=10 ro	Optimal	1302.59	1085155	2839.98	39565	382.1

Table 9.21: Share-of-Surplus LP Relaxation (1)

n	m	Scaling	CPLEX Status	Time	Obj Val w/o Scale	Obj Val	# Iter.	Scale
		No scale	Optimal	14410.04	3193481	3193480.95	191793	1.0
		100	Optimal	4305.10	3193481	31934.81	75242	100.0
60	20	maxR=1	Optimal	4541.50	3193481	828.40	79255	3855.0
		maxR=10	Optimal	10313.59	3193481	8284.00	160262	385.5
		maxR=10 ro	Optimal	5350.56	3226598	8369.91	88228	385.5
		No scale	Unknown	86441.56	–	–	305500	1.0
		100	Unknown	86734.86	–	–	354500	100.0
60	60	maxR=1	Optimal	75188.34	3990330	860.73	291119	4636.0
		maxR=10	Unknown	86545.40	–	–	323100	463.6
		maxR=10 ro	Unknown	88165.56	–	–	346400	463.6
		No scale	Optimal	50881.92	4835642	4835642.06	450708	1.0
		100	Optimal	15466.55	4835642	48356.42	167898	100.0
100	10	maxR=1	Optimal	11078.28	4835642	1315.82	141522	3675.0
		maxR=10	Optimal	26565.45	4835642	13158.21	171992	367.5
		maxR=10 ro	Optimal	25875.97	4865261	13238.80	230320	367.5
		No scale	Unknown	87733.20	–	–	424900	1.0
		100	Unknown	86407.83	–	–	401300	100.0
100	20	maxR=1	Optimal	71195.08	5306483	1127.36	321094	4707.0
		maxR=10	Unknown	88385.15	–	–	373700	470.7
		maxR=10 ro	Unknown	88625.64	–	–	220800	470.7
		No scale	Unknown	89220.89	–	–	296800	1.0
		100	Unknown	86497.65	–	–	246600	100.0
100	100	maxR=1	Unknown	86494.29	–	–	295200	5680.0
		maxR=10	Unknown	86523.08	–	–	322700	568.0
		maxR=10 ro	Unknown	86434.80	–	–	189500	568.0

Table 9.22: Share-of-Surplus LP Relaxation (2)

Chapter 10

Conclusion

This thesis presents ways to formulate and solve product pricing models using mathematical programming. We have discussed four different probabilistic choice models, all of which are based on reservation prices and are formulated as convex mixed-integer programming problems. The Uniform Distribution Model assumes that Pr_{ij}, the probability that segment i buys product j, is uniform among all products with nonnegative surplus. The Weighted Uniform Model assumes that Pr_{ij} is proportional to the reservation price R_{ij}. In the Share-of-Surplus Model, the probability Pr_{ij} depends on the surplus of the products. Using the assumption that demand increases as price decreases, the Price Sensitive Model uses Pr_{ij} that is inversely proportional to the price of the products with nonnegative surplus. A few special properties of the models have been shown and comparisons of the models' optimal solutions provide some indication of how the models behave. We have proposed and tested a few simple heuristics for finding feasible solutions and we conclude that using a starting feasible solution found by the heuristic does improve the solution time. Computational results of the various models are also presented and they show that the proposed models are difficult to solve for larger problems.

Further research is needed to develop better heuristics (perhaps heuristics tailored to each model) to find a good starting solution to improve the solution time. More investigations on different cuts should also be done, especially on the valid inequalities discussed in Section 7.2.

For the Share-of-Surplus Model, we may want to investigate other monotonically increasing functions to describe the probability which would perhaps lead to formulations that are easier to solve. We may like to examine the effect of the value of the constant c on the problem (4.9) and determine the ideal value for the constant. In addition, we currently do not fully understand the effect of scaling the reservation prices and this area should be explored further.

All the models discussed in this thesis assume that the company has no competitors. We should explore ways to consider competitor products in our models in order to correctly model the loss of revenue when the customers buy from other companies. We can easily incorporate competitor products in our formulations by considering the surplus of every segment for every competitor product. However, this may unrealistically increase the denominator of Pr_{ij} and collecting such detailed competitor information is very difficult. The challenge is to determine how to include competitor information without explicitly considering each competitor product individually.

The motive of this thesis is to show how some marketing models of customer choice behavior can be modelled exactly using mixed-integer programming. This preliminary work illustrates the modeling power of integer and convex nonlinear programming techniques, and we hope to extend our work to other product pricing and customer choice models in the future.

Appendix A

Estimating the Reservation Price

The reservation price data used in the computational experiments of Chapter 9 are estimated from actual purchase orders of a Canadian travel company. The customers are partitioned into segments according to their demographic information, purchase lead time and other characteristics. Suppose after the segmentation, there are n customers, with N_i customers in segment i, $i = 1, \ldots, n$. The company offers m products.

From the historical data, we know what fraction of customers of each segment purchased each product and how much they paid for it. Let

$$\begin{aligned} fr_{ij} &:= \text{ the fraction of segment } i \text{ customers who purchased product } j, \\ B_i &:= \{j | fr_{ij} > 0\} \text{ , i.e., set of products purchased by segment } i, \\ p_{ij} &:= \text{ the price that customers of segment } i \text{ paid for product } j. \end{aligned}$$

The price paid for a particular product may be slightly different from customer to customer depending on the time of sales and other anomalies. Thus, the above p_{ij} value is the average price paid by segment i for product j.

To estimate the reservation price R_{ij} of segment i for product j, we assumed that customers behaved according to the Share-of-Surplus Model of Chapter 4. Thus, fr_{ij} should be approximately equal to

$$\frac{R_{ij} - p_{ij}}{\sum_{k \in B_i} R_{ik} - p_{ik}}$$

where R_{ij}'s are now variables and p_{ij}'s are data.

We fit R_{ij}'s and the Share-of-Surplus Model to the data using least squares regression, i.e., for each segment i, we solved for R_{ij}'s, $j = 1, \ldots, m$, that minimizes

$$\sum_{j \in B_i} \left(f_{ij} - \frac{R_{ij} - p_{ij}}{\sum_{k \in B_i} R_{ik} - p_{ik}} \right)^2$$

or

$$\sum_{j \in B_i} \left(f_{ij} (\sum_{k \in B_i} R_{ik} - p_{ik}) - R_{ij} - p_{ij} \right)^2$$

subject to

$$\begin{array}{rcl} R_{ij} - p_{ij} & \geq & 0, \quad j \in B_i, \\ \sum_{k \in B_i} R_{ik} - p_{ik} & \geq & \delta \end{array}$$

where $\delta > 0$.

There are some further details that need to be addressed. One of the key issues is estimating R_{ij} for $j \notin B_i$. Currently, we have these R_{ij}'s set to 0, which is clearly an underestimate. Although we do not have any direct information about segment i's preference level of product j, we may be able to infer this from other segments that do purchase product j. As a future work, we can consider using data mining techniques such as clustering and collaborative filtering to determine these R_{ij}'s.

Bibliography

[1] E. Balas. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *1997,SIAM J. Alg. Disc. Meth.*, 6:466–486, 1985.

[2] G. Dobson and S. Kalish. Positioning and pricing a product line. *Marketing Science*, 7:107–125, 1988.

[3] G. Dobson and S. Kalish. Heuristics for pricing and positioning a product-line using conjoint and cost data. *Management Science*, 39:160–175, 1993.

[4] W. Hanson and R.K. Martin. Optimal bundle pricing. *Management Science*, 36:155–174, 1990.

[5] W.A. Kamakura and G.J. Russell. A probabilistic choice model for market segmentation and elasticity structure. *Journal of Marketing Research*, 26:379–390, 1989.

[6] U.G. Kraus and C.A. Yano. Product line selection and pricing under a share-of-surplus choice model. *European Journal of Operational Research*, 150:653–671, 2003.

[7] J. I. McGill and G. J. Van Ryzin. Revenue management: Research overview and prospects. *Transportation Science*, 33:233–256, 1999.

[8] G. L. Nemhauser and L.A. Wolsey. *Integer Programming and Combinatorial Optimization*. John Wiley and Sons, 1999.

[9] G. Van Ryzin and K.T. Talluri. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, 2004.