Indian School of Business

Essays in Revenue Management

by

Asrar Ahmed

A dissertation submitted

to the
Operations Management Department
Indian School of Business

November 2024

Committee Members

Name	**Signature**
Milind Sohoni (Chair)	____________________
Sumit Kunnumkal (Member)	____________________
Chaithanya Bandi (Member)	____________________
Vishwakant Malladi (Member)	____________________

Abstract

Revenue management is the strategic utilization of resources to maximize revenue and is widely studied across industries, ranging from online platforms to retail to transportation. The three essays in this thesis focus on two distinct but related problems within revenue management: assortment optimization and network capacity control. Together, these essays provide insights into how novel models and optimization techniques can be leveraged to maximize revenue in different contexts.

In the first essay, I consider the problem faced by an online service platform that matches suppliers with consumers. Unlike traditional matching models, which treat them as passive participants, I allow both sides of the market to exercise their choices. To model this setting, I introduce a two-sided assortment optimization model wherein each participant's choice is modeled using a multinomial logit choice function, and the platform's objective is to maximize its expected revenue. I first show that the problem is NP-hard even when the number of suppliers is limited to two and provide a mixed-integer linear programming formulation. Next, I discuss two simple greedy heuristics and argue that these can lead to arbitrarily bad solutions. I then develop relaxations that provide upper and lower bounds and investigate the tightness of these relaxations by obtaining parametric approximation guarantees. Finally, I present numerical results on synthetic data demonstrating the practical utility of these relaxations.

In the second essay, I consider discrete fractional programs, which are extensively used to model assortment optimization problems. I bridge the gap between the discrete and continuous fractional programming literature by solving a class of 0-1 fractional programs as continuous fractional programs. Specifically, I consider 0-1 linear fractional programs under cardinality-type constraints and provide a continuous reformulation with integral maxima, albeit with a higher number of ratio terms. Therefore, I consider the direct relaxation and use the insights from the reformulation to show that the resultant fractional solution can be rounded off with a parametric guarantee. As applications, I develop a Lagrange relaxation-based upper bound solution for assortment optimization under the mixture-of-multinomial logit model and show that it improves upon the existing discretization-based approach. I then derive, as corollaries, tighter parametric

bounds for a class of assortment optimization problems. Additionally, I illustrate that the reformulation can help improve the discrete local search heuristic solution by exploiting the continuous solution space. I substantiate this further numerically, showing that the reformulation is quite effective and provides significant performance gains over current approaches.

In the third essay, motivated by the operations of the Indian Railways, I consider a novel variant of the classical network revenue management problem. I have a firm that sells multiple products that use multiple resources and adopts a booking limit policy to control the sales of the products. In the traditional booking limit policy, the firm partitions its resource capacities by allocating a fixed amount of capacity to each product or a group of products in order to limit its sales. I consider a setting where, in addition to the partitioned capacities, the firm sets aside some capacity that is common to all the products. The common, pooled capacity is useful as it can capture spill-over demand for the products once their partitioned capacities are exhausted. The firm's decision problem is to determine the optimal partitioned and pooled capacities. I model the above problem as a dynamic program and discuss the conditions under which a simple only-partitioning (where all capacity is dedicated) strategy can be optimal. While an only-partitioning strategy is easy to compute, a hybrid allocation strategy with partitioned and pooled capacities is computationally difficult due to the large state space of the dynamic program. To address this, I first develop a new Lagrangian relaxation-based solution wherein I decompose the network problem by product and resource. I then show that the resultant relaxation-based hybrid allocation strategy can be computed efficiently. I evaluate the solution numerically against well-known upper bounds on a real-world dataset and find that the proposed approach provides a tighter upper bound on the optimal revenue. I also evaluate the solution with respect to revenue and observe that it offers significant improvements over existing heuristic strategy.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Milind Sohoni, for his endless support and encouragement throughout my research journey and beyond. His guidance, feedback, and kind words have been invaluable in shaping my work and helping me grow as a researcher. I am especially grateful for his patience when I made mistakes and for always being there to provide clarity and motivation. I feel incredibly fortunate to have him as my mentor, and I will always be indebted to him for the opportunity.

I would like to extend my sincere thanks to Professor Chaithanya Bandi, who helped me navigate the technical challenges I encountered, especially during the early stages of FPM. His expertise and insights were instrumental in overcoming these hurdles. I am also grateful to Professor Sumit Kunnumkal, whose courses deepened my understanding and appreciation of optimization techniques, which I was able to apply in my research. His guidance has been crucial in shaping my thinking and research approach. Additionally, I would like to thank Professor Vishwakant Malladi for his motivation and for being on my committee.

My heartfelt thanks go to Professor Sripad Devalkar, who introduced me to operations management research and taught me the importance of clarity in approaching problems. I am also thankful to the operations management faculty at large, especially Professor Pallavi Basu, for the courses offered and for their motivation throughout my journey.

I would also like to express my gratitude to my friends, especially Adithya Patil, Himanshu Arha, and Pankaj Jindal, for their feedback, long discussions, and for making my stay at ISB a memorable one. Special thanks go to my friends and batchmates Suresh, Sandhya, Bullipe, and Aindrila for their unwavering support.

Last but not least, I would like to thank my family, without whose support I wouldn't have been able to complete this journey. Thank you!

Contents

Chapter 1

Two-Sided Assortment Optimization

1.1 Introduction

The gig economy has witnessed significant growth over the last decade largely driven by the advent of online platforms that match supply and demand Page-Tickell and Yerby (2020). Examples include delivery platforms (e.g., Uber Eats, Deliveroo), freelance job portals (e.g., Upwork, Fiverr, TaskRabbit), ride sharing apps (e.g., Uber, Lyft) and matchmaking portals (e.g., eHarmony, Tinder, OkCupid). These platforms differ in the amount of autonomy given to the participants. For example, on Uber, the riders and drivers are matched by the platform and have limited freedom in selecting each other. On freelance job portals like Upwork and TaskRabbit, the employers as well as the applicants can exercise their choice. In this paper, we consider the latter setting wherein a platform has to match two groups of participants, and the individuals in each group have a preference over the other. Participants on these platforms are expected to reveal their preferences using specific attributes and the platform then uses the preferences of both sides of the market to provide the best possible recommendation to each participant. However, the final match depends on individual choices.

This setting differs from traditional matching literature like kidney exchange Roth (1984) and organ donation Roth et al. (2004, 2007), where the platform considers the preferences of both sides and provides *stable matches* without considering the uncertainty associated with the individual choice. It also differs from the literature on online retail marketplaces where only one-side of the market, namely the customers, exercise their choice over the products offered by the platform Caro et al. (2014); Golrezaei et al. (2014). However, we borrow the choice models that have been studied within this stream of literature to model individual choices. Specifically, we employ the Multinomial Logit (MNL) function to model participant choice over an offered assortment.

In the two-sided assortment optimization problem considered here, we assume the platform has a set of suppliers and consumers. Each participant has a preference value associated with individuals on the other side of the market and selects from the offered assortment according to the MNL choice model. A successful match generates revenue for the platform and its objective is to compute an optimal assortment for each participant to maximize the expected revenue. Thus, our model extends the MNL based assortment optimization literature to two-sided markets by incorporating the effect of choice decisions by both sides on the final outcome.

The paper is organized as follows. In §3.2, we discuss the related literature and summarize our contributions. In §3.3, we introduce the two-sided assortment optimization model, comment on the computational complexity and provide the mixed-integer linear programming (MILP) formulation. In §1.4, we present two simple greedy heuristics and argue that these can lead to arbitrarily bad solutions. In §1.5, we introduce the two relaxations and provide the corresponding parametric guarantees. In §1.6, we perform numerical analysis and evaluate the relaxations against the benchmark greedy heuristics. Finally, we conclude in §1.7. We provide proofs for the analytical results in 1.8.

1.2 Literature Review

In this section, we discuss the related work from the extant literature that can be broadly classified into two categories– *matching theory* and *assortment optimization*.

1.2.1 Matching theory

The static matching theory that has extensively been used in medical matches Roth (1984), course allocation Sönmez and Ünver (2010), organ donations Roth et al. (2004, 2007) and labor markets Roth (1991) began with the seminal work by Gale and Shapley Gale and Shapley (1962). These standard matching models consider two-sided markets wherein the participants are divided into bipartite set, and each set of participants have preferences over the other. Given the preference, the match maker's goal is to develop a matching mechanism that guarantees "stable matches."

Variants of the above model that incorporate dynamic arrival and departures Ünver (2010), matching failure Dickerson et al. (2013), and dynamic arrivals with threshold waiting time Anderson et al. (2014) have also been studied. Arnosti et al. Arnosti et al. (2014) examine congestion in a two-sided dynamic matching market and show that simple intervention policies that limit the visibility of one side, improve the social welfare of all participants on the platform.

In the above work, while the platform takes the preference of both sides of the market, it does not model the specific individual choice. In our setting, we explicitly model participant choice and limit the role of the platform to providing an assortment recommendation.

1.2.2 Assortment Optimization

The assortment optimization literature can be categorized into four threads. The first thread is the *static one-sided assortment optimization* traditionally used in retail settings, where the platform computes the optimal assortment of products to be offered, given the consumer preferences Talluri and Van Ryzin (2004). In their seminal work, Talluri and Ryzin Talluri and Van Ryzin (2004) showed that under the MNL choice model, the one-sided assortment optimization can be solved efficiently and, in fact, in closed form. Since then, various extensions have been proposed that consider more general choice models. For example, the nested logit model Davis et al. (2014); Li et al. (2015) and the mixture of multinomial logit model Rusmevichientong et al. (2014). Other extensions include incorporating additional constraints like the capacitated MNL Rusmevichientong et al. (2009) and MNL with product costs Kunnumkal and Martínez-de Albéniz (2019). These variants have been shown to be generally computationally difficult, and various algorithms have been developed which provide approximation guarantees.

The second thread is the *dynamic one-sided assortment optimization* which extends the static model to account for dynamic consumer arrivals. Golrezaei et al. Golrezaei et al. (2014) develop an *indexing* algorithm that incorporates the product inventory information to compute the optimal assortment in real-time for multiple customer types and provide approximation guarantees. Chen et al. Chen et al. (2021) study the dynamic setting under the nested choice model.

The third thread is the *static two-sided assortment optimization* which is of primary interest in this paper. Ashlagi et al. Ashlagi et al. (2019) introduced the two-sided assortment optimization problem when the platform's objective is to maximize the number of matches. They consider a sequential setting where the platform offers an assortment of suppliers to the consumers, who then simultaneously and independently select a supplier using the MNL choice function. Each supplier is then offered only the subset of consumers who selected that supplier. However, they restrict the choice function of suppliers to *uniform-MNL*. They show that the problem is NP-hard and provide a constant-factor approximation guarantee. For the above setting, Torrico et al. Torrico et al. (2020) provide an improved approximation guarantee of $\frac{1-e^{-1}}{8}$. Further, they extend the model to revenue maximization objective by defining supplier dependent rewards and consider the special case when one of the sides is *easy-to-match*– when the preference for the platform options for one side is at least as high as the outside option. When the consumers are easy-to-match, they provide a $\frac{1-e^{-1}}{2}$ approximation guarantee, and when the suppliers are easy-to-match, they provide a $\frac{1-e^{-1}}{4}$ approximation. However, under the revenue maximization objective, when both sides follow the general MNL choice function they do not provide any guarantees.

The final thread is the *dynamic two-sided assortment optimization* which extends the static setting to dynamic consumer arrivals. Aouad and Saban Aouad and Saban (2020) consider a platform with a set of suppliers and dynamically arriving consumers. They show that the problem is NP-hard, and develop algorithms that provides an approximation guarantee greater than $1 - \frac{1}{e}$ when the supplier's choice function is MNL or

Nested logit function. However, they consider the case when the platform's objective is to maximize the number of matches.

In this paper, we consider the static two-sided assortment optimization introduced in Ashlagi et al. (2019) and extend it in following ways:

(i) We present a general model for the two-sided assortment optimization wherein the platform's revenue and the participant's preference values depend on supplier-consumer pair and the platform's objective is to maximize the expected revenue. We allow both sides to follow the general MNL choice function and show that the problem is NP-hard even when the number of participants on one side of the market is limited to two.

(ii) We discuss two simple greedy heuristics. The first is the *revenue-ordered heuristic*, which is widely studied within assortment optimization literature and is known to be optimal for the static one-sided assortment optimization under the MNL choice function Talluri and Van Ryzin (2004). We show that it can lead to arbitrarily bad solutions. The second heuristic essentially solves for the optimal assortment for each supplier independently and, we again show this heuristic can lead to arbitrarily bad solutions as well. However, numerically we observe this heuristic provides tighter upper bound compared to the other solutions we present.

(iii) We introduce two relaxations, the *one-sided relaxation* and the *two-sided relaxation*. While the two-sided relaxation is easy to solve and reduces to the bipartite matching problem, we show that the one-sided relaxation is NP-hard. We discuss a continuous relaxation solution and provide parametric approximation guarantees for these solutions in general and then show that, when one of the sides is easy-to-match, the one-sided relaxation provides $\frac{1}{2}$ approximation, and when both sides are easy-to-match, the simple bipartite matching provides $\frac{1}{4}$ approximation guarantee.

1.3 Two-Sided Assortment Optimization Problem

In this section, we introduce the two-sided assortment optimization model, discuss the computational complexity and present the MILP formulation.

1.3.1 Two-sided Assortment Optimization Model

We consider a platform with n suppliers and m consumers. We let $\mathcal{N} = \{1, ..., n\}$ denote the set of suppliers and $\mathcal{M} = \{1, ..., m\}$ denote the set of consumers. Given an assortment, we assume each supplier (consumer) independently selects at most one consumer (supplier). We say a *match* between a supplier i and consumer j is successful if i and j select each other. We further assume, for a successful match between supplier i and consumer j, the platform receives a payoff r_{ij}. The platform's goal is to offer a subset, or an assortment, of consumers (suppliers) to each supplier (consumer) so as to maximize its expected revenue.

Let $\mathcal{S}_i^e \subseteq \mathcal{M}$ be the assortment of consumers offered to supplier i, $\mathcal{S}_j^c \subseteq \mathcal{N}$ be the assortment of suppliers offered to consumer j and $\mathcal{S} = \{\mathcal{S}_1^e, .., \mathcal{S}_n^e, \mathcal{S}_1^c, ..., \mathcal{S}_m^c\}$. $\mathcal{S}$ is said to be *consistent* if consumer j is offered to supplier i, then i is offered to j as well, i.e., $j \in \mathcal{S}_i^e$ iff $i \in \mathcal{S}_j^c$. Let $\mathcal{P}_{ij}(\mathcal{S}_i^e)$ be the probability that supplier i selects consumer j from the assortment $\mathcal{S}_i^e$ and $\mathcal{Q}_{ji}(\mathcal{S}_j^c)$ the probability that consumer j selects supplier i from the assortment $\mathcal{S}_j^c$. Then, the two-sided optimization problem is given by:

$$Z = \max_{\mathcal{S}} \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \cdot \mathcal{P}_{ij}(\mathcal{S}_i^e) \cdot \mathcal{Q}_{ji}(\mathcal{S}_j^c). \tag{1.3.1}$$

Note that $\mathcal{S}$ will be optimal only if it is consistent.

We assume the above choice probabilities follow the MNL choice function. Let v_{ij} denote supplier i's preference for consumer j and v_{i0} denote i's preference for the outside option. Let u_{ji} denote consumer j's preference for supplier i and u_{j0} denote j's preference for the outside option. Under the MNL model, the probabilities are given by

$$\mathcal{P}_{ij}(\mathcal{S}_i^e) = \frac{v_{ij}}{v_{i0} + \sum_{k \in \mathcal{S}_i^e} v_{ik}}; \quad \mathcal{Q}_{ji}(\mathcal{S}_j^c) = \frac{u_{ji}}{u_{j0} + \sum_{l \in \mathcal{S}_j^c} u_{jl}}, \ \forall i \in \mathcal{N}, \ j \in \mathcal{M}. \tag{1.3.2}$$

Since $\mathcal{S}$ is consistent, the above probabilities can be defined using the following binary variable. Let $X_{ij} \in \{0,1\}$ denote whether supplier i and consumer j are in each other's assortment. Let $X = \{X_{ij} : X_{ij} \in \{0,1\}, i \in \mathcal{N}, j \in \mathcal{M}\}$. Given X, the choice probabilities can equivalently be written as:

$$\mathcal{P}_{ij}(X) = \frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}}; \quad \mathcal{Q}_{ji}(X) = \frac{u_{ji} \cdot X_{ij}}{u_{j0} + \sum_{l=1}^{n} u_{jl} \cdot X_{lj}}, \ \forall i \in \mathcal{N}, \ j \in \mathcal{M}. \tag{1.3.3}$$

Let $Z(X)$ denote the expected payoff associated with assortment X, given as follows:

$$Z(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji} \cdot X_{ij}}{u_{j0} + \sum_{l=1}^{n} u_{jl} \cdot X_{lj}} \right). \tag{1.3.4}$$

The optimal two-sided assortment X^* can be obtained by solving the problem:

$$Z = \max_X \{Z(X) : X_{ij} \in \{0, 1\}\}. \quad (1.3.5)$$

We observe that, our model extends the earlier static models by allowing the revenue r_{ij} and the preference values v_{ij}, u_{ji}, to be dependent on supplier-consumer pair. In Ashlagi et al. Ashlagi et al. (2019) the revenue term is uniformly set to 1, as the platform's objective is to maximize the expected number of matches. As discussed in section 3.2, they provide approximation guarantees when the consumer's preference value depends only on the supplier, i.e., $u_{ji} = u_i$, $\forall i \in \mathcal{N}, j \in \mathcal{M}$, and the suppliers follow the *uniform-MNL* choice function, i.e., when $v_{ij} = 1$, $\forall i, j$. Torrico et al. Torrico et al. (2020) consider the revenue maximization objective for the special case when the rewards depend only on the suppler, $r_{ij} = r_i$, $\forall i, j$. Thus our model strictly generalizes the existing static models. However, we highlight that our model differs from the sequential setting considered in Ashlagi et al. (2019); Torrico et al. (2020) in the following way. In Ashlagi et al. (2019); Torrico et al. (2020), the platform computes an assortment for the consumers, and the supplier i's assortment is restricted to be the subset of consumers that select the supplier i. In our model, we allow the platform to simultaneously compute the optimal assortments for both sides. And, since we ensure the assortments are consistent, supplier i's assortment includes the set of consumers to whom supplier i is offered.

The stylized static model considered in this paper is applicable for platforms wherein the participant-platform interaction is *passive* and the platform need not ensure a successful match in real-time. Further, as discussed in Ashlagi et al. Ashlagi et al. (2019), the static model is relevant when there is a delay between participant's response to each other. This would allow the platform to use the existing pool of active participants to compute optimal recommendations offline, as long as their profiles are static. For example, on matchmaking websites like eHarmony.com or job portals like Monster.com, participant's profiles remain the same for reasonable duration of time. However, on platform where participants arrive or depart the system dynamically, or their profiles change frequently, the dynamic models would be more relevant. For example, on dating apps like Tinder, the user's location is used as an additional attribute to provide recommendations Tyson et al. (2016). In such settings, the platform cannot compute the assortments offline.

1.3.2 Computational Complexity

In this subsection, we show that the two-sided optimization is computationally difficult even with two suppliers, identical rewards, and no outside option for consumers. The result follows from the NP-hardness proof for the one-sided relaxation that we discuss in section 1.5.

Theorem 1.3.1. *The two-sided assortment optimization is NP-hard.*

While the general two-sided assortment optimization is computationally difficult even in simple settings, the following result shows that computing the optimal two-sided assortment for a single supplier reduces to the standard MNL optimization problem with an *adjusted reward function* which has an intuitive interpretation.

Observation 1.3.1. *For a single supplier i, the two-sided assortment optimization reduces to the following MNL:*

$$\max_{X_{ij}\in\{0,1\}} \left\{ \sum_{j=1}^{m} \tilde{r}_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \right\},$$

where $\tilde{r}_{ij} = r_{ij} \cdot \frac{u_{ji}}{u_{j0}+u_{ji}}$. Thus, the optimal assortment can be obtained by rank ordering the consumers with respect to $\tilde{r}_{ij}$ Talluri and Van Ryzin (2004). Note that, $\tilde{r}_{ij}$ is the platform's payoff on matching i, j *adjusted* with j's preference for i and its preference for the outside option. As the consumer j's affinity to the outside option increases, i.e., higher u_{j0}, the adjusted platform payoff on matching i, j decreases. Thus, for two consumers with identical rewards, the consumer with the better outside option is less likely to be included in i's assortment.

1.3.3 Exact MILP Reformulation

Since Z is non-linear, we provide an equivalent MILP formulation which can be readily solved using standard optimization software. Recall that the objective is given by:

$$Z = \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji} \cdot X_{ij}}{u_{j0} + \sum_{l=1}^{n} u_{jl} \cdot X_{lj}} \right) \right\}. \quad (1.3.6)$$

Let $X_{i0} = 1, X_{0j} = 1, \mathcal{N}^+ = \mathcal{N} \cup \{0\}$ and $\mathcal{M}^+ = \mathcal{M} \cup \{0\}$. We formulate the equivalent MILP by defining variables ρ_{ij}, $W_{ik,lj}$, $\forall i \in \mathcal{N}$, $\forall j \in \mathcal{M}$ as follows:

$$\rho_{ij} = \frac{1}{(v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}) \cdot (u_{j0} + \sum_{l=1}^{n} u_{jl} \cdot X_{lj})}, \quad W_{ik,lj} = \rho_{ij} \cdot X_{ik} \cdot X_{lj}. \quad (1.3.7)$$

Theorem 1.3.2. *The non-linear two-sided assortment optimization in 1.3.6 is equivalent to the following MILP:*

$$Z^{MILP} = \max_{W,X,\rho} \quad \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \cdot v_{ij} \cdot u_{ji} \cdot W_{ij,ij}, \quad (1.3.8a)$$

s.t.

$$v_{i0} \cdot u_{j0} \cdot \rho_{ij} + \sum_{l=1}^{n} v_{i0} \cdot u_{jl} \cdot W_{i0,lj} + \sum_{k=1}^{m} v_{ik} \cdot u_{j0} \cdot W_{ik,0j} + \sum_{l=1}^{n}\sum_{k=1}^{m} v_{ik} \cdot u_{jl} W_{ik,lj} = 1, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, \tag{1.3.8b}$$

$$W_{ik,lj} \leq X_{ik}, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, l \in \mathcal{N}^+, k \in \mathcal{M}^+, \tag{1.3.8c}$$

$$W_{ik,lj} \leq X_{lj}, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, l \in \mathcal{N}^+, k \in \mathcal{M}^+, \tag{1.3.8d}$$

$$W_{ik,lj} \leq \rho_{ij}, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, l \in \mathcal{N}^+, k \in \mathcal{M}^+, \tag{1.3.8e}$$

$$W_{ik,lj} \geq \rho_{ij} + X_{ik} + X_{lj} - 2, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, l \in \mathcal{N}^+, k \in \mathcal{M}^+, \tag{1.3.8f}$$

$$\rho_{ij} \geq 0, \ X_{ij} \in \{0,1\}, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, \tag{1.3.8g}$$

$$W_{ik,lj} \geq 0, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, l \in \mathcal{N}^+, k \in \mathcal{M}^+. \tag{1.3.8h}$$

The first term in the constraint 2.3.2c $v_{i0} \cdot u_{j0} \cdot \rho_{ij}$, is the probability that both supplier i and consumer j select their outside options, the second term $\sum_{l=1}^{n} v_{i0} \cdot u_{jl} \cdot W_{i0,lj}$, is the probability that i selects the outside option and j selects from the assortment, the third term $\sum_{k=1}^{m} v_{ik} \cdot u_{j0} \cdot W_{ik,0j}$, is the probability that j selects the outside option and i selects from the assortment, and the last term $\sum_{l=1}^{n}\sum_{k=1}^{m} v_{ik} \cdot u_{jl} \cdot W_{ik,lj}$, is the probability that both i and j select from the offered assortment. Constraints 2.3.2d-1.3.8f are the linearization constraints for the bilinear variable $W_{ik,lj}$. While the above MILP formulation is intractable, the linear programming relaxation, denoted by Z^{LP}, provides a benchmark upper bound.

Since it is difficult to compute the optimal solution for the two-sided formulation, we first consider two simple greedy heuristics and comment on their performance. We then discuss the two relaxations that allow us to obtain the upper and lower bound solutions with parametric guarantees.

1.4 Greedy Heuristics

In this section, we present two greedy heuristics and argue that both can lead to arbitrarily bad solutions with respect to the optimal value.

1.4.1 Revenue Ordered Heuristic:

In the assortment optimization literature, a *revenue-ordered* solution is an assortment that includes all products above a threshold revenue. While the standard MNL admits an optimal assortment that belongs to the set of revenue ordered assortments Talluri and Van Ryzin (2004), they are generally not optimal for extensions of the MNL like the mixture of multinomials Feldman and Topaloglu (2015) or the nested-logit choice models Davis et al. (2014).

We define the revenue-ordered heuristic for the two-sided assortment optimization problem as follows. Let $R_d = \{r_{1,1}, ..., r_{m,n}\}_{\neq}$ be the set of distinct reward values. For each $c_k \in R_d$, we define $X_k^r = \{X_{ij} : X_{ij} \in \{0,1\}\}$ to be an assortment such $X_{ij} = 1$ if $r_{ij} \geq c_k$ and 0 otherwise. Let $X^r = \{X_k^r,\ \forall c_k \in R_d\}$ be the set of revenue ordered assortments. We now argue that a revenue ordered heuristic that selects an assortment $X_k^r \in X^r$, such that $Z(X_k^r)$ is maximized, can lead to arbitrarily bad solutions.

Example 1: Consider two suppliers and two consumers. Let $r_{ij} = \frac{1}{2}$ and $v_{i0} = u_{j0} = 0$ for $i, j \in \{1,2\}$. Let $v_{1,1} = v_{2,2} = 1$, $v_{1,2} = v_{2,1} = \frac{\epsilon}{2}$, $u_{1,2} = u_{2,1} = 1$, and $u_{1,1} = u_{2,2} = \frac{\epsilon}{2}$. The two-sided optimal assortment assigns exactly one consumer (supplier) to each supplier (consumer), and since the outside preference is 0, it ensures that each supplier (consumer) selects the offered consumer (supplier) with probability 1, resulting in a total expected revenue of 1. For the revenue-ordered heuristics, since the rewards are identical, both the consumers (suppliers) are offered to both the suppliers (consumers). Supplier 1 selects consumer 1 with probability $\frac{1}{1+\frac{\epsilon}{2}}$. However, consumer 1 selects supplier 1 with probability $\frac{\frac{\epsilon}{2}}{1+\frac{\epsilon}{2}}$. This reduces the match probability of any supplier-consumer pair to $\frac{\frac{\epsilon}{2}}{(1+\frac{\epsilon}{2})^2}$ and the overall platform revenue reduces to $2 \cdot \frac{\frac{\epsilon}{2}}{(1+\frac{\epsilon}{2})^2}$, which converges to 0 as $\epsilon \to 0$.

While the example shows that the revenue-ordered heuristic can lead to arbitrarily bad solutions, it also highlights the observation in Arnosti et al. Arnosti et al. (2014) that in two-sided markets limiting the visibility of one side improves the overall social welfare.

1.4.2 Greedy Separable Heuristic

We now consider a second heuristic, which exploits observation 1.3.1, that computing the optimal assortment for one supplier is tractable. While we show that the heuristic can lead to an arbitrarily bad lower bound, numerically we observe it provides tighter upper bound than the one-sided and the two-sided relaxations that we introduce in the next section. Consider Z as defined in equation 1.3.6. We obtain a separable upper bound as follows:

$$Z = \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \cdot \left(\frac{u_{ji}\cdot X_{ij}}{u_{j0} + \sum_{l=1}^{n} u_{jl}\cdot X_{lj}} \right) \right\}, \tag{1.4.1a}$$

$$\leq \sum_{i=1}^{n} \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \cdot \left(\frac{u_{ji}\cdot X_{ij}}{u_{j0} + \sum_{l=1}^{n} u_{jl}\cdot X_{lj}} \right) \right\}, \tag{1.4.1b}$$

$$= \sum_{i=1}^{n} \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \cdot \left(\frac{u_{ji}}{u_{j0} + u_{ji}} \right) \right\}, \tag{1.4.1c}$$

$$= \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji}}{u_{j0} + u_{ji}} \right) \right\}. \tag{1.4.1d}$$

where, Equality 1.4.1c follows from observation 1.3.1. And, since 1.4.1c is separable across suppliers, 1.4.1d follows. Denote the separable upper bound with Z^g:

$$Z^g = \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji}}{u_{j0} + u_{ji}} \right) \right\}. \tag{1.4.2a}$$

Let X^g be the optimal solution of 1.4.2a. Since, X^g is feasible with respect to the two-sided assortment optimization problem, it provides a lower bound, $Z(X^g) \leq Z$. We now show that X^g, the assortment obtained by independently maximizing the revenue for each supplier, can lead to arbitrarily bad lower bounds.

Example 2: Consider an instance of two-sided assortment optimization with two suppliers and two consumers. We define the rewards as $r_{1,1} = 1$, $r_{1,2} = \omega, r_{2,1} = \frac{1}{3}, r_{2,2} = 1$. Let $v_{i,j} = 1, \forall i \in \mathcal{N}, j \in \mathcal{M}$, $u_{1,1} = u_{1,2} = u_{2,1} = 1$ and $u_{2,2} = \omega$ where $\omega > 2$. Let the outside preference for both consumers and suppliers be 1, $u_{j,0} = v_{i,0} = 1, \forall i \in \mathcal{N}, j \in \mathcal{M}$. The optimal two-sided assortment assigns consumer 1 to supplier 2 and consumer 2 to supplier 1. The resultant revenue is given by $\frac{3\omega+1}{12}$. The separable greedy solution that optimizes independently for each supplier assigns consumer 2 to both the suppliers. The resultant revenue is given by $\frac{1}{1+2/\omega}$. Note that, as ω increases, the revenue–due to the greedy heuristic solution converges to 1, whereas the revenue for the two-sided optimal solution increases with ω. Thus, the greedy separable heuristic can lead to arbitrarily solutions.

1.5 Relaxations and Bounds

In this section, we introduce the one-sided and two-sided relaxation based solutions and present the parametric bounds.

1.5.1 One-Sided Relaxation

We obtain the one-sided relaxation by setting the preference for the outside option of all consumers to zero. Define $\tilde{Z}$ as follows:

$$\tilde{Z} = \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji} \cdot X_{ij}}{\sum_{l=1}^{n} u_{jl} \cdot X_{lj}} \right) \right\}. \tag{1.5.1}$$

Let $\tilde{X}$ be the corresponding optimal one-sided relaxation solution. The following result holds for $\tilde{X}$:

Proposition 1.5.1. *For each consumer* $j \in \mathcal{M}$, $\sum_{i=1}^{n} \tilde{X}_{ij} \leq 1$.

Thus, when the preference for the outside option is zero, at optimality, consumers are recommended to at most one supplier. Proposition 1.5.1 allows us to rewrite the one-sided relaxation as follows:

$$\begin{aligned}
\tilde{Z} &= \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji} \cdot X_{ij}}{\sum_{l=1}^{n} u_{jl} \cdot X_{lj}} \right) \right\}, && \text{(1.5.2a)} \\
&= \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji} \cdot X_{ij}}{\sum_{l=1}^{n} u_{jl} \cdot X_{lj}} \right) \middle| \sum_{i=1}^{n} X_{ij} \leq 1,\ \forall j \in \mathcal{M} \right\}, && \text{(1.5.2b)} \\
&= \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \middle| \sum_{i=1}^{n} X_{ij} \leq 1, \forall j \in \mathcal{M} \right\}, && \text{(1.5.2c)}
\end{aligned}$$

where, the second equality follows from Proposition 1.5.1. And since $\sum_{l=1}^{n} X_{lj} \leq 1$, $\frac{u_{ji} \cdot X_{ij}}{\sum_{l=1}^{n} u_{jl} X_{lj}} = X_{ij}$, $\forall j \in \mathcal{M}$, the third equality follows.

We now show that $\tilde{Z}$ is NP-hard. Similar to Caro et al. (2014), we use the reduction from the PARTITION problem which is known to be NP-complete. We first make the following observation that when the rewards are identical, the one-sided relaxation assigns each consumer to exactly one supplier.

Observation 1.5.1. *If $r_{ij} = c$, $\forall i \in \mathcal{N}, j \in \mathcal{M}$ and $c > 0$, then for each consumer $j \in \mathcal{M}$, $\sum_{i=1}^{n} \tilde{X}_{ij} = 1$.*

Consider the following decision theoretic version of the one-sided relaxation.

<u>ONE-SIDED FEASIBILITY</u>

INPUTS: A set of m consumers indexed by j and two suppliers indexed $1, 2$ with $r_{1j} = r_{2j} = 1$, $\forall j \in \{1, ..., m\}$, the preference weights $v_{1,1}, .., v_{2,m}$, the preference for outside option $v_{1,0}, v_{2,0}$, and a target profit K (we slightly abuse the notation and use $v_{i,j}$ for v_{ij} to avoid ambiguity).

DECISION: Is there a partition S_1, S_2 of consumers such that $S_1 \cap S_2 = \emptyset$ and $S_1 \cup S_2 = \{1, 2, .., m\}$ such that

$$\frac{\sum_{j \in S_1} v_{1,j}}{v_{1,0} + \sum_{j \in S_1} v_{1,j}} + \frac{\sum_{j \in S_2} v_{2,j}}{v_{2,0} + \sum_{j \in S_2} v_{2,j}} \geq K. \tag{1.5.3}$$

Note that $S_1 \cup S_2 = \{1, 2, .., m\}$ follows from observation 1.5.1.

Theorem 1.5.1. *One-sided feasibility problem is NP-complete.*

Theorem 1.5.1 also proves theorem 1.5.3, since the one-sided relaxation is obtained by setting $u_{j0} = 0$, $\forall j \in \mathcal{M}$. While the general one-sided relaxation is computationally intractable, we observe that for the special case when $r_{ij} = 1$ or $r_{ij} = r_i, \forall i \in \mathcal{N}, j \in \mathcal{M}$, the optimization problem given in 2.5.19a, satisfies the submodularity property Han et al. (2020) [1]. Thus, a simple greedy algorithm can be constructed that guarantees $(1 - e^{-1})$-approximate solution Nemhauser et al. (1978). We, therefore, first provide parametric bound for an approximate solution.

Let $\tilde{X}^a$ be a γ-approximate solution to the one-sided relaxation problem in 2.5.19a. Let $u^{min} = \min_{\substack{i \in \mathcal{N} \\ j \in \mathcal{M}}} \{\frac{u_{ji}}{u_{j0}}\}$ and $\alpha = \frac{1}{1+u^{min}}$. Recall that X^* is the optimal solution for the two-sided assortment optimization problem. Then the following result holds:

Proposition 1.5.2. $Z(\tilde{X}^a) \geq \gamma \cdot (1 - \alpha) \cdot Z(X^*)$.

Thus, a γ approximate solution to the one-sided relaxation provides a $\gamma \cdot (1 - \alpha)$ approximation to the two-sided assortment optimization problem. It follows that, for the special case, when $r_{ij} = 1$ or $r_{ij} = r_i, \forall i \in \mathcal{N}, j \in \mathcal{M}$, a greedy solution that provides $1 - e^{-1}$ guarantee for one-sided relaxation problem, provides a $(1 - e^{-1}) \cdot (1 - \alpha)$ guarantee for the two-sided assortment optimization. And, when consumers are easy-to-match, $\alpha = \frac{1}{2}$ and we recover the $\frac{1-e^{-1}}{2}$ guarantee observed in Torrico et al. (2020).

We now consider the continuous relaxation of 2.5.19a and provide parametric bounds in expectation by constructing a solution using *randomized assignment*.

1.5.1.1 Continuous Relaxation

Consider the one-sided relaxation given in equation 2.5.19a.

$$\tilde{Z} = \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \middle| \sum_{i=1}^{n} X_{ij} \leq 1, \forall j \in \mathcal{M} \right\},$$

[1] For the sake of completeness, we include the submodularity proof in 1.8.1

$$= \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0}+v_{ij}+\sum_{\substack{k=1,\\ k\neq j}}^{m} v_{ik}\cdot X_{ik}} \right) \middle| \sum_{i=1}^{n} X_{ij}\leq 1, \forall j\in\mathcal{M} \right\}. \quad (1.5.4a)$$

Let $\tilde{Z}^c$ be the continuous relaxation obtained by replacing the binary constraints in 2.5.19b with $X_{ij} \in [0,1]$. Note that, we take the v_{ij} term out in the denominator in equation 2.5.19b before relaxation as it allows us to obtain the parametric bound. While the continuous relaxation is still computationally difficult, it's a special case of the fractional programming problem for which scalable solutions have been developed Benson (2007); Caro et al. (2014). Let $\tilde{X}^c = \{\tilde{X}^c_{ij}\}$ be the optimal solution to the continuous relaxation. Since $\tilde{X}^c_{ij} \in [0,1]$ and $\sum_i \tilde{X}^c_{ij} \leq 1$, we use *randomized assignment* to obtain an integral solution. Let Q_j be a random variable for consumer j such that $P(Q_j = i) = \tilde{X}^c_{ij}$. Q_j assigns each consumer j to supplier i with probability $\tilde{X}^c_{ij}$. Define the resultant assortment obtained using randomized assignment as $\tilde{X}^b = \{\tilde{X}^b_{ij} : \tilde{X}^b_{ij} = \mathbb{1}_{Q_j=i}\}$. Since $\tilde{X}^b_{ij}$ is binary, it is feasible with respect to Z. Let $\mathbb{E}_Q[Z(\tilde{X}^b)]$ be the expected value of the solution $\tilde{X}^c$, where the expectation is over the random variable $Q = \{Q_j : j \in \mathcal{M}\}$. Let α be defined as before, then following result holds:

Theorem 1.5.2. *$(1-\alpha)Z(X^*) \leq \mathbb{E}_Q[Z(\tilde{X}^b)] \leq Z(X^*)$.*

For the special case when consumers are easy-to-match, the randomized solution provides a $\frac{1}{2}$ approximation in expectation. We observe that, for both $\tilde{X}^a$ and $\tilde{X}^c$ solutions, as the preference parameter of the consumers for the platform increases (compared to the outside option), i.e., u_{ji} increases, α decreases and the performance guarantee increases.

1.5.2 Two-Sided Relaxation

We obtain the two-sided relaxation by setting the preference for the outside option for the suppliers and the consumers to zero. Define $\tilde{\tilde{Z}}$ as follows:

$$\tilde{\tilde{Z}} = \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{\sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji} \cdot X_{ij}}{\sum_{l=1}^{n} u_{jl} \cdot X_{lj}} \right) \right\}. \tag{1.5.5}$$

Let $\tilde{\tilde{X}}$ be the corresponding optimal two-sided assortment. Similar to one-sided relaxation, the following result holds for $\tilde{\tilde{X}}$

Proposition 1.5.3. *For each supplier $i \in \mathcal{N}$, $\sum_{j=1}^{m} \tilde{\tilde{X}}_{ij} \leq 1$ and for each consumer $j \in \mathcal{M}$, $\sum_{i=1}^{n} \tilde{\tilde{X}}_{ij} \leq 1$.*

Thus, when the preference for the outside option is zero for both sides of the market, each consumer (supplier) is recommended to at most one supplier (consumer). We use the above proposition to rewrite the two-sided assortment optimization as follows:

$$\tilde{\tilde{Z}} = \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{\sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji} \cdot X_{ij}}{\sum_{l=1}^{n} u_{jl} \cdot X_{lj}} \right) \right\}, \tag{1.5.6a}$$

$$= \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{\sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji} \cdot X_{ij}}{\sum_{l=1}^{n} u_{jl} \cdot X_{lj}} \right) \middle| \sum_{j=1}^{m} X_{ij} \leq 1,\ \sum_{i=1}^{n} X_{ij} \leq 1, \forall i \in \mathcal{N},\ \forall j \in \mathcal{M} \right\}, \tag{1.5.6b}$$

$$= \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \cdot X_{ij} \middle| \sum_{j=1}^{m} X_{ij} \leq 1,\ \sum_{i=1}^{n} X_{ij} \leq 1,\ \forall i \in \mathcal{N},\ \forall j \in \mathcal{M} \right\}, \tag{1.5.6c}$$

where, 1.5.6b follows from proposition 1.5.3. Therefore, the two-sided relaxation reduces to the standard bipartite matching problem. We now provide the parametric bound for the matching solution. Let $v^{min} = \min_{\substack{i \in \mathcal{N} \\ j \in \mathcal{M}}} \{\frac{v_{ij}}{v_{i0}}\}$, $\beta = \frac{1}{1+v^{min}}$, and α be defined as before. Then, we obtain:

Theorem 1.5.3. $(1-\alpha)(1-\beta)Z(X^*) \leq Z(\tilde{\tilde{X}}) \leq Z(X^*)$.

Similar to the one-sided approximation, the above result implies that, when both the sides are easy-to-match, a simple bipartite matching provides a $\frac{1}{4}$ approximation to the two-sided optimization problem. We note that while the matching solution provides a lower approximation guarantee, it is computationally faster.

1.6 Numerical Analysis

In this section, we evaluate the solutions under various parameter settings [2]. For the upper bounds, we report the linear programming relaxation Z^{LP}, the greedy separable heuristic Z^g, the one-sided continuous relaxation $\tilde{Z}^c$, and the two-sided relaxation $\tilde{\tilde{Z}}$. We report the lower bounds obtained using the revenue-ordered heuristic $Z(X^r)$, the greedy separable heuristic solution $Z(X^g)$, the randomized one-sided relaxation solution $\mathbb{E}_Q[Z(\tilde{X}^b)]$, and the two-sided relaxation solution $Z(\tilde{\tilde{X}})$. We compute $\mathbb{E}_Q[Z(\tilde{X}^b)]$ by simulating the continuous relaxation solution $\tilde{X}^c$, 10,000 times and report the average value.

Across all simulations, we set the outside option to 10 for all suppliers and consumers, and sample their platform affinities v_{ij}, u_{ji}, from a uniform distribution $[0, V]$, where V is a parameter of interest. We sample the reward values from a uniform distribution

[2] All the optimization solutions, except $\tilde{X}^c$, were obtained using CPLEX V12.9.0. For $\tilde{X}^c$, we used Gekko Beal et al. (2018). The simulations were run on Intel core i7, 16 GB RAM, Ubuntu 18.04.5 LTS using Python 3.6.9

Problem (δ, V)	$Z(X^{LP})$ UB	$Z(X^{LP})$ η_u	Z^g UB	Z^g η_u	$\tilde{Z}$ UB	$\tilde{Z}$ η_u	$\tilde{\tilde{Z}}$ UB	$\tilde{\tilde{Z}}$ η_u
(1.0, 10)	226.01±18.77	10.27±1.29	40.54±2.52	1.84±0.22	69.01±4.26	3.14±0.41	180.39±4.89	8.24±1.19
(1.0, 50)	906.31±43.96	10.89±1.06	108.52±4.8	1.3±0.11	127.06±6.81	1.53±0.15	179.08±4.74	2.15±0.2
(1.0, 100)	1096.93±43.72	9.84±0.73	134.6±5.08	1.21±0.08	144.21±6.37	1.29±0.08	179.54±5.0	1.61±0.12
(2.0, 10)	443.33±27.1	14.61±1.47	49.17±2.1	1.62±0.14	101.66±3.6	3.36±0.38	191.85±2.41	6.34±0.72
(2.0, 50)	1804.72±66.1	18.97±1.64	120.21±3.45	1.26±0.09	149.78±4.31	1.57±0.13	191.37±2.07	2.01±0.17
(2.0, 100)	2191.48±59.23	17.47±1.01	145.77±3.15	1.16±0.06	162.91±4.68	1.3±0.06	191.96±1.97	1.53±0.08
(3.0, 10)	663.79±29.76	18.13±1.44	54.38±1.47	1.49±0.11	119.05±2.6	3.25±0.27	194.98±1.44	5.33±0.44
(3.0, 50)	2735.88±68.03	26.49±1.63	127.25±2.94	1.23±0.07	161.73±3.38	1.57±0.1	194.78±1.62	1.89±0.12
(3.0, 100)	3302.62±72.5	25.61±1.34	150.91±2.51	1.17±0.06	170.18±3.19	1.32±0.06	194.89±1.57	1.51±0.07

Table 1.1: Upper bounds

over $[5, 20]$ and vary the consumer-to-supplier ratio, δ. Thus, each problem instance is characterized by the tuple (δ, V).

For each instance, we report the performance by taking the average across 100 simulations. For upper bounds, we report the average upper bound, denoted by UB, and the average ratio of the upper bound to the maximum lower bound, denoted by $\eta_u(> 1)$. For the lower bounds, we report the average lower bound, denoted by LB, and the average ratio of lower bound to the minimum upper bound denoted by $\eta_l(< 1)$.

We run the simulations by setting the number of suppliers to 10 and considering $\delta \in \{1, 2, 3\}$ and $V \in \{10, 50, 100\}$. The configuration $(\delta = 2, V = 10)$ implies, the number of suppliers $n = 10$, consumers $m = 20$, and the preference for platform options, v_{ij}, u_{ji}, for both suppliers and consumers, is less than the preference for the outside option with probability 1.

Problem (δ, V)	$Z(X^r)$ LB	$Z(X^r)$ η_l	$Z(X^g)$ LB	$Z(X^g)$ η_l	$\mathbb{E}_Q[Z(\tilde{X}^b)]$ LB	$\mathbb{E}_Q[Z(\tilde{X}^b)]$ η_l	$Z(\tilde{\tilde{X}})$ LB	$Z(\tilde{\tilde{X}})$ η_l
(1.0, 10)	18.69±2.2	0.46±0.04	16.2±1.59	0.4±0.03	21.91±3.81	0.54±0.08	16.88±4.1	0.42±0.09
(1.0, 50)	52.73±7.19	0.49±0.07	43.81±5.36	0.4±0.05	82.6±8.95	0.76±0.07	75.16±9.88	0.69±0.09
(1.0, 100)	67.92±9.36	0.5±0.06	59.47±9.03	0.44±0.06	109.94±9.25	0.82±0.06	105.29±9.45	0.78±0.06
(2.0, 10)	25.52±2.29	0.52±0.04	24.11±1.66	0.49±0.03	30.56±3.41	0.62±0.06	17.9±4.47	0.36±0.09
(2.0, 50)	68.2±8.21	0.57±0.06	62.83±7.02	0.52±0.05	95.64±7.79	0.8±0.06	77.11±11.85	0.64±0.09
(2.0, 100)	88.25±11.15	0.6±0.07	84.43±10.37	0.58±0.07	124.82±7.83	0.86±0.05	111.53±12.38	0.76±0.08
(3.0, 10)	29.94±2.58	0.55±0.04	30.09±1.8	0.55±0.03	36.8±3.03	0.68±0.05	18.06±4.03	0.33±0.07
(3.0, 50)	79.0±8.26	0.62±0.06	74.67±7.1	0.59±0.05	103.62±6.84	0.81±0.05	78.16±10.5	0.61±0.08
(3.0, 100)	98.28±10.73	0.65±0.07	94.19±10.09	0.62±0.06	128.62±6.68	0.85±0.04	113.55±11.78	0.75±0.07

TABLE 1.2: Lower bounds

Table 1.1 summarizes the upper bounds. Across all problem instances, the greedy separable upper bound Z^g provides the tightest upper bound and is within a factor of 2 of the maximum lower bound. It significantly outperforms the linear programming relaxation Z^{LP} as well as the one-sided and two-sided relaxation. Additionally, as δ or V increases, the upper bound becomes tighter.

With respect to the lower bounds (table 1.2), the one-sided randomized solution $\mathbb{E}_Q[Z(\tilde{X}^b)]$, outperforms the two-sided relaxation, the greedy separable solution, as well as the revenue-ordered heuristic across all problem instances. We observe that, on an average, it is well within a factor of 0.5 of the minimum upper bound and that these bounds are tighter than the theoretical guarantees [3]. We also observe that as V increases, the performance of the one-sided as well as the two-sided solution increases. This follows from the theoretical results that as the preference for platform options increases (compared to the

[3] In 1.8.2, for the particular case when $u_{j0} = \kappa$, and $u_{ji} \sim \mathcal{U}[\kappa_l, \kappa_u]$, we show the one-sided relaxation provides $1 - \frac{\kappa}{\kappa_u - \kappa_l} \ln(\frac{\kappa_u + \kappa}{\kappa_l + \kappa})$ guarantee in expectation. This implies, for $u_{j0} = 10$ and $u_{ji} \sim \mathcal{U}[0, V]$, we obtain theoretical guarantees of 0.30, 0.64, and 0.76 for $V = 10$, 50, and 100, respectively.

outside option), the performance guarantees of the two relaxations increases. Further, the lower bound for the randomized solution increases with the consumer-to-supplier, and for $\delta = 3$, on an average, it is within a factor of 0.69 of the minimum upper bound. Finally, we observe that, on an average, the revenue-ordered heuristic outperforms the greedy separable heuristic.

1.7 Conclusion

In this paper, we presented a general two-sided assortment optimization model that extends the existing literature by allowing the platform's revenue and the participant's preference to depend on supplier-consumer pair. The NP-hardness result, even with two suppliers, along with the negative results for the two benchmark greedy heuristics, highlight the difficulty of the two-sided assortment optimization problem.

We developed two simple relaxation based solutions. The one-sided relaxation sets the outside option to zero for consumers and the two-sided relaxation sets it to zero for both consumers and suppliers. Our parametric guarantees help highlight the performance as well as the limitations of simple matching solutions.

This work can be explored further in the multiple ways. The first is to develop bounds that scale well with the number of participants. While we numerically observed the performance of the one-sided relaxation increases with increase in the consumer-to-supplier ratio, it does not immediately follow from the current theoretical results. The second is to extend the static model to incorporate cardinality constraints and analyze special cases under which the parametric guarantees still hold e.g., limiting the assortment size to one for either or both sides of the market. Finally, given the relaxations provide upper and lower bound guarantees, it would be worthwhile to explore their efficacy within a branch-and-bound regime to compute optimal two-sided assortments.

1.8 Appendix

Observation 1.3.1: For a single supplier i, the two-sided assortment optimization reduces to the following MNL:

$$\max_{X_{ij}\in\{0,1\}} \left\{ \sum_{j=1}^{m} \tilde{r}_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \right\},$$

Proof: Consider the two-sided assortment optimization Z defined in 1.3.6 for a given supplier i:

$$Z = \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \cdot \left(\frac{u_{ji}\cdot X_{ij}}{u_{j0}+\sum_{l=1}^{n} u_{jl}\cdot X_{lj}} \right) \right\}, \tag{1.8.1a}$$

$$= \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \cdot \left(\frac{u_{ji}\cdot X_{ij}}{u_{j0}+u_{ji}+\sum_{\substack{l=1,\\ l\neq i}}^{n} u_{jl}\cdot X_{lj}} \right) \right\}, \tag{1.8.1b}$$

$$= \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \cdot \left(\frac{u_{ji}\cdot X_{ij}}{u_{j0}+u_{ji}} \right) \right\}, \tag{1.8.1c}$$

$$= \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{j=1}^{m} \tilde{r}_{ij} \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \right\}, \tag{1.8.1d}$$

where 1.8.1c follows from the observation that when maximizing with respect to supplier i, $X_{lj}=0, \forall l\neq i$ $\square$.

Theorem 1.3.1: The two-sided assortment optimization is NP-hard.

Proof: The proof follows from the observation that, a specific instance of a two-sided optimization problem with preference parameter $u_{j0} = 0, \forall j \ in \mathcal{M}$, is equivalent to a general instance of the one-sided relaxation. And it follows from theorem 1.5.1 that the one-sided relaxation is NP-hard $\square$.

Theorem 1.3.2: The non-linear two-sided assortment optimization in 1.3.6 is equivalent to the MILP 2.3.2a-2.3.2g.

Proof: Let $\hat{X}, \hat{W}, \hat{\rho}$ be any feasible for the MILP. It suffices to show that, for $\hat{X}_{ij} \in \{0, 1\}$, W and ρ satisfy 1.3.7. Consider the MILP given in 2.3.2a-2.3.2g. Since $\hat{X}_{ij}$ is binary, the linearization constraints 2.3.2d-1.3.8f ensure the following:

$$\hat{W}_{ik,lj} = \hat{\rho}_{ij} \cdot \hat{X}_{ik} \cdot \hat{X}_{lj}, \tag{1.8.2a}$$

$$\hat{W}_{i0,lj} = \hat{\rho}_{ij} \cdot \hat{X}_{lj}, \tag{1.8.2b}$$

$$\hat{W}_{ik,0j} = \hat{\rho}_{ij} \cdot \hat{X}_{ik}. \tag{1.8.2c}$$

Further, $\hat{\rho}$, $\hat{W}$ satisfy 2.3.2c and we obtain:

$$\begin{aligned} v_{i0} \cdot u_{j0} \cdot \hat{\rho}_{ij} + \sum_{l=1}^{n} v_{i0} \cdot u_{jl} \cdot \hat{\rho}_{ij} \cdot \hat{X}_{lj} + \sum_{k=1}^{m} v_{ik} \cdot u_{j0} \cdot \hat{\rho}_{ij} \cdot \hat{X}_{ik} \\ + \sum_{l=1}^{n} \sum_{k=1}^{m} v_{ik} \cdot u_{jl} \cdot \hat{\rho}_{ij} \cdot \hat{X}_{ik} \cdot \hat{X}_{lj} = 1, \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, \end{aligned} \tag{1.8.3a}$$

which ensures:

$$\hat{\rho}_{ij} = \frac{1}{(v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot \hat{X}_{ik}) \cdot (u_{j0} + \sum_{l=1}^{n} u_{jl} \cdot \hat{X}_{lj})}. \tag{1.8.4a}$$

$$\therefore \hat{W}_{ij,ij} = \frac{\hat{X}_{ij} \cdot \hat{X}_{ij}}{(v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot \hat{X}_{ik}) \cdot (u_{j0} + \sum_{l=1}^{n} u_{jl} \cdot \hat{X}_{lj})}, \tag{1.8.4b}$$

(1.8.4c)

The MILP objective then reduces to the non-linear two-sided optimization in 1.3.6:

$$Z^{MILP} = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{r_{ij}\cdot v_{ij}\cdot u_{ji}\cdot \hat{X}_{ij}\hat{X}_{ij}}{(v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot \hat{X}_{ik})\cdot(u_{j0}+\sum_{l=1}^{n} u_{jl}\cdot \hat{X}_{lj})}, \tag{1.8.5a}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\cdot \hat{X}_{ij}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot \hat{X}_{ik}}\right)\cdot\left(\frac{u_{ji}\cdot \hat{X}_{ij}}{u_{j0}+\sum_{l=1}^{n} u_{jl}\cdot \hat{X}_{lj}}\right)\square. \tag{1.8.5b}$$

Proposition 1.5.1: For each consumer $j \in \mathcal{M}$, $\sum_{i=1}^{n} \tilde{X}_{ij} \leq 1$.

Proof: Consider $\tilde{Z}$ defined in equation 1.5.1. Let the objective function value at X be $\tilde{Z}(X)$, given as follows:

$$\tilde{Z}(X) = \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\cdot X_{ij}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot X_{ik}}\right)\cdot\left(\frac{u_{ji}\cdot X_{ij}}{\sum_{l=1}^{n} u_{jl}\cdot X_{lj}}\right). \tag{1.8.6}$$

Let $\tilde{X}$ be the optimal solution of 1.5.1 i.e., $\tilde{X} = \text{argmax}_X \tilde{Z}(X)$.

Assume there exists j_1, such that $\sum_{i=1}^{n} \tilde{X}_{ij_1} > 1$. We prove by constructing a solution $\tilde{X}^f$ such that $\tilde{Z}(\tilde{X}^f) \geq \tilde{Z}$ and $\sum_{i=1}^{n} \tilde{X}^f_{ij_1} = 1$. Let $\tilde{\mathcal{S}}^c_{j_1} \in \mathcal{N}$ be the set of suppliers such that $\tilde{X}_{ij_1} = 1, \forall i \in \tilde{\mathcal{S}}^c_{j_1}$. Let i_1 be the index of a supplier such that:

$$i_1 = \underset{i\in\tilde{\mathcal{S}}^c_{j_1}}{\text{argmax}} \left\{ r_{ij_1}\left(\frac{v_{ij_1}}{v_{i0}+v_{ij_1}+\sum_{\substack{k=1,\\ k\neq j_1}}^{m} v_{ik}\tilde{X}_{ik}}\right)\right\}, \tag{1.8.7}$$

We construct the solution $\tilde{X}^f$ as follows:

$$\tilde{X}^f_{ij} = \begin{cases} \tilde{X}_{ij} & \text{if } j \neq j_1 \text{ and } i \neq i_1, \\ 0 & \text{if } j = j_1 \text{ and } i \neq i_1, \\ 1 & \text{if } j = j_1 \text{ and } i = i_1. \end{cases} \tag{1.8.8}$$

Thus, $\tilde{X}^f$ is identical to $\tilde{X}$ except for consumer j_1. For j_1, $\tilde{X}^f$ assigns value 1 to supplier i_1 satisfying equation 1.8.7. We now show that $\tilde{Z}(\tilde{X}^f) \geq \tilde{Z}$ and by definition of $\tilde{X}^f$, $\sum_i \tilde{X}^f_{ij_1} = 1$. Rewrite $\tilde{Z}(\tilde{X})$ as follows:

$$\tilde{Z}(\tilde{X}) = \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\tilde{X}_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik}\tilde{X}_{ik}}\right)\left(\frac{u_{ji}\tilde{X}_{ij}}{\sum_{l=1}^{n} u_{jl}\tilde{X}_{lj}}\right), \tag{1.8.9a}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}}{v_{i0} + v_{ij_1}\tilde{X}_{ij_1} + \sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}_{ik}}\right)\left(\frac{u_{ji}\tilde{X}_{ij}}{\sum_{l=1}^{n} u_{jl}\tilde{X}_{lj}}\right), \tag{1.8.9b}$$

$$= \sum_{j=1}^{m}\sum_{i=1}^{n} r_{ij}\left(\frac{v_{ij}}{v_{i0} + v_{ij_1}\tilde{X}_{ij_1} + \sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}_{ik}}\right)\left(\frac{u_{ji}\tilde{X}_{ij}}{\sum_{l=1}^{n} u_{jl}\tilde{X}_{lj}}\right), \tag{1.8.9c}$$

$$= \begin{matrix} \sum\limits_{\substack{j=1,\\ j\neq j_1}}^{m}\sum\limits_{i=1}^{n} r_{ij}\left(\frac{v_{ij}}{v_{i0} + v_{ij_1}\tilde{X}_{ij_1} + \sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}_{ik}}\right)\left(\frac{u_{ji}\tilde{X}_{ij}}{\sum_{l=1}^{n} u_{jl}\tilde{X}_{lj}}\right) \\ + \sum\limits_{i=1}^{n} r_{ij_1}\left(\frac{v_{ij_1}}{v_{i0} + v_{ij_1}\tilde{X}_{ij_1} + \sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}_{ik}}\right)\left(\frac{u_{j_1 i}\tilde{X}_{ij_1}}{\sum_{l=1}^{n} u_{j_1 l}\tilde{X}_{lj_1}}\right) \end{matrix}, \tag{1.8.9d}$$

$$= \begin{matrix} \sum\limits_{\substack{j=1,\\ j\neq j_1}}^{m}\sum\limits_{i=1}^{n} r_{ij}\left(\frac{v_{ij}}{v_{i0} + v_{ij_1}\tilde{X}_{ij_1} + \sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}_{ik}}\right)\left(\frac{u_{ji}\tilde{X}_{ij}}{\sum_{l=1}^{n} u_{jl}\tilde{X}_{lj}}\right) \\ + \sum\limits_{i=1}^{n} r_{ij_1}\left(\frac{v_{ij_1}}{v_{i0} + v_{ij_1} + \sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}_{ik}}\right)\left(\frac{u_{j_1 i}\tilde{X}_{ij_1}}{\sum_{l=1}^{n} u_{j_1 l}\tilde{X}_{lj_1}}\right) \end{matrix}, \tag{1.8.9e}$$

$$= \begin{array}{l} \displaystyle\sum_{\substack{j=1,\\ j\neq j_1}}^{m}\sum_{i=1}^{n} r_{ij}\left(\frac{v_{ij}}{v_{i0}+v_{ij_1}\tilde{X}_{ij_1}+\sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}^f_{ik}}\right)\left(\frac{u_{ji}\tilde{X}^f_{ij}}{\sum_{l=1}^{n} u_{jl}\tilde{X}^f_{lj}}\right) \\ \qquad + \displaystyle\sum_{i=1}^{n} r_{ij_1}\left(\frac{v_{ij_1}}{v_{i0}+v_{ij_1}+\sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}_{ik}}\right)\left(\frac{u_{j_1 i}\tilde{X}_{ij_1}}{\sum_{l=1}^{n} u_{j_1 l}\tilde{X}_{lj_1}}\right), \end{array} \tag{1.8.9f}$$

$$\leq \begin{array}{l} \displaystyle\sum_{\substack{j=1,\\ j\neq j_1}}^{m}\sum_{i=1}^{n} r_{ij}\left(\frac{v_{ij}}{v_{i0}+v_{ij_1}\tilde{X}^f_{ij_1}+\sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}^f_{ik}}\right)\left(\frac{u_{ji}\tilde{X}^f_{ij}}{\sum_{l=1}^{n} u_{jl}\tilde{X}^f_{lj}}\right) \\ \qquad + \underbrace{\displaystyle\sum_{i=1}^{n} r_{ij_1}\left(\frac{v_{ij_1}}{v_{i0}+v_{ij_1}+\sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}_{ik}}\right)\left(\frac{u_{j_1 i}\tilde{X}_{ij_1}}{\sum_{l=1}^{n} u_{j_1 l}\tilde{X}_{lj_1}}\right)}_{\text{term-2}}, \end{array} \tag{1.8.9g}$$

$$\leq \begin{array}{l} \displaystyle\sum_{\substack{j=1,\\ j\neq j_1}}^{m}\sum_{i=1}^{n} r_{ij}\left(\frac{v_{ij}}{v_{i0}+v_{ij_1}\tilde{X}^f_{ij_1}+\sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}^f_{ik}}\right)\left(\frac{u_{ji}\tilde{X}^f_{ij}}{\sum_{l=1}^{n} u_{jl}\tilde{X}^f_{lj}}\right) \\ \qquad + r_{i_1 j_1}\left(\frac{v_{i_1 j_1}\cdot\tilde{X}^f_{i_1 j_1}}{v_{i0}+v_{i_1 j_1}+\sum\limits_{k=1,k\neq j_1}^{m} v_{i_1 k}\tilde{X}^f_{i_1 k}}\right), \end{array} \tag{1.8.9h}$$

$$= \tilde{Z}(\tilde{X}^f), \tag{1.8.9i}$$

where, equality 1.8.9b follows from the observation that $\tilde{X}_{ij}\cdot\tilde{X}_{ij} = \tilde{X}_{ij}$, 1.8.9c follows from linearity of summation, 1.8.9e follows from the fact that $\tilde{X}$ is binary, and the equality 1.8.9f follows from the definition of $\tilde{X}^f$. The first inequality 1.8.9g follows from the observation that $\tilde{X}^f_{ij_1} \leq \tilde{X}_{ij_1},\ \forall i \in \tilde{\mathcal{S}}^c_{j_1}$, the second inequality 1.8.9h follows from the observation that term-2 in equation 1.8.9g is a convex combination of $\left\{ r_{ij_1}\left(\frac{v_{ij_1}}{v_{i0}+v_{ij_1}+\sum\limits_{k=1,k\neq j_1}^{m} v_{ik}\tilde{X}_{ik}}\right), i \in \tilde{\mathcal{S}}^c_{j_1}\right\}$ and from the definition of i_1 $\square$.

Observation 1.5.1 If $r_{ij} = c,\ \forall i \in \mathcal{N}, j \in \mathcal{M}$ and $c > 0$, then for each consumer j, $\sum_{i=1}^{n}\tilde{X}_{ij} = 1$.

Proof: Consider $\tilde{Z}$ defined in equation 2.5.19a for reward $r_{ij} = c$:

$$\tilde{Z} = c \cdot \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \middle| \sum_{i=1}^{n} X_{ij} \leq 1, \forall j \in \mathcal{M} \right\}, \tag{1.8.10a}$$

$$= c \cdot \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n} \left(\frac{\sum_{j=1}^{m} v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \middle| \sum_{i=1}^{n} X_{ij} \leq 1, \forall j \in \mathcal{M} \right\}, \tag{1.8.10b}$$

Let $\tilde{X}$ be the optimal solution of 1.8.10b. Assume there exists consumer j_1 such that $\sum_{i=1}^{n} \tilde{X}_{ij_1} = 0$. For each supplier i, let $y_i = \sum_{j=1}^{m} v_{ij} \cdot \tilde{X}_{ij}$. Since, $\frac{y_i}{v_{i0}+y_i}$ is increasing in y_i, assigning j_1 to any of the suppliers, say i_1, such that $v_{i_1 j_1} > 0$, would increase the objective, thereby contradicting the claim that $\tilde{X}$ is the optimal solution $\square$.

Theorem 1.5.1: The one-sided feasibility problem is NP-complete.

Proof: Similar to Caro et al. (2014), we use reduction from the partition problem which is known to be NP-complete. Consider the following general instance a partition problem:

INPUTS: A set of m products indexed by j; the weight associated with each product $c_1, .., c_m$ DECISION Is there a subset S of products such that $\sum_{j \in S} c_j = \sum_{j \notin S} c_j$

Let $C = \frac{1}{2} \sum_j c_j$. It follows that if there exists a subset S, then $\sum_{j \in S} c_j = \sum_{j \notin S} c_j = C$. Without loss of generality we assume $C \in \mathbb{Z}_+$. Recall that, one-sided feasibility problem is defined as follows:

INPUTS: A set of m consumers indexed by j and two suppliers indexed $1, 2$ with $r_{1j} = r_{2j} = 1$, $\forall j \in \{1, ..., m\}$, the preference weights $v_{1,1}, .., v_{2,m}$, the preference for outside option $v_{1,0}, v_{2,0}$, and a target profit K (we slightly abuse the notation and use $v_{i,j}, u_{j,i}$ for v_{ij}, u_{ji}, respectively to avoid ambiguity).

DECISION Is there a partition S_1, S_2 of consumers such that $S_1 \cap S_2 = \emptyset$ and $S_1 \cup S_2 =$

$\{1, 2, .., m\}$ such that

$$\frac{\sum_{j \in S_1} v_{1,j}}{v_{1,0} + \sum_{j \in S_1} v_{1,j}} + \frac{\sum_{j \in S_2} v_{2,j}}{v_{2,0} + \sum_{j \in S_2} v_{2,j}} \geq K. \tag{1.8.11}$$

We now define a specific instance of the above one-sided feasibility problem. Let $v_{1,j} = v_{2,j} = c_j$, $v_{1,0} = v_{2,0} = 1$ and let $K = \frac{2C}{1+C}$. The remaining proof follows from Caro et al. (2014) which we include for the sake of completeness. First, if there exists a partition of products such that $\sum_{j \in S} c_j = \sum_{j \notin S} c_j = C$, then the inequality 1.8.11 is satisfied as equality. We now show that if there exists an assortment whose revenue is $K = \frac{2C}{1+C}$, then there exists a product partition as well. Let $\sum_{j \in S_1} c_j = y$. Since the maximum value of y is $2C$, the left hand side of inequality 1.8.11 is bounded by the maximum of $\frac{y}{1+y} + \frac{2C-y}{1+2C-y}$, which is concave over $[0, 2C]$ and achieves its maximum at $y = C$, thus completing the proof $\square$.

Proposition 1.5.2: $Z(\tilde{X}^a) \geq (1-\alpha) \cdot \gamma \cdot Z(X^*)$.

Proof: Consider the one-sided relaxation in 2.5.19a, let $\tilde{X}$ be the optimal one-sided relaxation solution. Let $\tilde{Z}(X)$ be the objective value at X, given as follows:

$$\tilde{Z}(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right), \tag{1.8.12}$$

then, the solution $\tilde{X}^a$ is γ-approximate if $\tilde{Z}(\tilde{X}^a) \geq \gamma \cdot \tilde{Z}(\tilde{X})$.

$$\begin{aligned} Z(\tilde{X}^a) &= \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \tilde{X}^a_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \tilde{X}^a_{ij}} \right) \cdot \left(\frac{u_{ji} \tilde{X}^a_{ij}}{u_{j0} + \sum_{l=1}^{n} u_{jl} \tilde{X}^a_{lj}} \right), && \text{(1.8.13a)} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \left(\frac{v_{ij} \cdot \tilde{X}^a_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \tilde{X}^a_{ik}} \right) \cdot \left(\frac{u_{ji} \tilde{X}^a_{ij}}{u_{j0} + u_{ji}} \right), && \text{(1.8.13b)} \end{aligned}$$

$$\geq \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\cdot\tilde{X}^a_{ij}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\tilde{X}^a_{ik}}\right)\cdot\left(\frac{u^{min}}{1+u^{min}}\right), \tag{1.8.13c}$$

$$= \tilde{Z}(\tilde{X}^a)\cdot(1-\alpha), \tag{1.8.13d}$$

$$\geq (1-\alpha)\cdot\gamma\cdot\tilde{Z}(\tilde{X}), \tag{1.8.13e}$$

$$\geq (1-\alpha)\cdot\gamma\cdot Z(X^*), \tag{1.8.13f}$$

where, the second equality 1.8.13b, follows from the observation that $\sum_{l=1}^{n}\tilde{X}^a_{lj}\leq 1$ $\square$.

Theorem: 1.5.2: $(1-\alpha)Z(X^*)\leq \mathbb{E}_Q[Z(\tilde{X}^b)]\leq Z(X^*)$.

Proof: Since $\tilde{X}^b$ is feasible w.r.t Z, we have $\mathbb{E}_Q[Z(\tilde{X}^b)]\leq Z(X^*)\leq\tilde{Z}^c$ where, we recall that $Q=\{Q_j\}$ is the random variable that assigns consumer j to supplier i with probability $\tilde{X}^c$. Consider the $Z(X)$ defined in equation 1.5.5. We obtain:

$$\mathbb{E}_Q[Z(\tilde{X}^b)] = \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\,\mathbb{E}_Q\left[\left(\frac{v_{ij}\tilde{X}^b_{ij}}{v_{i0}+\sum\limits_{k=1}^{m} v_{ik}\tilde{X}^b_{ik}}\right)\cdot\left(\frac{u_{ji}\tilde{X}^b_{ij}}{u_{j0}+\sum_{l=1}^{n} u_{jl}\tilde{X}^b_{lj}}\right)\right], \tag{1.8.14a}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\,\mathbb{E}_Q\left[\left(\frac{v_{ij}}{v_{i0}+v_{ij}+\sum\limits_{k=1,k\neq j}^{m} v_{ik}\tilde{X}^b_{ij}}\right)\cdot\left(\frac{u_{ji}\tilde{X}^b_{ij}}{u_{j0}+u_{ji}}\right)\right], \tag{1.8.14b}$$

$$\geq \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}}{v_{i0}+v_{ij}+\sum\limits_{k=1,k\neq j}^{m} v_{ik}\tilde{X}^c_{ik}}\right)\cdot\mathbb{E}_{Q_j}\left[\left(\frac{u_{ji}\tilde{X}^b_{ij}}{u_{j0}+u_{ji}}\right)\right], \tag{1.8.14c}$$

$$\geq \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}}{v_{i0}+v_{ij}+\sum\limits_{k=1,k\neq j}^{m} v_{ik}\tilde{X}^c_{ik}}\right)\cdot\mathbb{E}_{Q_j}\left[\left(\frac{u^{min}\tilde{X}^b_{ij}}{1+u^{min}}\right)\right], \tag{1.8.14d}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\tilde{X}^c_{ij}}{v_{i0}+v_{ij}+\sum\limits_{k=1,k\neq j}^{m} v_{ik}\tilde{X}^c_{ik}}\right)\cdot\left(\frac{u^{min}}{1+u^{min}}\right), \tag{1.8.14e}$$

$$= \tilde{Z}^c\cdot(1-\alpha), \tag{1.8.14f}$$

$$(1-\alpha)\cdot\tilde{Z}^c\leq\mathbb{E}_Q[Z(\tilde{X}^b)]\leq Z(X^*). \tag{1.8.14g}$$

The second equality 1.8.14b follows from the observation that $\tilde{X}^b$ is binary and for each consumer j, $\sum_{l=1}^{n} \tilde{X}_{lj}^b \leq 1$. The first inequality 1.8.14c follows from Jensen's inequality and the fact that Q_j are independent. **Proposition 1.5.3**: For each supplier i, $\sum_{j=1}^{m} \tilde{\tilde{X}}_{ij} \leq 1$ and for each consumer j, $\sum_{i=1}^{n} \tilde{\tilde{X}}_{ij} \leq 1$.

Proof: The proof follows from the convexity argument used in proposition 1.5.1 □.

Theorem 1.5.3:$(1-\alpha)(1-\beta)Z(X^*) \leq Z(\tilde{\tilde{X}}) \leq Z(X^*)$.

Proof: Since $\tilde{\tilde{X}}$ is feasible w.r.t Z, we have $Z(\tilde{\tilde{X}}) \leq Z(X^*) \leq \tilde{\tilde{Z}}(\tilde{\tilde{X}})$.

$$\tilde{\tilde{Z}}(\tilde{\tilde{X}}) - Z(\tilde{\tilde{X}}) = \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\Bigg[\left(\frac{v_{ij}\cdot\tilde{\tilde{X}}_{ij}}{\sum_{k=1}^{m} v_{ik}\cdot\tilde{\tilde{X}}_{ik}}\right)\left(\frac{u_{ji}\cdot\tilde{\tilde{X}}_{ij}}{\sum_{l=1}^{n} u_{jl}\cdot\tilde{\tilde{X}}_{lj}}\right) - \left(\frac{v_{ij}\cdot\tilde{\tilde{X}}_{ij}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot\tilde{\tilde{X}}_{ik}}\right)\left(\frac{u_{ji}\cdot\tilde{\tilde{X}}_{ij}}{u_{j0}+\sum_{l=1}^{n} u_{jl}\cdot\tilde{\tilde{X}}_{lj}}\right)\Bigg], \tag{1.8.15a}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\cdot\tilde{\tilde{X}}_{ij}}{\sum_{k=1}^{m} v_{ik}\cdot\tilde{\tilde{X}}_{ik}}\right)\left(\frac{u_{ji}\cdot\tilde{\tilde{X}}_{ij}}{\sum_{l=1}^{n} u_{jl}\cdot\tilde{\tilde{X}}_{lj}}\right)\Bigg[1-\left(\frac{\sum_{k=1}^{m} v_{ik}\cdot\tilde{\tilde{X}}_{ik}}{v_{i0}+\sum_{k=1}^{m} v_{ik}\cdot\tilde{\tilde{X}}_{ik}}\right)\left(\frac{\sum_{l=1}^{n} u_{jl}\cdot\tilde{\tilde{X}}_{lj}}{u_{i0}+\sum_{l=1}^{n} u_{jl}\cdot\tilde{\tilde{X}}_{lj}}\right)\Bigg], \tag{1.8.15b}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\cdot\tilde{\tilde{X}}_{ij}}{\sum_{k=1}^{m} v_{ik}\cdot\tilde{\tilde{X}}_{ik}}\right)\left(\frac{u_{ji}\cdot\tilde{\tilde{X}}_{ij}}{\sum_{l=1}^{n} u_{jl}\cdot\tilde{\tilde{X}}_{lj}}\right)\Bigg[1-\left(\frac{v_{ij}}{v_{i0}+v_{ij}}\right)\left(\frac{u_{ji}}{u_{i0}+u_{ji}}\right)\Bigg], \tag{1.8.15c}$$

$$\leq \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\cdot\tilde{\tilde{X}}_{ij}}{\sum_{k=1}^{m} v_{ik}\cdot\tilde{\tilde{X}}_{ik}}\right)\left(\frac{u_{ji}\cdot\tilde{\tilde{X}}_{ij}}{\sum_{l=1}^{n} u_{jl}\cdot\tilde{\tilde{X}}_{lj}}\right)\left[1-\frac{v^{min}}{1+v^{min}}\frac{u^{min}}{1+u^{min}}\right] \tag{1.8.15d}$$

$$= \tilde{\tilde{Z}}(\tilde{\tilde{X}})\left[1-\frac{v^{min}}{1+v^{min}}\frac{u^{min}}{1+u^{min}}\right], \tag{1.8.15e}$$

$$(1-\alpha)(1-\beta)\tilde{\tilde{Z}}(\tilde{\tilde{X}}) \leq Z(\tilde{\tilde{X}}) \leq Z(X^*), \tag{1.8.15f}$$

where 1.8.15c follows from proposition 1.5.3 □.

1.8.1 Submodularity of One-sided Relaxation

Proposition 1.8.1. *If $r_{ij} = r_i$, $\forall i \in \mathcal{N}, j \in \mathcal{M}$, then the one-sided relaxation $\tilde{Z}$ (2.5.19a) is submodular.*

Proof: Consider the one-sided relaxation in 2.5.19a with $r_{ij} = r_i$:

$$\tilde{Z} = \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n}\sum_{j=1}^{m} r_i \left(\frac{v_{ij}\cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \middle| \sum_{i=1}^{n} X_{ij} \leq 1, \forall j \in \mathcal{M} \right\} \tag{1.8.16a}$$

$$= \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n} r_i \left(\frac{\sum_{j=1}^{m} v_{ij}\cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik}\cdot X_{ik}} \right) \middle| \sum_{i=1}^{n} X_{ij} \leq 1, \forall j \in \mathcal{M} \right\} \tag{1.8.16b}$$

Recall that, $\mathcal{S}_i^e$ denotes the assortment of consumers offered to supplier i and let $\mathcal{S}^e = \{\mathcal{S}_1^e, ..., \mathcal{S}_n^e\}$. Then 1.8.16b can be rewritten as

$$\tilde{Z} = \max_{\mathcal{S}^e} \left\{ \sum_{i=1}^{n} r_i \left(\frac{\sum_{j\in\mathcal{S}_i^e} v_{ij}}{v_{i0} + \sum_{k\in\mathcal{S}_i^e} v_{ik}} \right) \middle| \bigcap_{i=1}^{n} \mathcal{S}_i^e = \emptyset \right\} \tag{1.8.17}$$

For supplier i, denote $f_i(\mathcal{S}_i^e) = \frac{\sum_{j\in\mathcal{S}_i^e} v_{ij}}{v_{i0}+\sum_{k\in\mathcal{S}_i^e} v_{ik}}$. Then, it follows from Han et al. (2020) that $f_i(\mathcal{S}_i^e)$ is submodular for each i. And since $r_i \geq 0$, $\sum_{i=1}^{n} r_i \cdot f_i(\mathcal{S}_i^e)$ is also submodular □.

1.8.2 Uniform Distribution

Proposition 1.8.2. *If* $u_{j0} = \kappa$ *and* $u_{ji} \sim \mathcal{U}[\kappa_l, \kappa_u]$, $\forall j \in \mathcal{M}$ *where* $\kappa > 0$, $\kappa_u > \kappa_l > 0$, *then the randomized solution* $\tilde{X}^b$ *provides* $1 - \frac{\kappa}{\kappa_u - \kappa_l}\ln(\frac{\kappa_u+\kappa}{\kappa_l+\kappa})$ *approximation in expectation.*

Proof: Let $\mathbb{E}_U[\cdot]$ denote the expectation over random variable $U = \{u_{ji}\}$. From inequality 1.8.14c, we have

$$\mathbb{E}_U[\mathbb{E}_Q[Z(\tilde{X}^b)]] \geq \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}}{v_{i0} + v_{ij} + \sum\limits_{k=1,k\neq j}^{m} v_{ik}\tilde{X}^c_{ik}}\right) \cdot \mathbb{E}_U\left[\mathbb{E}_{Q_j}\left[\left(\frac{u_{ji}\tilde{X}^b_{ij}}{u_{j0} + u_{ji}}\right)\right]\right], \tag{1.8.18a}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\tilde{X}^c_{ij}}{v_{i0} + v_{ij} + \sum\limits_{k=1,k\neq j}^{m} v_{ik}\tilde{X}^c_{ik}}\right) \cdot \mathbb{E}_U\left[\frac{u_{ji}}{u_{j0} + u_{ji}}\right], \tag{1.8.18b}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\tilde{X}^c_{ij}}{v_{i0} + v_{ij} + \sum\limits_{k=1,k\neq j}^{m} v_{ik}\tilde{X}^c_{ik}}\right) \cdot \mathbb{E}_U\left[1 - \frac{u_{j0}}{u_{j0} + u_{ji}}\right], \tag{1.8.18c}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\left(\frac{v_{ij}\tilde{X}^c_{ij}}{v_{i0} + v_{ij} + \sum\limits_{k=1,k\neq j}^{m} v_{ik}\tilde{X}^c_{ik}}\right) \cdot \mathbb{E}_U\left[1 - \frac{\kappa}{\kappa + u_{ji}}\right], \tag{1.8.18d}$$

$$= \tilde{Z}^c\left(1 - \frac{\kappa}{\kappa_u - \kappa_l}\ln\left(\frac{\kappa_u + \kappa}{\kappa_l + \kappa}\right)\right), \tag{1.8.18e}$$

where 1.8.18e follows from the observation that, if $u_{ji} \sim \mathcal{U}[\kappa_l, \kappa_u]$ then $\kappa + u_{ji} \sim \mathcal{U}[\kappa + \kappa_l, \kappa + \kappa_u]$, and $\frac{1}{\kappa+u_{ji}}$ follows the inverse-uniform distribution, whose mean is given by $\frac{1}{\kappa_u-\kappa_l}\ln\left(\frac{\kappa_u+\kappa}{\kappa_l+\kappa}\right)$ $\square$.

Chapter 2

On Solving Discrete Fractional Programs and Its Applications to Assortment Optimization

2.1 Introduction

Linear fractional programs (LFP) involve maximizing the sum of linear ratios over a linear constraint set. The two standard classes of the LFP that find wide applications are the discrete 0-1 linear fractional programs (ZOFP) and the continuous linear fractional programs (CFP). Applications of ZOFP include but are not limited to assortment optimization (Méndez-Díaz et al., 2014; Bront et al., 2009), crew scheduling (Arora et al., 1977), information retrieval (Hansen et al., 1990), and k-choice facility location (Tawarmalani et al., 2002). Applications of CFP include cluster analysis (Rao, 1971), queuing-location (Drezner et al., 1990), and geometric programming problems (Chen et al., 2000), among others. Both ZOFP and CFP are computationally difficult (Borrero et al., 2017), except in special cases; for example, when the number of ratios is limited to one, the unconstrained ZOFP with a positive denominator and the unconstrained CFP is polynomially solvable. To this end, scalable algorithms that exploit different underlying structures have been developed for each class. However, beyond the observation that relaxing ZOFP leads to CFP, there is limited work that studies the underlying relationship between them.

In this paper, we consider a special class of ZOFP, called the *0-1 Fractional Programs with Cardinality Constraints over Subsets (ZOFP-CS)*. ZOFP-CS readily models a wide range of applications mentioned above, including the unconstrained and the cardinality-constrained assortment optimization under the mixture of multinomial logit choice (Désir et al., 2022), information retrieval (Hansen et al., 1990), the k-choice facility location problem (Tawarmalani et al., 2002) and as subproblems in choice-based network revenue management (Bront et al., 2009). We present an equivalent continuous reformulation called the *Reformulated Continuous Fractional Program with Cardinality Constraints over Subsets (RCFP-CS)* and show that any local optimal of RCFP-CS is integral. While the reformulation is computationally difficult, since the CFP is NP-hard, it allows us to apply the tools and methods commonly used for the continuous fractional program to the discrete problem. It also allows us to develop better bounds for a class of assortment optimization problems.

Our work is closely related to the literature on solving discrete problems via continuous relaxations, which typically involves computing approximate or sometimes even exact solutions by solving either a direct continuous relaxation or a continuous reformulation of the discrete problem. These methods provide additional insights into the discrete problem structure and help develop scalable solutions (see Pardalos, 1996) for a summary of continuous approaches for discrete optimization problems and their advantages). While they have been used extensively in the linear and quadratic integer programming literature, they have not been well-explored in the fractional programming domain.

The rest of the paper is organized as follows. In §3.2, we discuss relevant literature and highlight our contribution. In §2.3, we introduce the class of ZOFP-CS and the traditional benchmark solutions. In §2.4, we provide the reformulation and the bounds on direct relaxation. In §2.5, we provide applications by considering work in the assortment optimization literature. Finally, in §3.5, we perform numerical analysis and conclude in §3.6.

2.2 Related Work and Contribution

In this section, we first review the existing literature on the 0-1 linear fractional program and the continuous fractional programs. We then discuss recent work in assortment optimization that uses the continuous relaxation of the underlying discrete problem to compute approximate solutions with parametric bounds and conclude the section with a summary of our contributions.

2.2.1 Literature Review

Based on the number of ratio terms, the extant literature on ZOFP can broadly be categorized into two threads; the *single-ratio* and the *multiple-ratio* case. The single ratio case with a strictly positive denominator is polynomially solvable (Hammer et al., 1968; Boros and Hammer, 2002; Hansen et al., 1991), either when it is unconstrained or if the constraint space is such that any class of linear objective functions can be maximized over it in polynomial time (Megiddo, 1979; Avadhanula et al., 2016). In fact, the unconstrained case can be solved in closed form (Boros and Hammer, 2002).

The multiple ratio case, which is of primary interest in this paper, is computationally difficult even when the number of ratios is limited to two regardless of the constraint set and whether the denominator is strictly positive or not (Skiscim and Palocsay, 2001). Prokopyev et al. 2005 shows that it is NP-hard to compute a constant factor approximation for the unconstrained multiple ratio ZOFP. To this end, various algorithms have been proposed. One common approach is to formulate the problem as a mixed-integer linear programming problem which can be readily solved using off-the-shelf solvers (Tawarmalani et al., 2002; Li, 1994; Wu, 1997). Méndez-Díaz et al. 2014 develop valid inequalities to improve the MILP performance. More recently, Sen et al. 2018 propose a conic mixed integer programming formulation, in the context of assortment optimization, and show that it provides better runtime than the MILP formulation.

However, for large-scale problems, the above algorithms quickly become intractable. An alternative is to develop scalable upper and lower-bound solutions. For example, for assortment optimization under the mixture-of-multinomial logit model, which can be readily modeled as a 0-1 linear fractional program, Feldman and Topaloglu (2015) develop a Lagrange relaxation-based upper bound. They provide a parametric relaxation, discretize the parameter space, and solve the continuous relaxation of the resultant 0-1 knapsack problem. Kunnumkal and Martínez-de Albéniz (2019) improve the bound by characterizing the optimal value of the parameter. We develop a similar parametric Lagrange relaxation solution based on our reformulation and show that it improves upon the upper bound in Feldman and Topaloglu (2015). We characterize the optimal value of the parameter and additionally show that the resultant continuous knapsack problem has integral solutions.

A common approach for the lower bound solution is to use the local search heuristic, which starts from a feasible solution and moves greedily to a neighboring solution until a local optimum is obtained. For example, for choice-based network revenue management, Bront et al. (2009) formulate the column-generation subproblem as the unconstrained 0-1 fractional program and report that the heuristic provides close to the optimal solution for most problem instances. However, we observe in the numerical exercise that the heuristic performs poorly for the constrained case, and the reformulation can significantly improve the discrete local optima by utilizing the continuous solution space. Further, the performance gains increase as the constraints become tighter.

Similar to ZOFP, the extant literature on CFP can be categorized into the single-ratio and the multiple ratio case. The single ratio problem is polynomially solvable, and various algorithms have been developed that include Charnes and Cooper's transformation, which solves an equivalent linear program (Charnes and Cooper, 1962), the parametric approach by Dinkelbach (Dinkelbach, 1967), and the interior-point method (Freund and Jarre, 2001). The multiple ratio CFPs, on the other hand, are computationally difficult even when the number of ratios is limited to two (Matsui, 1996). However, efficient algorithms have been developed for the continuous case. For example, Kuno (Kuno, 2002, 2005) develops a branch-and-bound algorithm based on trapezoidal partitioning of the solution space when all the numerators and denominators are positive. Benson 2007 present a simplicial branch-and-bound algorithm that only assumes a non-zero denominator. Depetrini and Locatelli (2011) develop a fully-polynomial time approximation scheme (FPTAS) when the number of ratio terms is fixed. In both Benson (2007) and Depetrini and Locatelli (2011), the algorithm complexity depends on the number of ratio terms. While the reformulation we consider increases the number of ratio terms, we also show that it allows for the direct relaxation fractional solution to be rounded off with parametric guarantees.

In the context of assortment optimization, which can be readily modeled as a 0-1 linear fractional program, two recent papers (Caro et al., 2014; Ahmed et al., 2022) develop parametric approximation guarantees by solving the continuous relaxation of the underlying discrete problem. Caro et al. (2014) consider the assortment packing problem and formulate it as a fractional program. They then develop parametric bounds in expectation by solving the relaxed problem and using a randomized heuristic. We show that our reformulation provides tighter bounds. Ahmed et al. (2022) also provide parametric bounds in expectation using a similar approach for the two-sided assortment optimization problem. Our reformulation is similar to Ahmed et al. (2022). However, we show that the reformulation, in fact, gives an integral solution.

2.2.2 Contribution

In this paper, we consider the class of 0-1 fractional programs under cardinality-type constraints and make the following contributions:

i. We provide a continuous reformulation and show that any local optimum of the reformulation is integral.

ii. Since the reformulation increases the number of ratio terms, we consider the direct relaxation and show that the resultant fractional solution can be rounded off with parametric approximation guarantees.

iii. We provide a Lagrange relaxation based upper bound for assortment optimization under the mixture-of-multinomial and show that it improves upon the parametric discretization-based approach by Feldman and Topaloglu (2015).

iv. As corollaries, we derive tighter parametric bounds for the *assortment packing problem* (Caro et al., 2014) and the *two-sided assortment optimization* (Ahmed et al., 2022).

v. We provide an illustrative example to argue that the reformulation can significantly improve the discrete local optima by exploiting the continuous solution space. Numerically, we observe the performance gains to be as high as 60% when the constraints are tight.

To the best of our knowledge, our work is the first to solve a class of discrete fractional programs as a continuous fractional program. We believe this has broader implications for solving large discrete fractional programs, as it brings to bear the scalable methods associated with continuous optimization.

2.3 ZOFP-CS: Formulation and Standard Solutions

2.3.1 Formulation

Let $\mathbf{y} \in \{0,1\}^n$ be a n-dimensional binary variable, $S = \{S_1, ..., S_m\}$ be m mutually exclusive subsets of indices of $\mathbf{y}$ and let $M_j \in \mathbb{Z}^+$ be the cardinality size associated with each subset $S_j,\ j = \{1, ..., m\}$. Let a_i, c_i be n-dimensional vectors such that $a_{il}, c_{il} \geq 0, \forall l \in \{1, ..., n\}$. The 0-1 fractional program with cardinality constraints over subsets (ZOFP-CS) is given by:

$$\mathbf{ZOFP-CS}: \quad Z = \max_{\mathbf{y}\in\{0,1\}^n} \left\{ \sum_{i=1}^{p} \frac{a_i^T \cdot \mathbf{y}}{b_i + c_i^T \cdot \mathbf{y}} \;\middle|\; \sum_{k\in S_j} y_k \leq M_j,\ \forall j \in \{1, ..., m\} \right\}, \tag{2.3.1}$$

where we assume $b_i \geq 0$ and $p \geq 2$. Note that, although we limit our discussion to $\sum_{k \in S_j} y_k \leq M_j$, the results trivially extend to the case with equality constraints, i.e., $\sum_{k \in S_j} y_k = M_j$, as well as the unconstrained case (by letting $M_j \geq n$, or $M_j = 1$ and restricting each set S_j to contain at most one element, we obtain the class of unconstrained 0-1 fractional programs).

It is easy to check that ZOFP-CS models the unconstrained and the cardinality-constrained MMNL (Rusmevichientong et al., 2014), information retrieval (Hansen et al., 1990), and the k-choice facility location problems (Tawarmalani et al., 2002). However, in this work, we discuss two recent papers in assortment optimization literature in greater detail. Theoretically, these papers are of interest to us as they solve the discrete optimization problem using continuous relaxation and provide parametric bounds in expectation.

2.3.2 Standard Solutions

Observe that the inapproximability of ZOFP-CS immediately follows from the inapproximability of the mixture of multinomial logit (Désir et al., 2022). The two commonly used solutions for the ZOFP are the exact mixed-integer linear programming formulation and the heuristic local search solution. We use these solutions to benchmark the reformulation against the optimal solution and to highlight the performance improvement it offers over the discrete local maxima.

2.3.2.1 Mixed Integer Linear Programming Formulation.

Following Tawarmalani et al. 2002, we formulate the ZOFP-CS as MILP by defining the variable $x_i = \frac{1}{b_i + \sum\limits_{r=1}^{n} c_{ir} y_r}$ and linearize the product terms $w_{ir} = x_i y_r$ using big-M constraints. The resultant MILP is given by:

$$Z^m = \max_{w,x,y} \sum_{i=1}^{p} \sum_{r=1}^{n} a_{ir} \cdot w_{ir}, \tag{2.3.2a}$$

s.t.

$$\sum_{k \in S_j} y_k \leq M_j, \quad \forall j \in \{1, ..., m\}, \tag{2.3.2b}$$

$$b_i x_i + \sum_{r=1}^{n} c_{ir} w_{ir} = 1, \quad \forall i \in \{1, ..., p\}, \tag{2.3.2c}$$

$$w_{ir} \leq x_i, \quad \forall i \in \{1, ..., p\}, \forall r \in \{1, ..., n\}, \tag{2.3.2d}$$

$$w_{ir} \leq K y_r, \quad \forall i \in \{1, ..., p\}, \forall r \in \{1, ..., n\}, \tag{2.3.2e}$$

$$w_{ir} \geq x_i + K(y_r - 1), \quad \forall i \in \{1, ..., p\}, \forall r \in \{1, ..., n\}, \tag{2.3.2f}$$

$$w_{ir} \geq 0, \ y_i \in \{0, 1\}, \ x_i \in [0, 1] \quad \forall i \in \{1, ..., p\}, \forall r \in \{1, ..., n\}, \tag{2.3.2g}$$

where K is a large positive constant. Constraints (2.3.2d)-(2.3.2f) are linearization constraints and constraint (2.3.2c) ensures x_i definition is satisfied. The MILP formulation allows us to compute the optimal solution using off-the-shelf solvers, at least for smaller problem instances.

2.3.2.2 Local Search Heuristic.

Local search heuristics are commonly used in large-scale discrete optimization problems where computing the optimal solution might be computationally difficult. The heuristic starts from a feasible solution and moves greedily to a *neighboring solution* until it converges to a potentially local optima.

For ZOFP-CS, we construct the heuristic solution as follows. Given a feasible solution $\mathbf{y}^{ls} = (y_1^{ls}, ..., y_n^{ls}) \in \mathbb{B}^n$, let $\mathbf{y}^e = (\tilde{y}_1, ..., \tilde{y}_e, ..., \tilde{y}_n)$ such that $\tilde{y}_k = y_k^{ls}$ if $k \neq e$ and $\tilde{y}_k = 1 - y_k^{ls}$ otherwise, and $\mathcal{N}(\mathbf{y}^{ls}) = \{\mathbf{y}^e : \sum_{k \in S_j} \tilde{y}_k \leq M_j, \forall j \in \{1, ..., m\},\ \forall e \in \{1, ..., n\}\}$ be the feasible neighborhood of $\mathbf{y}^{ls}$. We start from the feasible solution $\mathbf{y}^{ls}$ and iteratively move to a neighboring point $\mathbf{y} \in \mathcal{N}(\mathbf{y}^{ls})$ such that $\mathbf{y} = \underset{\mathbf{y}^e \in \mathcal{N}(\mathbf{y}^{ls})}{\operatorname{argmax}}\ Z(\mathbf{y}^e)$ and $Z(\mathbf{y}) > Z(\mathbf{y}^{ls})$. If there is no point in the neighborhood of $\mathbf{y}^{ls}$ such that $Z(\mathbf{y}) > Z(\mathbf{y}^{ls})$, then $\mathbf{y}^{ls}$ is considered the *discrete local optimal solution* and the associated objective value denoted as Z^{ls} is the discrete local optima.

2.4 The Continuous Reformulation and The Direct Relaxation

We begin with a simple example to show that a direct relaxation of the binary variables y in ZOFP-CS leads to a fractional solution and introduce the reformulation for the example before discussing the general case.

Example 2.4.0.1. *Consider the following ZOFP-CS:*

$$Z = \max_{y \in \{0,1\}^n} \left\{ \frac{4y_1}{1+2y_1} + \frac{3y_2}{1+y_2} \middle| y_1 + y_2 \leq 1 \right\}. \tag{2.4.1}$$

It is easy to check that for the above ZOFP-CS, the optimal integral solution is given by $\{y_1 = 0,\ y_2 = 1\}$ with $Z = \frac{3}{2}$ and the optimal continuous relaxation solution is given by $\{y_1 = 0.415,\ y_2 = 0.584\}$ with the objective value of 2.01. We now reformulate the optimization as follows:

$$Z = \max_{y \in \{0,1\}^n} \left\{ \frac{4y_1}{1+2y_1} + \frac{3y_2}{1+y_2} \middle| y_1 + y_2 \leq 1 \right\}, \tag{2.4.2a}$$

$$= \max_{y \in \{0,1\}^n} \left\{ \frac{4y_1}{3} + \frac{3y_2}{2} \middle| y_1 + y_2 \leq 1 \right\}, \tag{2.4.2b}$$

$$= \max_{y \in [0,1]^n} \left\{ \frac{4y_1}{3} + \frac{3y_2}{2} \middle| y_1 + y_2 \leq 1 \right\}. \tag{2.4.2c}$$

Equation (2.4.2b) follows from the integrality of y, and (2.4.2c) follows since the objective is linear under cardinality type constraint with the optimal solution given by $\{y_1 = 0,\ y_2 = 1\}$ and $Z = \frac{3}{2}$.

2.4.1 Reformulation

We now discuss the above continuous reformulation for the general case:

$$Z = \max_{\mathbf{y} \in \{0,1\}^n} \left\{ \sum_{i=1}^{p} \frac{a_i^T \cdot \mathbf{y}}{b_i + c_i^T \cdot \mathbf{y}} \middle| \sum_{q \in S_j} y_q \leq M_j,\ \forall j \in \{1, ..., m\} \right\}, \tag{2.4.3a}$$

$$= \max_{\mathbf{y} \in \{0,1\}^n} \left\{ \sum_{i=1}^{p} \sum_{l=1}^{n} \frac{a_{il} \cdot y_l}{b_i + c_{il} y_l + \sum\limits_{k=1, k \neq l}^{n} c_{ik} \cdot y_k} \middle| \sum_{q \in S_j} y_q \leq M_j,\ \forall j \in \{1, ..., m\} \right\}, \tag{2.4.3b}$$

$$= \max_{\mathbf{y} \in \{0,1\}^n} \left\{ \sum_{i=1}^{p} \sum_{l=1}^{n} \frac{a_{il} \cdot y_l}{b_i + c_{il} + \sum\limits_{k=1, k \neq l}^{n} c_{ik} \cdot y_k} \middle| \sum_{q \in S_j} y_q \leq M_j,\ \forall j \in \{1, ..., m\} \right\}. \tag{2.4.3c}$$

Equation (2.4.3c) follows from (2.4.3b) since y_l is binary. The reformulation essentially makes the denominator free from the term in the numerator. The reformulated continuous fractional program associated with ZOFP-CS in (2.3.1) is given by:

$$\mathbf{RCFP-CS}: \quad Z^r = \max_{\mathbf{y}\in[0,1]^n} \left\{ \sum_{i=1}^{p} \sum_{l=1}^{n} \frac{a_{il} \cdot y_l}{b_i + c_{il} + \sum\limits_{k=1,k\neq l}^{n} c_{ik} \cdot y_k} \;\middle|\; \sum_{r\in S_j} y_r \le M_j,\ \forall j \in \{1,...,m\} \right\}. \tag{2.4.4}$$

RCFP-CS is computationally difficult since it is equivalent to the multiple-ratio 0-1 fractional program, which is known to be NP-hard. However, as we show below, RCFP-CS has integral local optima.

Let the discrete and continuous feasible solution space be Y^d and Y^c, respectively, defined as as follows:

$$Y^d = \{\mathbf{y} \in \{0,1\}^n : \sum_{r\in S_j} y_r \le M_j, \forall j \in \{1,...,m\}\}. \tag{2.4.5}$$

$$Y^c = \{\mathbf{y} \in [0,1]^n : \sum_{r\in S_j} y_r \le M_j, \forall j \in \{1,...,m\}\}. \tag{2.4.6}$$

For any feasible solution $\mathbf{y} \in Y^c$, let $Z(\mathbf{y})$ and $Z^r(\mathbf{y})$ be the objective values of ZOFP-CS and RCFP-CS, respectively:

$$Z(\mathbf{y}) = \sum_{i=1}^{p} \frac{\sum_{l=1}^{n} a_{il} \cdot y_l}{b_i + \sum_{k=1}^{n} c_{ik} \cdot y_k}, \tag{2.4.7a}$$

$$Z^r(\mathbf{y}) = \sum_{i=1}^{p} \sum_{l=1}^{n} \frac{a_{il} \cdot y_l}{b_i + c_{il} + \sum\limits_{k=1,k\neq l}^{n} c_{ik} \cdot y_k}. \tag{2.4.7b}$$

The reformulation provides a lower bound to ZOFP-CS, and the following lemma trivially follows:

Lemma 2.1. *For any* $\mathbf{y}^d \in Y^d$, $Z(\mathbf{y}^d) = Z^r(\mathbf{y}^d)$ *and for any* $\mathbf{y}^c \in Y^c$, $Z(\mathbf{y}^c) \ge Z^r(\mathbf{y}^c)$.

We now present the main result of the paper, which states that given any fractional solution $\mathbf{y}^c \in Y^c$, the objective value $Z^r(\mathbf{y}^c)$ can be improved by iteratively rounding off the fractional values. Let $\mathbf{y}^c_{-k} = \{y^c_l : l \neq k, l = \{1, ..., n\}\}$, i.e., $\mathbf{y}^c_{-k}$ is the set of variables except y_k, $Z^r(y_k|\mathbf{y}^c_{-k})$ be the objective as a function of y_k, given the remaining variables $\mathbf{y}^c_{-k}$ and $B_j(\mathbf{y}^c)$ be the number of fractional terms of $\mathbf{y}^c$ with indices in subset S_j, then:

Theorem 2.4.1. *Given any fractional solution $\mathbf{y}^c \in Y^c$ with $B_j(\mathbf{y}^c)$ fractional terms, $\exists \tilde{\mathbf{y}}^c \in Y^c$ with $B_j(\mathbf{y}^c) - 1$ fractional terms such that $Z^r(\tilde{\mathbf{y}}^c) \geq Z^r(\mathbf{y}^c)$.*

Proof: We proceed by arguing that given a continuous relaxation solution $\mathbf{y}^c$, with $B_j(\mathbf{y}^c)$ number of fractional variables with indices in subset S_j, the objective can be increased by reducing fractional terms to $B_j(\mathbf{y}^c) - 1$. Consider the base case with $B_j(\mathbf{y}^c) = 1$ for some $j \in \{1, ..., m\}$. Without loss of generality, assume index $1 \in S_j$, such that $y^c_1 \in (0, 1)$, then given $\mathbf{y}^c_{-1}$, $Z^r(y_1|\mathbf{y}^c_{-1})$ can be written as:

$$Z^r(y_1|\mathbf{y}^c_{-1}) = \sum_{i=1}^{p} \left\{ \frac{a_{i1} \cdot y_1}{b_i + c_{i1} + \sum\limits_{r \geq 2}^{n} c_{ir} \cdot y^c_r} + \sum_{l \geq 2}^{n} \frac{a_{il} \cdot y^c_l}{b_i + c_{il} + c_{i1} y_1 + \sum\limits_{r \geq 2, r \neq l}^{n} c_{ir} \cdot y^c_r} \right\}. \quad (2.4.8)$$

The first term in (2.4.8) is linear, and the second term is convex in y_1. It follows that $Z^r(y_1|\mathbf{y}^c_{-1})$ is also convex, as the summation of convex functions is convex. Therefore, maximizing $Z^r(y_1|\mathbf{y}^c_{-1})$ ensures $y_1 \in \{0, 1\}$, since maximizing a convex function over closed, convex set leads to boundary solutions. Note that the constraint $\sum_{r \in S_j} y_r \leq M_j$ is not violated by rounding off y_1 since we assume M_j to be integral.

Consider the case when $B_j(y^c) > 1$ for some $j \in \{1, ..., m\}$. Without loss of generality, assume indices $1, 2 \in S_j$, such that $y_1^c, y_2^c \in (0, 1)$ and $y_1^c + y_2^c = \epsilon$. Then given $y_r^c, r \notin \{1, 2\}$, Z^r can be written as:

$$Z^r(y_1, y_2|\mathbf{y}^c_{-\{1,2\}}) = \begin{array}{r} \sum_{i=1}^{p} \left\{ \dfrac{a_{i1} \cdot y_1}{b_i + c_{i1} + c_{i2}y_2 + \sum_{r\geq 3}^{n} c_{ir} \cdot y_r^c} + \dfrac{a_{i2} \cdot y_2}{b_i + c_{i1}y_1 + c_{i2} + \sum_{r\geq 3}^{n} c_{ir} \cdot y_r^c} + \right. \\ \left. \sum_{l\geq 3}^{n} \dfrac{a_{il} \cdot y_l^c}{b_i + c_{i1}y_1 + c_{i2}y_2 + c_{il} + \sum_{r\geq 3, r\neq l}^{n} c_{ir} \cdot y_r^c} \right\} \end{array}, \tag{2.4.9a}$$

$$= \sum_{i=1}^{p} \left\{ \frac{a_{i1} \cdot y_1}{\tilde{b}_{i1} + c_{i2}y_2} + \frac{a_{i2} \cdot y_2}{\tilde{b}_{i2} + c_{i1}y_1} + \sum_{l\geq 3}^{n} \frac{\tilde{a}_{il}}{\tilde{b}_{il} + c_{i1}y_1 + c_{i2}y_2} \right\}, \tag{2.4.9b}$$

where $\tilde{a}_{il} = a_{il}y_l^c$ and $\tilde{b}_{il} = b_i + c_{il} + \sum_{r\geq 3, r\neq l}^{n} c_{ir} \cdot y_r^c$. Since we assume $y_1^c + y_2^c = \epsilon$, $Z^r(y_1, y_2|y^c_{-\{1,2\}})$ can be rewritten as:

$$Z^r(y_1|\mathbf{y}^c_{-\{1,2\}}, \epsilon) = \begin{array}{r} \sum_{i=1}^{p} \left\{ \dfrac{a_{i1} \cdot y_1}{\tilde{b}_{i1} + c_{i2}(\epsilon - y_1)} + \dfrac{a_{i2} \cdot (\epsilon - y_1)}{\tilde{b}_{i2} + c_{i1}y_1} + \right. \\ \left. \sum_{l\geq 3}^{n} \dfrac{\tilde{a}_{il}}{\tilde{b}_{il} + c_{i1}y_1 + c_{i2}(\epsilon - y_1)} \right\} \end{array}, \tag{2.4.10a}$$

where $y_1 \in [0, \epsilon]$ if $\epsilon \leq 1$ and $y_1 \in [\epsilon - 1, 1]$ if $\epsilon > 1$. We now show each of the three terms is convex.

Let $R_{i1}(y_1) = \frac{a_{i1} \cdot y_1}{\tilde{b}_{i1} + c_{i2}(\epsilon - y_1)}$. Then

$$\frac{\mathrm{d}R_{i1}(y_1)}{\mathrm{d}y_1} = \frac{(\tilde{b}_{i1} + c_{i2}(\epsilon - y_1))a_{i1} - (a_{i1}y_1)(-c_{i2})}{(\tilde{b}_{i1} + c_{i2}(\epsilon - y_1))^2}, \tag{2.4.11a}$$

$$= \frac{a_{i1}\tilde{b}_{i1} + \epsilon \cdot a_{i1}c_{i2}}{(\tilde{b}_{i1} + c_{i2}(\epsilon - y_1))^2}. \tag{2.4.11b}$$

$$\frac{\mathrm{d}^2 R_{i1}(y_1)}{\mathrm{d}y_1^2} = \frac{2c_{i2}(a_{i1}\tilde{b}_{i1} + \epsilon \cdot a_{i1}c_{i2})}{(\tilde{b}_{i1} + c_{i2}(\epsilon - y_1))^3}, \tag{2.4.12a}$$

$$\geq 0. \tag{2.4.12b}$$

Let $R_{i2}(y_1) = \frac{a_{i2} \cdot (\epsilon - y_1)}{\tilde{b}_{i2} + c_{i1}y_1}$. Then

$$\frac{\mathrm{d}R_{i2}(y_1)}{\mathrm{d}y_1} = \frac{(\tilde{b}_{i2} + c_{i1}y_1)(-a_{i2}) - a_{i2}(\epsilon - y_1)c_{i1}}{(\tilde{b}_{i2} + c_{i1}y_1)^2}, \tag{2.4.13a}$$

$$= \frac{-a_{i2}\tilde{b}_{i2} - \epsilon \cdot a_{i2}c_{i1}}{(\tilde{b}_{i2} + c_{i1}y_1)^2}. \tag{2.4.13b}$$

$$\frac{\mathrm{d}^2 R_{i2}(y_1)}{\mathrm{d}y_1^2} = \frac{2c_{i1}(a_{i2}\tilde{b}_{i2} + \epsilon \cdot a_{i2}c_{i1})}{(\tilde{b}_{i2} + c_{i1}y_1)^3}, \tag{2.4.14a}$$

$$\geq 0. \tag{2.4.14b}$$

Let $R_{i3}(y_1) = \frac{\tilde{a}_{il}}{\tilde{b}_{il} + c_{i1}y_1 + c_{i2}(\epsilon - y_1)}$. Then

$$\frac{\mathrm{d}R_{i3}(y_1)}{\mathrm{d}y_1} = \frac{-\tilde{a}_{il}(c_{i1} - c_{i2})}{(\tilde{b}_{il} + c_{i1}y_1 + c_{i2}(\epsilon - y_1))^2}. \tag{2.4.15a}$$

$$\frac{\mathrm{d}^2 R_{i3}(y_1)}{\mathrm{d}y_1^2} = \frac{2\tilde{a}_{il}(c_{i1} - c_{i2})^2}{(\tilde{b}_{i2} + c_{i1}y_1 + c_{i2}(\epsilon - y_1))^3}, \tag{2.4.16a}$$

$$\geq 0. \tag{2.4.16b}$$

Since the second-order derivatives are positive, the objective is convex in y_1. Therefore, $Z^r(y_1|\mathbf{y}^c_{-\{1,2\}}, \epsilon)$ attains its maximum at $(y_1, y_2) = (0, \epsilon)$ or $(y_1, y_2) = (\epsilon, 0)$ for $\epsilon \leq 1$ and for $\epsilon > 1$, $(y_1, y_2) = (\epsilon - 1, 1)$ or $(y_1, y_2) = (1, \epsilon - 1)$. Thus, the continuous relaxation solutions can be improved by appropriately rounding off the fractional terms pair-wise. Again note that rounding off the values does not violate any constraint. □.

Algorithm 1: Greedy roundoff.

Input : $\mathbf{y}^c \in Y^c$
Output: $y^d \in Y^d : Z^r(y^d) \geq Z^r(\mathbf{y}^c)$

$\mathbf{y}^d \leftarrow \mathbf{y}^c$;
$B_j(y^d) \leftarrow |\{k \mid y^d_k \in (0,1), k \in S_j\}|, \forall j \in \{1, ..., m\}$;
while $\exists j \in \{1, ..., m\} \mid B_j(\mathbf{y}^d) \geq 1$ **do**
    $l \leftarrow j \mid B_j(\mathbf{y}^d) \geq 1, j \in \{1, ..., m\}$;
    if $B_l(\mathbf{y}^d) = 1$ **then**
        $k \leftarrow q \mid y^d_q \in (0,1), q \in S_l$;
        if $Z^r(y^d_k = 1|y^d_{-k}) \geq Z^r(y^d_k = 0|\mathbf{y}^d_{-k})$ **then** $y^d_k \leftarrow 1$ **else** $y^d_k \leftarrow 0$;
    else
        $k_1, k_2 \leftarrow q \mid y^d_q \in (0,1), q \in S_l$;
        $\epsilon \leftarrow y_{k_1} + y_{k_2}$;
        if $\epsilon \leq 1$ **then**
            if $Z^r(y^d_{k_1} = \epsilon, y^d_{k_2} = 0|\mathbf{y}^d_{-k}) \geq Z^r(y^d_{k_1} = 0, y^d_{k_2} = \epsilon|\mathbf{y}^d_{-k})$ **then** $(y^d_{k_1} y^d_{k_2}) \leftarrow (\epsilon, 0)$ **else** $(y^d_{k_1} y^d_{k_2}) \leftarrow (0, \epsilon)$;
        else
            if $Z^r(y^d_{k_1} = \epsilon - 1, y^d_{k_2} = 1|\mathbf{y}^d_{-k}) \geq Z^r(y^d_{k_1} = 1, y^d_{k_2} = \epsilon - 1|\mathbf{y}^d_{-k})$ **then** $(y^d_{k_1} y^d_{k_2}) \leftarrow (\epsilon - 1, 1)$ **else** $(y^d_{k_1} y^d_{k_2}) \leftarrow (1, \epsilon - 1)$;
        end
    end
end

The proof is constructive as it provides us with a rounding scheme. Given any feasible fractional solution, a discrete integer solution with a higher objective value can be obtained by rounding off values iteratively. We outline the greedy roundoff steps in algorithm 1. We select a subset S_j with one or more indices having fractional values. If the

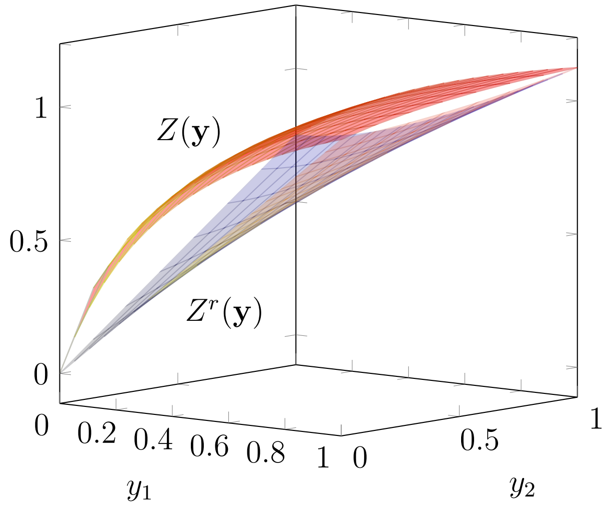

FIGURE 2.1: $Z(\mathbf{y}) = \frac{3y_1}{1+2y_1+y_2} + \frac{3y_2}{1+4y_1+3y_2}$, $Z^r(\mathbf{y}) = \frac{3y_1}{3+y_2} + \frac{3y_2}{4+4y_1}$.

subset has exactly one index with a fractional value, it is rounded off to 0 or 1, whichever gives a higher objective value (with respect to $Z^r(\mathbf{y})$, holding all other values constant. If the subset has two or more indices with fractional values that add to $\epsilon \leq 1$, they are rounded-off to $(0, \epsilon)$ or $(\epsilon, 0)$ and if $\epsilon \geq 1$, they are rounded-off to $(\epsilon - 1, 1)$ or $(1, \epsilon - 1)$, whichever gives a higher objective value.

Corollary 2.4.1. *Given any fractional solution* $\mathbf{y}^c \in Y^c$*, Algorithm 1 return an integral solution* $\mathbf{y}^d \in Y^d$*, such that* $Z^r(\mathbf{y}^d) \geq Z^r(\mathbf{y}^c)$.

Alternatively, the RCFP-CS can be solved using any off-the-shelf nonlinear solver with the fractional solution as an initial input to obtain a locally optimal integer solution.

While maximizing the reformulation gives integral solutions, it increases the number of ratio terms from p to $p \cdot n$. To this end, we consider the direct relaxation of ZOFP-CS and show that the resultant fractional solution can be rounded off with parametric approximation guarantees.

2.4.2 Direct Relaxation

We first discuss a simple example shown in figure 2.1 where we consider $Z(\mathbf{y}) = \frac{3y_1}{1+2y_1+y_2} + \frac{3y_2}{1+4y_1+3y_2}$ and its continuous reformulation $Z^r(y) = \frac{3y_1}{3+y_2} + \frac{3y_2}{4+4y_1}$. As stated in lemma 2.1, the functions coincide at integral points, and $Z^r(\mathbf{y})$ is a lower bound on $Z(\mathbf{y})$. It follows from simple algebraic manipulation that the relative difference between the first terms in $Z(\mathbf{y})$ and $Z^r(\mathbf{y})$ is bounded by $\frac{2(1-y_1)}{3+y_2}$ and by $\frac{3(1-y_2)}{4+4y_1}$ for the second terms. It immediately follows that the relative gap between $Z(\mathbf{y})$ and $Z^r(\mathbf{y})$ is bounded $\frac{3}{4}$.

We use a similar approach for the general case and first quantify the gap between the reformulation and the direct relaxation. From lemma 2.1, for any feasible fractional solution $\mathbf{y}^c \in Y^c$, $Z(\mathbf{y}^c) \geq Z^r(\mathbf{y}^c)$. Let $\gamma = \max_{i,j}\{\frac{c_{ij}}{b_i}\}$, then the following result holds:

Proposition 2.4.1. *For any fractional solution* $\mathbf{y}^c \in Y^c$, $Z^r(\mathbf{y}^c) \geq \frac{1}{1+\gamma} Z(\mathbf{y}^c)$.

Proof: From lemma 2.1, we have $Z(\mathbf{y}^c) \geq Z^r(\mathbf{y}^c)$.

$$Z(\mathbf{y}^c) - Z^r(\mathbf{y}^c) = \sum_{i=1}^{p}\sum_{j=1}^{n}\Big\{\frac{a_{ij}y_j^c}{b_i + c_{ij}y_j^c + \sum_{k\neq j} c_{ik}y_k^c} - \frac{a_{ij}y_j^c}{b_i + c_{ij} + \sum_{k\neq j} c_{ik}y_k^c}\Big\}, \tag{2.4.17a}$$

$$= \sum_{i=1}^{p}\sum_{j=1}^{n}\frac{a_{ij}y_j^c}{b_i + c_{ij}y_j^c + \sum_{k\neq j} c_{ik}y_k^c}\Big\{1 - \frac{b_i + c_{ij}y_j^c + \sum_{k\neq j} c_{ik}y_k^c}{b_i + c_{ij} + \sum_{k\neq j} c_{ik}y_k^c}\Big\}, \tag{2.4.17b}$$

$$= \sum_{i=1}^{p}\sum_{j=1}^{n}\frac{a_{ij}y_j^c}{b_i + c_{ij}y_j^c + \sum_{k\neq j} c_{ik}y_k^c}\Big\{\frac{c_{ij}(1-y_j^c)}{b_i + c_{ij} + \sum_{k\neq j} c_{ik}y_k^c}\Big\}, \tag{2.4.17c}$$

$$\leq \sum_{i=1}^{p}\sum_{j=1}^{n}\frac{a_{ij}y_j^c}{b_i + c_{ij}y_j^c + \sum_{k\neq j} c_{ik}y_k^c}\Big\{\frac{c_{ij}}{b_i + c_{ij}}\Big\}, \tag{2.4.17d}$$

$$\leq \sum_{i=1}^{p}\sum_{j=1}^{n}\frac{a_{ij}y_j^c}{b_i + c_{ij}y_j + \sum_{k\neq j} c_{ik}y_k^c}\Big\{\frac{\gamma}{1+\gamma}\Big\}, \tag{2.4.17e}$$

$$= Z(\mathbf{y}^c)\Big(\frac{\gamma}{1+\gamma}\Big), \tag{2.4.17f}$$

$$Z^r(\mathbf{y}^c) \geq \frac{1}{1+\gamma} Z(\mathbf{y}^c).\square \tag{2.4.17g}$$

Let $\tilde{\mathbf{y}}^c$ be the solution obtained by direct relaxation of ZOFP-CS i.e., $\tilde{\mathbf{y}}^c = \text{argmax}_{\mathbf{y}\in[0,1]^n} Z(y)$ and $\tilde{\mathbf{y}}^d$ be the integral solution obtained by rounding off the fractional values as outlined in algorithm 1, then:

Theorem 2.4.2. *$Z(\tilde{\mathbf{y}}^d) \geq \frac{1}{1+\gamma} Z$.*

Proof: Since $\tilde{\mathbf{y}}^c$ is the optimal solution of the relaxed problem, we have $Z(\tilde{\mathbf{y}}^c) \geq Z$ and from proposition 2.4.1, we have $Z^r(\tilde{\mathbf{y}}^c) \geq \frac{1}{1+\gamma} Z(\tilde{\mathbf{y}}^c)$. Using corollary 2.4.1 and lemma 2.1 we get the desired result. $\square$

Thus, the reformulation allows us to solve the discrete ZOFP-CS either exactly as the continuous fractional program RCFP-CS or approximately by solving the direct relaxation and rounding off the resultant fractional solution.

2.5 Applications: Tighter Bounds and Improved Local Maxima

We now consider three applications. We first consider the assortment optimization under MMNL and develop a Lagrange relaxation-based solution and show that it is tighter than the parametric discretization-based approach by Feldman and Topaloglu (2015). We then consider the assortment-packing and the two-sided assortment optimization problem and show that our reformulation improves existing parametric bounds. Finally, we illustrate that the reformulation can improve discrete local maxima by utilizing the continuous solution space.

Tighter Upper bounds for the Mixture of Multinomial Logit Model

The assortment optimization under the mixture of Multinomial logit models is characterized by n_c customer types, m products, the probability θ_i of observing customer-type i, their preference v_{ij} for product j and their preference for outside option v_{i0}. The platform offers a single assortment, and the customers select at most one product from the offered set based on the Logit-choice model. The platform obtains reward r_j if product j is selected, and its objective is to maximize the expected revenue. Let X_j be a binary variable that takes value one if product j is included in the assortment and zero otherwise. The platforms expected revenue given an assortment $X = \{X_j \in \{0,1\}, \forall j \in 1,...,m\}$ is:

$$V(X) = \sum_{i=1}^{n_c} \theta_i \frac{\sum\limits_{j=1}^{m} r_j v_{ij} \cdot X_j}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_k}, \tag{2.5.1a}$$

$$V^* = \max_{X \in \{0,1\}^n} V(X). \tag{2.5.1b}$$

For a single customer-type, the problem reduces to standard assortment optimization with multinomial logit choice, and can be solved efficiently. However, for $n_c > 1$, the problem is known to be NP-hard and various approximation algorithms have been developed.

Feldman and Topaloglu (2015) propose a Lagrange relaxation-based solution to compute upper bounds on the expected revenue. They define auxiliary variables X_j^i for each customer type i, product j and introduce the constraint $X_j^i = X_j^{\phi}, \forall i, j$. Let λ_j^i be the Lagrange multiplier associated with the above constraint. They provide the following Lagrangian relaxation:

$$V(X,\lambda) = \sum_{i=1}^{n_c} \left\{ \theta_i \frac{\sum\limits_{j=1}^{m} r_j v_{ij} \cdot X_j^i}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_k^i} \right\} - \sum_j \lambda_j^i (X_j^i - X_j^\phi), \tag{2.5.2a}$$

$$= \sum_{i=1}^{n_c} \left\{ \sum_{j=1}^{m} \left[\frac{\theta_i r_j v_{ij} \cdot X_j^i}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_k^i} - \lambda_j^i X_j^i \right] \right\} + \sum_{j=1}^{m} (\sum_{i=1}^{n_c} \lambda_j^i) X_j^\phi, \tag{2.5.2b}$$

$$V^{UB}(\lambda) = \max_{X_k^i \in \{0,1\}} V(X,\lambda) \tag{2.5.2c}$$

Given λ, $V^{UB}(\lambda)$ provides a valid upper bound to (2.5.1b). To obtain tight bounds, Feldman and Topaloglu (2015) solve

$$V^{UB} = \min_{\lambda : \sum_{i=1}^{n_c} \lambda_j^i = 0} V^{UB}(\lambda). \tag{2.5.3}$$

where, $\sum_{i=1}^{n_c} \lambda_j^i = 0$ ensures 2.5.3 is feasible.

Given λ, $V(X,\lambda)$ is separable across customer types. However, the inner optimization problem is still difficult to compute. Feldman and Topaloglu (2015), therefore, propose a discretization-based approach. Denote $t_i = v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_k$, and let the lower and upper bounds be $t_i^l = v_{i0}$ and $t_i^u = v_{i0} + \sum_k v_{ik}$, respectively. The interval $t_i \in [t_i^l, t_i^u]$ is discretized into subintervals $[t_i^g, t_i^{g+1}], \forall g \in \{1, ..., G\}$ where G is a fixed constant and $t_i^1 = t_i^l$, $t_i^{G+1} = t_i^u$. The separable parametric reformulation of (2.5.2b) is given by:

$$V^{UB}(\lambda) = \sum_{i=1}^{n_c} \max_{t_i \in [t_i^l, t_i^u]} \max_{X \in \{0,1\}^n} \left\{ \sum_{j=1}^{m} \left(\frac{\theta_i r_j v_{ij}}{t_i} - \lambda_j^i \right) X_j^i \middle| v_i + \sum_{k=1}^{m} v_{ik} \cdot X_k^i \leq t_i, \forall i \right\}, \tag{2.5.4a}$$

$$\leq \sum_{i=1}^{n_c} \max_{g} \max_{X \in \{0,1\}^n} \left\{ \sum_{j=1}^{m} \left(\frac{\theta_i r_j v_{ij}}{t_i^g} - \lambda_j^i \right) X_j^i \middle| v_i + \sum_{k=1}^{m} v_{ik} \cdot X_k^i \leq t_i^{g+1}, \forall i \right\}, \tag{2.5.4b}$$

$$\leq \sum_{i=1}^{n_c} \max_{g} \max_{X\in[0,1]^n} \left\{ \sum_{j=1}^{m} \left(\frac{\theta_i r_j v_{ij}}{t_i^g} - \lambda_j^i \right) X_j^i \middle| v_i + \sum_{k=1}^{m} v_{ik} \cdot X_k^i \leq t_i^{g+1}, \forall i \right\}. \quad (2.5.4c)$$

The first inequality follows from the observation that for each interval, the inner optimization is solved by replacing t_i with its lower bound in the objective and its upper bound in the constraint. The second inequality follows since the integrality constraint on X is relaxed. Therefore, for a given λ and g, Feldman and Topaloglu (2015) solve the continuous knapsack (2.5.4c) to obtain an upper bound.

We observe that three steps add to the gap between V^* and V^{UB}. First is the Lagrange relaxation step, and the second is the discretization step, where, given λ, instead of solving with respect to the optimal value of t_i, the optimization is done over discrete intervals by considering the lower or upper bounds of t_i. Finally, given λ, g, the 0-1 knapsack is solved as a continuous knapsack. We now use our reformulation to develop an alternative relaxation that avoids the last two steps, i.e., given a set of Lagrange multipliers, we essentially solve the inner optimization to optimality, in fact, in closed form by characterizing the optimal t, X.

The reformulated MMNL is given by:

$$\tilde{V}(X) = \sum_{i=1}^{n_c} \sum_{j=1}^{m} \frac{\theta_i r_j v_{ij} \cdot X_j}{v_{i0} + v_{ij} + \sum_{k\neq j}^{m} v_{ik} \cdot X_k}, \quad (2.5.5a)$$

$$\tilde{V}^* = \max_{X\in[0,1]^n} \tilde{V}(X). \quad (2.5.5b)$$

We introduce auxiliary variables X_k^{ij}, $X_k^{\phi}, \forall i \in 1, ..., n,\ \forall j, k \in \{1, ..., m\}$ and the constraint $X_k^{ij} = X_k^{\phi}, \forall i, j, k$. Let $\tilde{\lambda}_k^{ij}$ be the Lagrange multiplier associated with the above constraint. The Lagrange relaxation is given by:

$$\tilde{V}(X,\tilde{\lambda}) = \sum_{i=1}^{n_c}\sum_{j=1}^{m}\Big[\frac{\theta_i r_j v_{ij}\cdot X_j^{ij}}{v_{i0}+v_{ij}+\sum_{k\neq j}^{m} v_{ik}\cdot X_k^{ij}}\Big] - \sum_k\sum_{i,j}\tilde{\lambda}_k^{ij}(X_k^{ij}-X_k^{\phi}), \quad (2.5.6a)$$

$$= \sum_{i=1}^{n_c}\sum_{j=1}^{m}\Big[\frac{\theta_i r_j v_{ij}\cdot X_j^{ij}}{v_{i0}+v_{ij}+\sum_{k\neq j}^{m} v_{ik}\cdot X_k^{ij}} - \sum_k \tilde{\lambda}_k^{ij}X_k^{ij}\Big] + \sum_k(\sum_{i,j}\tilde{\lambda}_k^{ij})X_k^{\phi}, \quad (2.5.6b)$$

$$\tilde{V}^{UB}(\tilde{\lambda}) = \max_{X\in[0,1]^n}\tilde{V}(X,\tilde{\lambda}). \quad (2.5.6c)$$

Given $\tilde{\lambda}$, it is easy to check that $\tilde{V}^{UB}(\tilde{\lambda})$ is a valid upper bound for $\tilde{V}^*$. To obtain a tight upper bound, we minimize with respect to $\tilde{\lambda}$:

$$\tilde{V}^{UB} = \min_{\tilde{\lambda}:\sum_{i=1}^{n_c}\sum_{j=1}^{m}\tilde{\lambda}_k^{ij}=0} V^{UB}(\lambda). \quad (2.5.7)$$

where, $\sum_{i=1}^{n_c}\sum_{j=1}^{m}\tilde{\lambda}_k^{ij}=0$ ensures (2.5.7) is feasible.

We introduce parameter $t_{ij} = v_{i0}+v_{ij}+\sum_{k\neq j}^{m} v_{ik}\cdot X_k^{ij}$ with the lower bound given by $t_{ij}^l = v_{i0}+v_{ij}$ and the upper bound $t_{ij}^u = v_{i0}+v_{ij}+\sum_{k\neq j}^{m} v_{ik}$. The parametric reformulation of (2.5.6b) is given by:

$$\tilde{V}^{UB}(\tilde{\lambda}) = \sum_{i=1}^{n_c}\sum_{j=1}^{m}\max_{t_{ij}\in[t_{ij}^l,t_{ij}^u]}\max_{X_k^{ij}\in[0,1]}\left\{\frac{\theta_i r_j v_{ij}\cdot X_j^{ij}}{t_{ij}} - \sum_k\tilde{\lambda}_k^{ij}X_k^{ij}\ \middle|\ v_{i0}+v_{ij}+\sum_{k\neq j}^{m} v_{ik}\cdot X_k^{ij}\leq t_{ij}\right\}, \quad (2.5.8a)$$

$$= \sum_{i=1}^{n_c}\sum_{j=1}^{m}\max_{t_{ij}\in[t_{ij}^l,t_{ij}^u]}\max_{X_k^{ij}\in[0,1]}\left\{\Big(\frac{\theta_i r_j v_{ij}}{t_{ij}} - \lambda_j^{ij}\Big)X_j^{ij} - \sum_{k\neq j}\tilde{\lambda}_k^{ij}X_k^{ij}\ \middle|\ v_{i0}+v_{ij}+\sum_{k\neq j}^{m} v_{ik}\cdot X_k^{ij}\leq t_{ij}\right\}. \quad (2.5.8b)$$

We now show that given $\tilde{\lambda}$, $V^{UB}(\lambda)$ can be solved efficiently, in fact, in closed form. To give an overview, We proceed by first observing that given t_{ij}, the problem reduces to a continuous knapsack and can be solved efficiently by rank-ordering with respect to $\frac{-\tilde{\lambda}_k^{ij}}{v_{ik}}, \forall k \neq j$. We then show that we only need to consider $O(n)$ discrete set of values for t_{ij} and characterize this set. It follows from the same result that at these discrete values of t_{ij}, X_k^{ij} is integral.

For a given i, j, let $\tilde{V}^{UB}(\tilde{\lambda})$ be the inner optimization defined in (2.5.8b):

$$\tilde{V}_{ij}^{UB}(\tilde{\lambda}) = \max_{t_{ij}\in[t_{ij}^l, t_{ij}^u]} \max_{X_k^{ij}\in[0,1]} \left\{ \left(\frac{\theta_i r_j v_{ij}}{t_{ij}} - \tilde{\lambda}_j^{ij}\right) X_j^{ij} - \sum_{k\neq j} \tilde{\lambda}_k^{ij} X_k^{ij} \;\middle|\; v_{i0} + v_{ij} + \sum_{k\neq j}^{m} v_{ik} \cdot X_k^{ij} \leq t_{ij} \right\}. \tag{2.5.9}$$

Without loss of generality, assume the indices $k = \{1, ..., j-1, j+1, ..., n\}$ are rank ordered with respect to $\frac{-\tilde{\lambda}_k^{ij}}{v_{ik}}, \forall k \neq j$ i.e., $\frac{-\tilde{\lambda}_1^{ij}}{v_{i1}} \geq \frac{-\tilde{\lambda}_2^{ij}}{v_{i2}} \geq \frac{-\tilde{\lambda}_{j-1}^{ij}}{v_{i,j-1}} \geq \frac{-\tilde{\lambda}_{j+1}^{ij}}{v_{i,j+1}} \geq ... \geq \frac{-\tilde{\lambda}_n^{ij}}{v_{in}}$. Denote the optimal value of t_{ij} with t_{ij}^* and let $\Gamma_{ij} = \{v_{i0} + v_{ij}, v_{i0} + v_{ij} + v_{i1}, ..., v_{i0} + v_{ij} + \sum_{k\neq j} v_{ik}\}$. For simplicity, let $\tilde{r}_{ij} = \theta_i r_j v_{ij}$.

Proposition 2.5.1. *At optimality,* $X_j^{ij} \in \{0, 1\}$ *and* $t_{ij}^* \in \Gamma_{ij}$*. And for any* $t_{ij}^* \in \Gamma_{ij}$*,* $X_k^{ij} = \{0, 1\}, \ \forall k \neq j$*.*

Proof: Since $\tilde{V}_{ij}^{UB}(\tilde{\lambda})$ is linear in X_j^{ij} and there is no constraint associated with it, for any given t_{ij}, $X_j^{ij} = \{0, 1\}$. If $\frac{\tilde{r}_{ij}}{t_{ij}} - \tilde{\lambda}_j^{ij} \leq 0$ then $X_j^{ij} = 0$ and $X_j^{ij} = 1$ otherwise. If $X_j^{ij} = 0$, it is easy to check that $t_{ij}^* = v_{i0} + v_{ij} + \sum_{k\neq j} v_{ik}$ is optimal. If $X_j^{ij} = 1$, since for any given t_{ij} $Z_{ij}^{rl}(\lambda)$ is a continuous knapsack, the optimal solution is obtained by rank-ordering with respect to $\frac{-\tilde{\lambda}_k^{ij}}{c_{ik}}, \forall k \neq j$. If $t_{ij}^* \in \Gamma_{ij}$ then $X_k^{ij}\{0, 1\}, \ \forall k \neq j$ and the proposition holds trivially. If $t_{ij}^* \notin \Gamma_{ij}$, then there exists $\tilde{k}(\neq j)$ such that $X_{\tilde{k}}^{ij}$ takes fractional value.

However, as we argue below that, if there exists $X^{ij}_{\tilde{k}} \in (0,1)$ for a given t^*_{ij}, then it is optimal to increase t^*_{ij} such that $t^*_{ij} \in \Gamma_{ij}$ and set $X^{ij}_{\tilde{k}} = 1$.

Let $\tilde{V}^{UB*}_{ij}(\tilde{\lambda})$ be the optimal value when $t^*_{ij} \notin \Gamma_{ij}$ and $X^{ij}_{\tilde{k}} \in (0,1)$. Let $\tilde{V}^{UB^d}_{ij}(\tilde{\lambda})$ be the optimal value when $t_{ij} = t^*_{ij} - c_{i\tilde{k}} X^{ij}_{\tilde{k}}$ in which case $X^{ij}_{\tilde{k}} = 0$, and $\tilde{V}^{UB^u}_{ij}(\tilde{\lambda})$ be the optimal value when $t_{ij} = t^*_{ij} + c_{i\tilde{k}}(1 - X^{ij}_{\tilde{k}})$ in which case $X^{ij}_{\tilde{k}} = 1$. That is, $\tilde{V}^{UB^d}_{ij}(\tilde{\lambda})$ and $\tilde{V}^{UB^u}_{ij}(\tilde{\lambda})$ are the optimal values obtained by changing t_{ij} such that $X^{ij}_{\tilde{k}}$ is rounded down or rounded up, respectively.

If $X^{ij}_{\tilde{k}} \in (0,1)$ is optimal, then $\tilde{V}^{UB*}_{ij}(\tilde{\lambda}) \geq \tilde{V}^{UB^d}_{ij}(\tilde{\lambda})$. Comparing the terms, this holds if:

$$\frac{-\tilde{\lambda}^{ij}_{\tilde{k}}}{v_{i\tilde{k}}} \geq \frac{\tilde{r}_{ij}}{t^*_{ij}(t^*_{ij} - v_{i\tilde{k}} X^{ij}_{\tilde{k}})}. \tag{2.5.10}$$

Next, comparing $\tilde{V}^{UB^u}_{ij}(\tilde{\lambda})$ $\tilde{V}^{UB*}_{ij}(\tilde{\lambda})$, we observe that $\tilde{V}^{UB^u}_{ij}(\tilde{\lambda}) \geq \tilde{V}^{UB*}_{ij}(\tilde{\lambda})$ if:

$$\frac{-\tilde{\lambda}^{ij}_{\tilde{k}}}{v_{i\tilde{k}}} \geq \frac{\tilde{r}_{ij}}{t^*_{ij}(t^*_{ij} + v_{i\tilde{k}}(1 - X^{ij}_{\tilde{k}}))}. \tag{2.5.11}$$

If (2.5.10) holds, then (2.5.11) is trivially true. Which implies that for given t_{ij} if setting $X^{ij}_{\tilde{k}}$ to a fractional value gives higher objective value than setting to 0, then it is optimal to increase t_{ij} such that $t_{ij} \in \Gamma_{ij}$ and $X^{ij}_{\tilde{k}} = 1$ □.

We use the above result to show that the reformulation based relaxation provides tighter upper bound than Feldman and Topaloglu (2015). Consider the continuous knapsack in (2.5.4c) for a given i and interval g. Let $\lambda = \bar{\lambda}$ be the optimal Lagrange multipliers:

$$V^{UB}_i(\bar{\lambda}, g) = \max_{X \in [0,1]^n} \left\{ \sum_{j=1}^{m} \left(\frac{\theta_i r_j v_{ij}}{t^g_i} - \bar{\lambda}^i_j \right) X^i_j \middle| v_i + \sum_{k=1}^{m} v_{ik} \cdot X^i_k \leq t^{g+1}_i \right\}, \tag{2.5.12}$$

We introduce auxiliary variables $X_k^{ij}, X_k^{i\phi}$ along with the constraint $X_k^{ij} = X_k^{i\phi}$. Since (2.5.12) is a linear program, there exists optimal dual variables $\hat{\lambda}_k^{ij}$ associated with this constraint such that (2.5.12) can we rewritten as:

$$\begin{aligned} V_i^{UB}(\bar{\lambda}, g) = & \max_{X\in[0,1]^n} \left\{ \sum_{j=1}^{m} \left(\frac{\theta_i r_j v_{ij}}{t_i^g} - \bar{\lambda}_j^i \right) X_j^{ij} - \sum_{j,k} \hat{\lambda}_k^{ij} (X_k^{ij} - X_k^{i\phi}) \right. \\ & \left. \middle| v_i + \sum_{k=1}^{m} v_{ik} \cdot X_k^{ij} \le t_i^{g+1}, \forall j \in \{1, ..., m\} \right\}, \end{aligned} \tag{2.5.13a}$$

$$\begin{aligned} = & \sum_{j=1}^{m} \max_{X\in[0,1]^n} \left\{ \left(\frac{\theta_i r_j v_{ij}}{t_i^g} - \bar{\lambda}_j^i \right) X_j^{ij} - \sum_{k} \hat{\lambda}_k^{ij} (X_k^{ij} - X_k^{i\phi}) \right. \\ & \left. \middle| v_i + \sum_{k=1}^{m} v_{ik} \cdot X_k^{ij} \le t_i^{g+1} \right\}, \end{aligned} \tag{2.5.13b}$$

$$\begin{aligned} = & \sum_{j=1}^{m} \max_{X\in[0,1]^n} \left\{ \left(\frac{\theta_i r_j v_{ij}}{t_i^g} - \bar{\lambda}_j^i - \hat{\lambda}_j^{ij} \right) X_j^{ij} - \sum_{k\neq j} \hat{\lambda}_k^{ij} X_k^{ij} \right. \\ & \left. \middle| v_i + \sum_{k=1}^{m} v_{ik} \cdot X_k^{ij} \le t_i^{g+1} \right\}. \end{aligned} \tag{2.5.13c}$$

where (2.5.13b) follows from the separability of objective function and (2.5.13c) follows from dual feasibility which requires $\sum_j \hat{\lambda}_k^{ij} = 0, \forall i, k$.

Let $V_{ij}^{UB}(\bar{\lambda}, g)$ be the inner optimization problem.

$$V_{ij}^{UB}(\bar{\lambda}, g) = \max_{X\in[0,1]^n} \left\{ \left(\frac{\theta_i r_j v_{ij}}{t_i^g} - \bar{\lambda}_j^i - \hat{\lambda}_j^{ij} \right) X_j^{ij} - \sum_{k\neq j} \hat{\lambda}_k^{ij} X_k^{ij} \middle| v_i + \sum_{k=1}^{m} v_{ik} \cdot X_k^{ij} \le t_i^{g+1} \right\}. \tag{2.5.14}$$

We now show that for any given interval, the reformulation based relaxation provides tighter upper bound than the Feldman and Topaloglu (2015) bound.

Proposition 2.5.2. *The reformulation based relaxation provides tighter upper bound i.e.,* $\tilde{V}^{UB} \le V^{UB}$.

Proof: Let $\tilde{\lambda}_j^{ij} = \bar{\lambda}_j^i + \hat{\lambda}_j^{ij}$ and $\tilde{\lambda}_j^{ij} = \hat{\lambda}_k^{ij}$. Note that the assignment satisfies dual feasibility constraint $\sum_{ij} \tilde{\lambda}_k^{ij} = 0, \forall k$. Consider the reformulation based relaxation given in (2.5.9) computed at $\tilde{\lambda}$ and interval g:

$$\tilde{V}_{ij}^{UB}(\tilde{\lambda}) = \max_{t_{ij}\in[t_i^g, t_i^{g+1}]} \max_{X_k^{ij}\in[0,1]} \left\{ \left(\frac{\theta_i r_j v_{ij}}{t_{ij}} - \tilde{\lambda}_j^{ij}\right) X_j^{ij} - \sum_{k\neq j} \tilde{\lambda}_k^{ij} X_k^{ij} \middle| v_{i0} + v_{ij} + \sum_{k\neq j}^{m} v_{ik} \cdot X_k^{ij} \leq t_{ij} \right\}, \tag{2.5.15a}$$

$$= \max_{t_{ij}\in[t_i^g, t_i^{g+1}]} \max_{X_k^{ij}\in[0,1]} \left\{ \left(\frac{\theta_i r_j v_{ij}}{t_{ij}} - \bar{\lambda}_j^{i} - \hat{\lambda}_j^{ij}\right) X_j^{ij} - \sum_{k\neq j} \hat{\lambda}_k^{ij} X_k^{ij} \middle| v_{i0} + v_{ij} + \sum_{k\neq j}^{m} v_{ik} \cdot X_k^{ij} \leq t_{ij} \right\}, \tag{2.5.15b}$$

$$= \max_{t_{ij}\in[t_i^g, t_i^{g+1}]} \max_{X_k^{ij}\in[0,1]} \left\{ \left(\frac{\theta_i r_j v_{ij}}{t_{ij}} - \bar{\lambda}_j^{i} - \hat{\lambda}_j^{ij}\right) X_j^{ij} - \sum_{k\neq j} \hat{\lambda}_k^{ij} X_k^{ij} \middle| v_{i0} + \sum_{k}^{m} v_{ik} \cdot X_k^{ij} \leq t_{ij} - v_{ij}(1 - X_{ij}) \right\}, \tag{2.5.15c}$$

$$\leq \max_{X_k^{ij}\in[0,1]} \left\{ \left(\frac{\theta_i r_j v_{ij}}{t_i^g} - \bar{\lambda}_j^{i} - \hat{\lambda}_j^{ij}\right) X_j^{ij} - \sum_{k\neq j} \hat{\lambda}_k^{ij} X_k^{ij} \middle| v_{i0} + \sum_{k}^{m} v_{ik} \cdot X_k^{ij} \leq t_i^{g+1} \right\}, \tag{2.5.15d}$$

$$= V_{ij}^{UB}(\bar{\lambda}, g). \tag{2.5.15e}$$

Improving Expectation Bounds

Assortment Packing Problem.

The assortment packing problem (APP) consists of a retailer with K products that are introduced over a finite horizon H. Each product can be introduced in at most one of the time periods and remains in the assortment thereafter. Customers have an initial preference value v_k for product k which decreases to $\kappa_{k,t-t_k} \cdot v_k$ in period t if product k is introduced in t_k, where $\kappa_{k,t-t_k} \in [0,1]$. The customers are assumed to follow the multinomial logit choice function. The retailer's objective is to offer an assortment at each time step to maximize its expected revenue. Let X_{kt} be a binary variable which is 1 of product k is introduced in time step t and 0 otherwise, r_k be the retailer's revenue for product k and β_t be the discount factor associated with time step t. The platforms expected revenue, given X_{kt} and the optimal revenue is given by:

$$V(X) = \sum_{t=1}^{H} \beta_t \sum_{k=1}^{K} r_k \left(\frac{v_k \sum_{u=1}^{t} \kappa_{k,t-u} X_{ku}}{v_0 + \sum_{l=1}^{K} v_l \sum_{u=1}^{t} \kappa_{l,t-u} X_{lu}} \right), \tag{2.5.16a}$$

$$APP: \qquad V^* = \max_{X_{kt} \in \{0,1\}} \left\{ V(X) \middle| \sum_{t=1}^{H} X_{kt} \leq 1, \forall k \in \{1, ..., K\} \right\}. \tag{2.5.16b}$$

Caro et al. (2014) show the problem is NP-hard and develop a randomized heuristic by solving the continuous relaxation of (2.5.16b). Let $\tilde{X}_{kt}$ be the solution to the continuous relaxation, and Q_k be a random variable associated with product k such that $P(Q_k = t) = \tilde{X}_{kt}$. Let $f(Q_1, ..., Q_k)$ be the objective value associated with the $Q = (Q_1, ..., Q_K)$:

$$f(Q) = \sum_{t=1}^{H} \frac{\beta_t \cdot \sum_{k=1}^{K} \sum_{u=1}^{t} r_k v_k \kappa_{k,t-u} \mathbb{1}_{ku}}{v_0 + \sum_{l=1}^{K} \sum_{u=1}^{t} v_l \alpha_{l,t-u} \mathbb{1}_{lu}}.$$

Caro et al. (2014) provide the following parametric bound for the randomized rounding solution.

Theorem 2.5.1. *Let* $\rho = \frac{v_0}{v_0 + \max_{k=\{1,...,K\}} v_k}$. *Then,* $\mathbb{E}[f(Q)] \geq \rho \cdot V^*$ *(Caro et al., 2014).*

It immediately follows that APP is a special case of ZOFP-CS. The RCFP-CS reformulation of the APP is given by:

$$V^* = \max_{X_{jt} \in \{0,1\}} \left\{ \sum_{t=1}^{H} \beta_t \sum_{j=1}^{K} r_j \left(\frac{v_j \sum_{u=1}^{t} \kappa_{j,t-u} X_{ju}}{v_0 + \sum_{l=1}^{K} v_l \sum_{u=1}^{t} \kappa_{l,t-u} X_{lu}} \right) \middle| \sum_{t=1}^{H} X_{jt} \leq 1, \forall j \in \{1, ..., K\} \right\}, \tag{2.5.17a}$$

$$= \max_{X_{jt} \in \{0,1\}} \left\{ \sum_{t=1}^{H} \sum_{j=1}^{K} \sum_{u=1}^{t} \frac{\beta_t r_j \kappa_{j,t-u} X_{ju}}{v_0 + \sum_{l=1}^{K} v_l \sum_{u=1}^{t} \kappa_{l,t-u} X_{lu}} \middle| \sum_{t=1}^{H} X_{jt} \leq 1, \forall j \in \{1, ..., K\} \right\}, \tag{2.5.17b}$$

$$= \max_{X_{jt} \in \{0,1\}} \left\{ \sum_{t=1}^{H} \sum_{j=1}^{K} \sum_{u=1}^{t} \frac{\beta_t r_j \kappa_{j,t-u} X_{ju}}{v_0 + \sum_{u=1}^{t} v_j \kappa_{j,t-u} + \sum_{l=1, l \neq j}^{K} v_l \sum_{u=1}^{t} \kappa_{l,t-u} X_{lu}} \middle| \sum_{t=1}^{H} X_{jt} \leq 1, \forall j \in \{1, ..., K\} \right\}. \tag{2.5.17c}$$

Corollary 2.5.1.

- *The continuous relaxation of (2.5.17c) gives integral solution.*
- *Let* $\tilde{X}_{kt}^d$ *be the integral solution obtained using algorithm 1 with* $\tilde{X}_{kt}$ *as the input, then* $V(\tilde{X}^d) \geq \rho \cdot V^*$.

Thus, the reformulation improves the expectation bound in Caro et al. (2014) in two ways. First, it provides an equivalent continuous fractional program (2.5.17c) that gives integral solutions. Second, it provides a greedy rounding scheme for the fractional solution, which is ρ-optimal.

Two-sided Assortment Optimization:

The two-sided assortment optimization consists of a platform with n_s suppliers and m_c consumers. Each supplier i has a preference value v_{ij} associated with consumer j and a preference for an outside option v_{i0}. Similarly, the preference values for consumer j are defined as u_{ji} and u_{j0}. On offering an assortment of consumers (suppliers), each supplier (consumer), simultaneously and independently, selects at most one consumer (supplier) with probability given by the multinomial logit function. The platform receives a revenue r_{ij} if i and j select each other, and its goal is to maximize its expected revenue. Let X_{ij} be a binary variable that takes value one if i and j are offered to each other and zero otherwise. The platform's expected revenue associated with X_{ij} and the optimal expected revenue is given by:

$$V(X) = \sum_{i=1}^{n_s} \sum_{j=1}^{m_c} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m} v_{ik} \cdot X_{ik}} \right) \cdot \left(\frac{u_{ji} \cdot X_{ij}}{u_{j0} + \sum_{l=1}^{n} u_{jl} \cdot X_{lj}} \right), \tag{2.5.18a}$$

$$V^* = \max_X V(X). \tag{2.5.18b}$$

Ahmed et al. (2022) prove (2.5.18b) is NP-hard and provide a *one-sided* relaxation (OSR) by setting the outside option of consumers to 0, i.e., $u_{j0} = 0, \forall j$. They show that OSR reduces to:

$$\tilde{Z} = \max_{X_{ij} \in \{0,1\}} \left\{ \sum_{i=1}^{n_s} \sum_{j=1}^{m_c} r_{ij} \left(\frac{v_{ij} \cdot X_{ij}}{v_{i0} + \sum_{k=1}^{m_c} v_{ik} \cdot X_{ik}} \right) \middle| \sum_{i=1}^{n_s} X_{ij} \leq 1, \forall j \in \{1, ..., m_c\} \right\}, \tag{2.5.19a}$$

$$= \max_{X_{ij}\in\{0,1\}} \left\{ \sum_{i=1}^{n_s}\sum_{j=1}^{m_c} r_{ij}\left(\frac{v_{ij}\cdot X_{ij}}{v_{i0}+v_{ij}+\sum_{k=1,k\neq j}^{m_c} v_{ik}\cdot X_{ik}}\right) \middle| \sum_{i=1}^{n_s} X_{ij}\leq 1, \forall j\in\{1,...,m_c\}\right\}. \tag{2.5.19b}$$

Ahmed et al. (2022) consider the continuous relaxation of (2.5.19b) and provide parametric bounds in expectation. Let $\tilde{X}$ be the relaxed solution of (2.5.19b) and $\tilde{X}^b$ be the integral solution obtained using randomized rounding. The following bound holds:

Theorem 2.5.2. *Let $\rho = \min_{i,j}\{\frac{u_{ji}}{u_{j0}}\}$. Then $\mathbb{E}[V(\tilde{X}^b)] \geq \frac{\rho}{1+\rho}\cdot V^*$ (Ahmed et al., 2022).*

However, from theorem 2.4.1, the continuous relaxation of (2.5.19b) gives an integral solution. Therefore, the bound in theorem 2.5.2 is, in fact, exact.

Corollary 2.5.2. *$\tilde{X}$ is integral and $V(\tilde{X}) \geq \frac{\rho}{1+\rho}\cdot V^*$*

Improved Local Maxima.

We now provide an example to demonstrate that solving the continuous reformulation can improve the discrete local maxima.

Example 2.5.0.1. *Consider the following ZOFP-CS:*

$$Z = \max_{y\in\{0,1\}^3}\left\{\frac{5y_1}{1+y_1+y_2+y_3}+\frac{3y_2}{1+2y_1+3y_2+y_3}+\frac{3y_3}{1+y_1+y_2+3y_3}\middle| y_1+y_2+y_3\leq 2\right\}. \tag{2.5.20}$$

Observe that $y = [0,1,1]$ is a discrete local maxima of (2.5.20) with $Z = 1.2$. Now, consider the continuous reformulation

$$Z = \max_{y\in[0,1]^3}\left\{\frac{5y_1}{2+y_2+y_3}+\frac{3y_2}{4+2y_1+y_3}+\frac{3y_3}{4+y_1+y_2}\middle| y_1+y_2+y_3\leq 2\right\}. \tag{2.5.21}$$

Checking the KKT conditions, it follows that $y = [0, 1, 1]$ is no longer a local optimum of (2.5.21). Using $y = [0, 1, 1]$ as a starting solution, solving the continuous reformulation using any nonlinear continuous solver leads to the solution $y = [1, 0, 0]$ with $Z = 2.5$.

2.6 Numerical Experiments

We now investigate the practical utility of the reformulation in improving discrete local maxima by considering two problem sets. For the smaller problem set we explore $p \in \{5, 10\}$, $n \in \{20, 35\}$, $m \in \{2, 5\}$ and the right-hand side of the constraints $M_j \in \{2, 4\}$. For larger instances, we explore $p \in \{75, 150\}$, $n \in \{300, 600\}$, $m \in \{10, 25\}$ and $M_j \in \{2, 4\}$. Across all simulations, we sample c_{ij} from a uniform distribution $\mathcal{U}[0, 100]$ and $b_i \in \{200, 1000\}$. For a_{ij}, we consider two cases; in the base case we choose $a_{ij} \sim \mathcal{U}[0, 100]$ and for the case with *outliers*, we set $a_{ij} \sim \mathcal{U}[0, 100]$ with probability 0.9 and $a_{ij} \sim \mathcal{U}[0, 1000]$ with probability 0.1. Thus, we have 64 combinations in the parameter space, and for each combination, we report the average performance across ten instances.

For the MILP formulation, we use CPLEX 12.9.0 and set the cutoff time to 600 seconds. For RCFP-CS, we use the open-source sequential least squares programming algorithm (SLSQP) available within the python library *numpy* along with *numba* compiler for objective value computation. SLSQP is a quasi-Newton method that computes local maxima for non-concave functions. We, therefore, report the optimality gaps for the reformulation on smaller instances where the MILP returns the optimal solution within a threshold time. The simulations were run on Intel core i7, 16 GB RAM, Ubuntu 18.04.5 LTS using Python 3.6.9.

Recall that Z^m is the optimal MILP objective, $\mathbf{y}^{ls}$ is the discrete local search heuristic solution and Z^{ls} is objective value at $\mathbf{y}^{ls}$, and Z^r is the optimal (possibly local) RCFP-CS objective. We use the same random feasible solution as input for both Z^{ls} and Z^r. Let Z^r_{ls} be the RCFP-CS objective with $\mathbf{y}^{ls}$ as the initial solution. For the smaller problem sets, we report the following metrics:

- $\Delta_{or} = 100 \times \frac{Z^m - Z^r}{Z^m}$, the percentage optimality gap for the RCFP-CS solution starting from a random initial solution.
- $\Delta_{ol} = 100 \times \frac{Z^m - Z^{ls}}{Z^m}$, the percentage optimality gap for the discrete local search heuristic.
- $\Delta_{rl} = 100 \times \frac{Z^r_{ls} - Z^{ls}}{Z^r_{ls}}$, the percentage increase in the local search heuristic solution by using RCFP-CS reformulation with y_{ls} as the initial solution. Thus, Δ_{rl} measures the improvement the reformulation offers over the discrete local optima.

For the larger problem set, we observe that the RCFP-CS solution Z^r is higher than the MILP solution Z^m computed within the threshold time across all instances. We, therefore, report the percentage difference between them $\Delta_{ro} = 100 \times \frac{Z^r - Z^m}{Z^r}$ along with the runtime τ for the RCFP-CS solution.

Table 2.1 provides the results for the smaller problem set. We observe that the RCFP-CS solution, starting from a random initial feasible solution, provides close to the optimal solution, with $\Delta_{or} \leq 5.0\%$ across all parameter settings. The optimality gap is significantly higher for the local search heuristic, as seen in the column Δ_{ol}, especially for $M_j = 2$, with the average Δ_{ol} increasing from 10% for $M_j = 4$ to 18% for $M_j = 2$ without outliers and from 10% to 30% with outliers, indicating that as the constraints become tighter, the local search heuristic moves further away from the optimal solution. On the other hand,

Problem	Without Outliers						With Outliers					
	$M_j = 4$			$M_j = 2$			$M_j = 4$			$M_j = 2$		
(b, m, n, p)	Δ_{or}	Δ_{ol}	Δ_{rl}	Δ_{or}	Δ_{ol}	Δ_{rl}	Δ_{or}	Δ_{ol}	Δ_{rl}	Δ_{or}	Δ_{ol}	Δ_{rl}
(200, 2, 20, 5)	0.90	12.89	11.18	2.16	29.72	27.98	0.30	0.16	0.00	0.17	44.80	44.73
(200, 2, 20, 10)	1.49	14.40	12.86	4.55	25.01	21.23	0.30	1.54	1.07	0.15	25.79	25.59
(200, 2, 35, 5)	2.01	20.57	18.52	1.23	25.62	24.62	0.88	0.06	0.00	0.68	36.03	35.57
(200, 2, 35, 10)	1.37	21.11	19.73	3.90	21.33	18.15	1.07	7.54	6.67	1.86	30.93	29.57
(200, 5, 20, 5)	0.72	0.01	0.00	1.41	1.57	0.25	0.52	0.04	0.00	0.29	4.15	4.10
(200, 5, 20, 10)	0.65	0.00	0.00	1.75	7.28	5.53	0.49	0.00	0.00	0.28	6.25	6.07
(200, 5, 35, 5)	1.12	1.09	0.72	1.63	6.72	5.39	0.77	0.00	0.00	0.68	2.97	2.46
(200, 5, 35, 10)	0.48	0.92	0.72	2.31	9.46	6.99	0.51	0.01	0.00	0.71	9.24	8.87
Average	1.09	8.87	7.97	2.37	15.84	13.77	0.61	1.17	0.97	0.60	20.02	19.62
(1000, 2, 20, 5)	0.91	17.97	17.10	0.90	28.31	27.62	0.07	31.36	31.31	0.10	54.26	54.22
(1000, 2, 20, 10)	0.57	14.67	14.15	0.10	20.75	20.67	0.11	26.43	26.34	0.21	43.11	42.91
(1000, 2, 35, 5)	0.46	27.69	27.36	0.95	30.53	29.84	0.18	46.50	46.39	0.06	60.51	60.50
(1000, 2, 35, 10)	0.62	18.36	17.84	0.28	26.18	25.97	0.16	41.69	41.59	0.24	44.00	43.89
(1000, 5, 20, 5)	0.16	0.00	0.00	0.59	14.89	14.38	0.14	0.00	0.00	0.06	22.44	22.34
(1000, 5, 20, 10)	0.02	0.00	0.00	0.59	9.65	9.13	0.10	0.00	0.00	0.07	17.76	17.67
(1000, 5, 35, 5)	0.33	10.72	10.21	0.56	17.60	17.07	0.06	5.23	5.09	0.21	42.39	42.27
(1000, 5, 35, 10)	0.23	7.94	7.71	0.29	17.81	17.59	0.10	3.17	3.10	0.23	35.97	35.71
Average	0.41	12.17	11.80	0.53	20.71	20.28	0.12	19.30	19.23	0.15	40.06	39.94
Average	0.75	10.52	9.88	1.45	18.28	17.02	0.36	10.23	10.10	0.38	30.04	29.78

TABLE 2.1: Small problem set.

for the unconstrained case $(m = 5, n = 20, M_j = 4)$, Δ_{ol} is close to 0 for any value of b_i, with and without outliers. This substantiates the numerical results observed in Bront et al. (2009), where the local search heuristic provides close to the optimal solution for the unconstrained case. The optimality gap also increases with b_i as the number of variables set to one increases with b_i, and therefore the constraints are more likely to be tighter. This is analogous to what is observed in assortment optimization literature, where the assortment size increases with the preference for the outside option. The optimality gap

Problem	Without Outliers						With Outliers					
	$M_j = 4$			$M_j = 2$			$M_j = 4$			$M_j = 2$		
(b, m, n, p)	Δ_{ro}	τ	Δ_{rl}	Δ_{ro}	τ	Δ_{rl}	Δ_{ro}	τ	Δ_{rl}	Δ_{ro}	τ	Δ_{rl}
(200, 10, 300, 75)	11.43	2.48	0.69	8.19	1.32	12.00	14.86	1.49	0.09	8.44	0.84	1.96
(200, 10, 300, 150)	9.39	1.83	4.11	9.09	1.09	8.11	19.08	1.16	0.00	19.33	0.70	4.90
(200, 10, 600, 75)	15.88	15.95	0.64	11.37	9.29	14.63	31.90	8.81	0.04	13.31	6.07	1.15
(200, 10, 600, 150)	10.61	13.13	6.28	10.70	6.91	11.25	23.75	7.82	0.02	25.30	4.78	5.09
(200, 25, 300, 75)	8.48	4.08	0.00	10.86	3.12	0.74	18.24	3.21	0.00	18.03	1.95	0.03
(200, 25, 300, 150)	6.85	3.96	0.06	8.24	2.65	1.56	16.34	2.70	0.00	19.61	1.63	0.27
(200, 25, 600, 75)	12.66	28.94	0.02	14.56	19.21	0.28	26.58	21.07	0.00	30.24	11.50	0.08
(200, 25, 600, 150)	9.52	24.66	0.02	9.93	14.45	1.62	18.77	17.42	0.00	22.05	10.72	0.12
Average	10.60	11.88	1.48	10.37	7.25	6.27	21.19	7.96	0.02	19.54	4.77	1.70
(1000, 10, 300, 75)	7.93	2.87	11.30	2.38	1.92	12.33	6.84	1.67	22.66	3.00	1.11	30.91
(1000, 10, 300, 150)	8.32	2.34	8.27	8.64	1.68	8.86	15.96	1.49	18.03	21.01	0.96	21.03
(1000, 10, 600, 75)	11.57	18.99	13.20	12.08	12.76	14.59	24.86	10.50	26.40	15.58	7.61	33.93
(1000, 10, 600, 150)	9.26	14.72	9.89	8.67	10.72	10.22	22.92	10.28	21.95	23.11	6.13	25.09
(1000, 25, 300, 75)	8.23	4.82	6.90	10.03	3.51	10.22	15.57	3.40	1.46	8.66	2.30	16.96
(1000, 25, 300, 150)	6.10	4.17	5.61	7.17	3.04	7.35	13.35	3.12	3.50	16.23	1.91	16.21
(1000, 25, 600, 75)	11.03	30.42	10.14	11.83	22.33	12.95	24.08	19.84	1.67	22.29	13.99	19.93
(1000, 25, 600, 150)	8.54	26.73	8.21	9.39	18.27	9.37	17.71	17.18	3.53	21.43	11.99	21.69
Average	8.87	13.13	9.19	8.77	9.28	10.73	17.66	8.44	12.40	16.42	5.75	23.22
Average	9.74	12.51	5.34	9.57	8.27	8.50	19.43	8.20	6.21	17.98	5.26	12.46

TABLE 2.2: Large problem set.

increases as n increases, indicating the performance of the local search heuristic decreases with an increase in problem dimensionality. However, the reformulation significantly improves the local search heuristic across all parameter settings, as seen in the column Δ_{rl}. In fact, it again computes close to the optimal value as we observe that the percentage increase it offers (Δ_{rl}) almost coincides with the optimality gap (Δ_{ol}).

Table 2.2 provides the results for the larger problem set. The reformulation is quite scalable and takes less than 30 seconds on average across all parameters. And as seen in column Δ_{ro}, it also provides significant improvements over the solution computed by the MILP within the cutoff time. Further, the performance gains are higher with outliers, indicating the MILP computes a better solution within a given time without the outliers; Δ_{ro} increases from 9.7% with outliers to 19.4% without outliers for $M_j = 4$ and from 9.5% to 17.9 for $M_j = 2$. With respect to the local search heuristic, we observe similar trends as before, where RCFP-CS provides higher performance gains when the constraints are tighter; Δ_{rl} increases from 5.3% for $M_j = 4$ to 8.5% for $M_j = 2$ without outliers and from 6.2% to 12.46% with outliers. Similarly, the performance gains increase with b_i; for example, for $M_j = 2$, Δ_{rl} increases from 1.7% for $b_i = 200$ to 23.2% for $b_i = 1000$, with outliers.

2.7 Conclusions

In this work, we consider the class of 0-1 linear fractional programs under cardinality-type constraints and provide a simple reformulation to solve it as a continuous linear fractional program. We show that the direct relaxation solution can be rounded off with parametric guarantees. We obtain tighter parametric bounds for a class of assortment optimization problems and illustrate that the reformulation can improve the commonly used local search heuristic by exploiting the continuous solution space. We believe our work has broader implications for solving large discrete fractional programs as it facilitates the application of tools and algorithms developed in the continuous optimization literature to a discrete problem.

This work can be extended further in multiple directions. First, since the reformulation provides a greedy procedure to obtain a discrete solution starting from any feasible fractional solution, one can investigate the utility of the reformulation within a branch-and-bound based algorithm to scale up the MILP formulation. Second, one can explore more general constraints under which the reformulation gives integral solutions. Finally, one can examine the reformulation in nonlinear settings for which the traditional MILP formulation is likely to be even more intractable than the linear setup.

Chapter 3

Capacity Pooling for Network Revenue Management

3.1 Introduction

Network Revenue Management (NRM) involves the strategic use of limited resources to fulfill the demand for products that rely on one or more of these resources. It has been widely studied and applied in industries such as airlines, hotels, and car rentals. Two common control methods used in NRM are virtual nesting control and bid-price control.

In virtual nesting control, products are assigned to virtual buckets based on their potential revenue and demand. Incoming requests are matched against these virtual nests, allowing for the dynamic sharing of resources among products. This method was pioneered by American Airlines in 1983 to leverage the network structure of airline itineraries and address the limitations of rigid partitioned controls.

Bid-price control, on the other hand, assigns a *bid price* to each resource, representing its marginal value. A demand request is accepted if the revenue from fulfilling the request exceeds the total bid prices of all the resources consumed by the product.

In this study, we consider a novel control policy used within Indian Railways (IR). IR is one of the largest public sector enterprises in India and ranks as the fourth-largest railway network globally, serving nearly 8 billion passengers during the 2018-19 fiscal year and generating gross earnings of approximately 51,000 crore INR (Ministry of Railways, 2019). Given that IR is a public sector enterprise, it has operational constraints that limit the usage of virtual nesting or bid-price control strategies. Each passenger train in IR offers multiple classes that are physically distinct, with prices for each origin-destination pair varying by class. These prices are regulated and do not change dynamically, except for a few express trains. As such, this setting aligns more closely with the single-fare, multi-leg scenario discussed by Ciancimino et al. (1999). As Ciancimino et al. (1999) argue, the nested controls commonly used in airlines are not directly applicable in these settings. Additionally, IR cannot reject an incoming itinerary request if it has the capacity to fulfill the demand. This operational constraint limits the use of bid-price control strategies, which rely on dynamic accept-reject decisions to protect resources.

In response to these challenges, IR has developed a unique control policy based on quota allocation to optimize revenue. This paper focuses on two primary types: partitioned quotas and pooled quotas. The partitioned quota allocates a fixed number of seats to a specific origin-destination (O-D) pair, and any demand for that O-D pair is initially met from these partitioned resources. To manage excess demand and reduce the risk of unsold inventory, IR also maintains a pooled quota, which is utilized once the partitioned seats for a given O-D pair are exhausted. The allocation of seats between partitioned and pooled quotas depends on pricing and anticipated demand.

We consider the above quota allocation problem, model it as a dynamic program, and discuss the conditions under which simple only partitioning can be optimal. While an only-partitioning strategy is easy to compute, a hybrid allocation strategy with partitioned and pooled capacities is computationally difficult due to the large state space of the dynamic program. To address this, we develop a new Lagrangian relaxation-based solution wherein we decompose the problem by resource type. We then discuss the well-known deterministic linear programming solution for the partitioned and pooled setting, and show that it provides an upper bound to our problem. We additionally show that the Lagrange relaxation solution of Topaloglu (2009) for the standard network revenue management problem is also a valid upper bound. We compare our relaxation against the two upper-bound solutions on a real-world data set and find that our approach provides a tighter upper bound on the optimal revenue. We also report the corresponding revenues and observe that the our solution offers significant improvements over various benchmark strategies.

The rest of the paper is organized as follows. In §3.2, we discuss related literature. In §3.3, we introduce dynamic programming model and highlight the tradeoff between partitioned and pooled capacities along with the special cases when the only-partitioning strategy is optimal. In §3.4, we provide the Lagrange relaxation-based solution. Finally, in §3.5, we perform numerical analysis and conclude in §3.6.

3.2 Literature Review

Our work is closely related to the capacity control literature in network revenue management. We first discuss the common capacity controls studied in network revenue management and then discuss the literature specific to railways.

The two broad classes of capacity controls commonly used in network revenue management are virtual nesting control and bid-price controls.

American Airlines is credited with developing the virtual nesting control approach to incorporate the network structure into their allocation decisions (Smith et al., 1992). Each product is assigned to a virtual nest using an *indexing* step, and incoming requests are fulfilled against these virtual nests. Bertsimas and De Boer (2005) developed discrete simulation-based methods to improve the robustness of virtual nesting controls with respect to demand uncertainty. Van Ryzin and Vulcano (2008) developed a continuous simulation-based method that addresses some of the limitations of the discrete approach.

Williamson (1992) was one of the earliest works to study bid-price control in network settings and apply it to airline seat allocation. Talluri and Van Ryzin (1998) showed the asymptotic optimality of bid prices and laid the foundation for understanding the interplay between fare classes and resource allocation. However, computing optimal bid-price control strategy is computationally difficult. Therefore, various approximation solutions have been developed. Topaloglu (2009) developed a Lagrange relaxation-based solution where the problem is decomposed into single-resource problems, and approximate bid prices are computed using this decomposition. They show the relaxation provides a tighter upper bound than the well-known deterministic linear programming-based (DLP) upper bound. Other approximations include the affine value-function approximation by Adelman (2007) and, more recently, the product-based approximation by Zhang et al. (2022).

The above control strategies have been more widely studied and applied in airline settings than railways. One reason, Ciancimino et al. (1999) highlight, is that these controls were largely developed in United States (US) after deregulation of airline industry in 1970s and air transport is a more common mode of transport in US than railways.

Ciancimino et al. (1999) is credited as one of the earliest works to use mathematical optimization models to address seat allocation problem in passenger rails. Seat allocation problem involves determining the number of seats to be sold to each origin-destination pair. They argue that the nested fare classes of airlines does not immediately carry over to railway settings, as railways usually carry physically distinct classes for distinct fares. They, therefore, consider a single-fare, multi-leg setting and provide a deterministic linear and a stochastic non-linear model. They benchmark their strategies on real-world dataset and show the non-linear model outperforms the first-come-first-serve solution as well the linear model. Since then, various extensions have been proposed. You (2008) extend the non-linear model to a two-fare setting - a full fare and a discounted fare segment. Jiang et al. (2015) consider dynamic seat allocation using short-term demand forecasting. Yan et al. (2020) extend it to flexible train capacity.

Gopalakrishnan and Rangaraj (2010), Dutta and Ghosh (2012) develop deterministic linear programming based solution for Indian Railways settings. In our works, instead of planning against mean demand, we explicitly account for dynamic arrival rates. We model the novel control policy used in IR as a dynamic program and use a Lagrange relaxation approach, similar to (Topaloglu, 2009), to address the curse of dimensionality. Numerically, we observe our relaxation to be tighter than both DLP and Topaloglu (2009).

3.3 Model

Network revenue management (NRM) is characterized by a set of m resources (or legs) denoted by L and a set of n products denoted by K. We assume that each product consumes at most one unit of each resource and that there is initial capacity $\mathbf{c} = \langle c_l \rangle$ of resources available. For each product $k \in K$, we define $\mathbf{a}_k = \langle a_{lk} \rangle$ as a vector of size m, such that $a_{lk} = 1$ if product k consumes resource l and 0 otherwise. Let $\mathbf{A} = [\mathbf{a}_1, ...\mathbf{a}_n]$ be the $m \times n$ resource consumption matrix. We define $L_k = \{l : a_{lk} = 1, \forall l \in L\}$ as the set of resources consumed by product k and $K_l = \{k : a_{lk} = 1, \forall k \in K\}$ as the set of products using resource l.

We assume demand arrives sequentially with at most one request at each time step. We denote the probability of a request arriving at time step t by λ_t and, conditional on an arrival, it is for product k with probability $\lambda_{k,t}$ ($\sum_k \lambda_{k,t} = 1$). On fulfilling the demand for product k, the platform obtains a revenue of r_k. The objective of the platform is to maximize its total expected revenue over the finite planning horizon H by optimally partitioning its resources between partitioned and pooled quotas.

While capacity partitioning has been widely studied in NRM, our setting involves a novel mechanism whereby the platform carries a subset of its resources as pooled capacity, which it uses to serve demand across products along with dedicated capacities for each resource. The platform uses a first-come, first-serve (FCFS) policy, and a request for product k is accepted if the product's dedicated capacity or the pooled resources are available. Additionally, it uses the pooled resources to fulfill a request for product k only after the dedicated capacity for k is fully utilized. Thus, the control variable is characterized by the booking limits assigned to each product along with a pooled capacity at the start of the planning period.

Let $\mathbf{x} = \langle x_k \rangle,\ \forall k \in K$ be the vector of partitioned capacities for the products and let $\mathbf{y} = \langle y_k \rangle, \forall k \in L$ be the vector of pooled resources. Note that the partitioned capacity is defined with respect to products, and the pooled capacity is defined with respect to resources. We define the state space as $\mathbf{s} = \langle \mathbf{x}, \mathbf{y} \rangle$. Let $\mathbb{1}_k(\mathbf{x})$ be an indicator function for product k which is 1 if $x_k \geq 1$ and 0 otherwise. Let $\mathbb{1}_k(\mathbf{y})$ be an indicator function for product k which is 1 if $y_l \geq a_{lk}$ for all $l \in L_k$ i.e. the functions $\mathbb{1}_k(\cdot)$ indicates whether or not there are sufficient resources to fulfill a request for product k.

Let $V^t(s)$ be the value function for state $\mathbf{s}$ at time step t. Let $\mathbf{e}_k$ be a vector of length n which is 1 at index k and 0 otherwise. The dynamic programming formulation as is given as follows:

$$V^t(\mathbf{s}) = (1 - \lambda_t) \cdot [0 + V^{t+1}(\mathbf{s})] + \sum_k \lambda_t \cdot \lambda_{k,t} \Bigg[\mathbb{1}_k(\mathbf{x}) \cdot [r_k + V^{t+1}(\mathbf{x} - \mathbf{e}_k, \mathbf{y})] + \\ [1 - \mathbb{1}_k(\mathbf{x})] \cdot \mathbb{1}_k(\mathbf{y}) \cdot [r_k + V^{t+1}(\mathbf{x}, \mathbf{y} - \mathbf{a}_k)] + [1 - \mathbb{1}_k(\mathbf{x})] \cdot [1 - \mathbb{1}_k(\mathbf{y})] \cdot [0 + V^{t+1}(\mathbf{s})] \Bigg]. \tag{3.3.1}$$

If there is no arrival at time step t, the system remains in the same state. If a request for product k arrives, and the partitioned capacity is available ($x_k > 0$) then the platform fulfills the demand and obtains the reward r_k, and the system transitions to $x_k - 1$. If there is no partitioned capacity but pooled capacity is available ($\mathbb{1}_k(\mathbf{y}) = 1$), the platform fulfills the demand using the pooled capacity, and the pooled capacity is consumed across all legs L_k. If neither partitioned nor pooled capacity is available, the incoming request is declined, and the system remains in the same state.

The objective of the platform is to maximize the total expected value given by $V_0(s^0)$ such that the following capacity constraint is satisfied at time step 0:

$$\mathbf{A} \cdot \mathbf{x}^0 + \mathbf{y}^0 \leq \mathbf{c}.$$

The constraint ensures that the sum of pooled and partitioned resources on any given leg is not greater than the total capacity available on that leg.

We now illustrate the model along with the primary trade-off between partitioned and pooled capacities with the following examples.

Example 1: Consider a train travelling from station s_1 s_3 via s_2 and 1 seat for each leg. Denote product (s_1, s_2) with 1, product (s_1, s_3) with 2 and (s_2, s_3) with 3. Similarly, denote resource or leg (s_1, s_2) with 1 and (s_2, s_3) with 2. Assume $H = 3$, $r_1 = r_3 = 10$, $r_2 = 15$, $\lambda_t = 1, \forall t$ and $\lambda_{2,1} = 1$, $\lambda_{1,2} = 1$ and $\lambda_{3,3} = 1$ i.e., there are a deterministic arrivals with product 2 request coming first followed by product 1 and 3.

Consider the case when the seat is assigned to pooled capacity on each leg at time step 0. Product 2 request arrives at the first time step and is fulfilled using the pooled capacity. The subsequent requests for products 1 and 3 are declined since no seats are left. The resultant value function is given by $V^1(0, 0, 0; 1, 1) = 15$ Now, consider the case when the seat is assigned to partitioned capacity for products 1 and 3. Product 2 request is declined, and product 1 and 3 requests that arrive at subsequent time steps are fulfilled. The resultant value function is given by $V^1(0, 0, 0; 1, 1) = 20$. Thus, partitioned capacities help protect the resources for products that might arrive later.

Example 2: Consider the same setting as before, but instead of deterministic arrivals, assume the arrivals are equally likely $\lambda_{j,t} = \frac{1}{3}$.

If the seat is assigned to pool capacity, the value function is given by $V^3(0,0,0;0,0) = 0$, $V^3(0,0,0;1,0) = \frac{10}{3}$, $V^3(0,0,0;0,1) = \frac{10}{3}, V^3(0,0,0;0,1) = \frac{45}{3}$ for $t = 3$, $V^2(0,0,0;0,0) = 0$, $V^2(0,0,0;1,0) = \frac{50}{9}$, $V^2(0,0,0;0,1) = \frac{50}{9}, V^2(0,0,0;1,1) = \frac{125}{9}$ for $t = 2$, and $V^1(0,0,0;1,1) = \frac{415}{27}$ for $t = 1$. The optimal partitioned capacity is to set $x_1 = x_3 = 1$ and the associated value function at $t = 1$ $v^1(1,0,1;0,0) = \frac{380}{27}$. Pooling performs better because partitioning the resource prevents it from being used for other product requests.

The above examples highlight the trade-off between the partitioning and pooling: partitioned capacity helps protect resources for future higher revenue arrivals. However, it might lead to unsold inventory. Pooling, on the other hand, reduces unsold inventory; however, it fails to protect resources.

Some of the observations in the above examples can be generalized.

Lemma 3.1.

- *If* $\lambda_{k,t} \in \{0,1\}, \forall k \in K, \ \forall t$, *then only-partitioning is optimal.*
- *If the horizon is sufficiently long, (* $H > \max_{l \in L} \max_{t:\lambda_t \lambda_{kt} > 0} \{\frac{c_l}{\lambda_t \lambda_{kt}}\}$*), then only-partitioning is optimal.*

Proof: We use the deterministic linear programming formulation (see section 3.4), along with the observation that in our setting, with single-fare multi-leg products, the resource consumption matrix $\mathbf{A}$ is totally-unimodular (Ciancimino et al. (1999)). We provide additional details in the appendix 3.7. $\square$.

We now show that only-partitioning can be computed efficiently. Define $v_k^t(x_j)$ as follows:

$$v_k^t(x_k) = (1 - \lambda_t \lambda_{k,t}) \cdot v_k^{t+1}(x_k) + \lambda_t \lambda_{k,t} \Big[\mathbb{1}_k(x_k)[r_k + v_k^{t+1}(x_k - 1)] + (1 - \mathbb{1}_k(x_k)) v_k^{t+1}(x_k)] \Big].$$

Proposition 3.3.1. *If $y_l = 0, \forall l \in L$, then $V^t(\mathbf{x},\mathbf{y}) = \sum_k v_k^t(x_k)$.*

Proof: We first argue it holds for the last time step, $t = H$.

$$V^H(\mathbf{x},\mathbf{0}) = \sum_k \lambda_t \lambda_{k,t} [\mathbb{1}_k(\mathbf{x})(r_k)], \tag{3.3.2a}$$

$$= \sum_k v_k^H(x_k). \tag{3.3.2b}$$

Assume it holds for time step $t+1$. Then, for any time step t:

$$V^t(\mathbf{x},\mathbf{0}) = (1-\lambda_t)V^{t+1}(\mathbf{x},\mathbf{0}) + \sum_k \lambda_t \lambda_{k,t} \Big[\mathbb{1}_k(\mathbf{x})[r_k + V^{t+1}(\mathbf{x}-\mathbf{e}_k,\mathbf{0})] + (1-\mathbb{1}_k(\mathbf{x}))V^{t+1}(\mathbf{x},\mathbf{0}) \Big], \tag{3.3.3a}$$

$$= \begin{array}{l} (1-\lambda_t)\sum_k v_k^{t+1}(x_k) + \sum_k \lambda_t \lambda_{k,t} \left[\mathbb{1}_k(\mathbf{x})[r_k + \sum_{j \neq k} v_j^{t+1}(x_j) + v_k^{t+1}(x_k - 1)] \right. \\ \qquad \left. + (1-\mathbb{1}_k(\mathbf{x}))(\sum_j v_j^{t+1}(x_j)) \right], \end{array} \tag{3.3.3b}$$

$$= \begin{array}{l} (1-\lambda_t)\sum_k v_k^{t+1}(x_k) + \sum_k \lambda_t \lambda_{k,t} \left[\mathbb{1}_k(\mathbf{x})[r_k + v_k^{t+1}(x_k - 1)] \right. \\ \qquad \left. + (1-\mathbb{1}_k(\mathbf{x}))v_k^{t+1}(x_k) + \sum_{j \neq k} v_j^{t+1}(x_j) \right], \end{array} \tag{3.3.3c}$$

$$= \begin{array}{l} (1-\lambda_t)\sum_k v_k^{t+1}(x_k) + \sum_k \lambda_t \lambda_{k,t} \left[\mathbb{1}_k(\mathbf{x})[r_k + v_j^{t+1}(x_k - 1)] \right. \\ \qquad \left. + (1-\mathbb{1}_k(\mathbf{x}))v_k^{t+1}(x_k) + \sum_j v_j^{t+1}(x_j) - v_k^{t+1}(x_k) \right], \end{array} \tag{3.3.3d}$$

$$= \begin{aligned} &(1-\lambda_t)\sum_k v_k^{t+1}(x_k) + \sum_k \lambda_t\lambda_{k,t}\left[\mathbb{1}_k(\mathbf{x})[r_k + v_k^{t+1}(x_k-1)] + (1-\mathbb{1}_k(\mathbf{x}))v_j^{t+1}(x_k)\right] \\ &\qquad + \sum_j \lambda_t v_j^{t+1}(x_j)\sum_k \lambda_{k,t} - \sum_k \lambda_t\lambda_{k,t}v_k^{t+1}(x_k) \end{aligned}, \tag{3.3.3e}$$

$$= \begin{aligned} &\sum_k v_k^{t+1}(x_k) - \sum_k \lambda_t\lambda_{k,t}v_k^{t+1}(x_k) + \\ &\sum_k \lambda_t\lambda_{k,t}\left[\mathbb{1}_k(\mathbf{x})[r_k + v_k^{t+1}(x_k-1)] + (1-\mathbb{1}_k(\mathbf{x}))v_k^{t+1}(x_k)\right] \end{aligned}, \tag{3.3.3f}$$

$$= \sum_k (1-\lambda_t\lambda_{k,t})v_k^{t+1}(x_k) + \sum_k \lambda_t\lambda_{k,t}\left[\mathbb{1}_k(\mathbf{x})[r_k + v_k^{t+1}(x_k-1)] + (1-\mathbb{1}_k(\mathbf{x}))v_k^{t+1}(x_k)\right], \tag{3.3.3g}$$

$$= \sum_j (1-\lambda_t\lambda_{j,t})v_j^{t+1}(x_j) + \sum_j \lambda_t\lambda_{j,t}\left[\mathbb{1}_k(x_k)[r_k + v_k^{t+1}(x_k-1)] + (1-\mathbb{1}_k(x_k))v_k^{t+1}(x_k)\right], \tag{3.3.3h}$$

$$= \sum_j v_k^t(x_k). \tag{3.3.3i}$$

Thus, the value function becomes separable across products. And, since the value function for a single product $v_k(x_k)$ is known to be concave in x_k (Gallego et al. (2019)), the optimal only-partitioning allocation can be computed efficiently.

3.4 Upper Bounds

We now present three upper-bound solutions. The first is the Lagrange relaxation-based upper bound where we reformulate and relax a subset of constraints, and decompose the problem across partitioned and pooled resources. The second is deterministic linear programming based upper bound where we compute the allocations based on the mean demand. Lastly, we show that the Lagrange relaxation of the standard network revenue management problem provides a valid upper bound.

3.4.1 Lagrange Relaxation

Lagrange relaxation-based methods decouple large state space problems into smaller, separable sub-problems by relaxing the right set of constraints. The resultant sub-problem usually have smaller dimensions and can be solved efficiently. The complexity of the resultant sub-problem is determined by which constraints are relaxed. Here, we present a relaxation that separates the problem across pooled and partitioned resources.

We reformulate the model by defining decision variable $u_{lk}^{ty} \in \{0,1\}$ which is 1 if at time step t demand for product k is fulfilled using pooled capacity on leg l. We coordinate the decisions across legs for each product by introducing the constraint $u_{lk}^{ty} = u_{\phi k}^{ty}$. The value function can be rewritten as follows:

$$V^t(\mathbf{s}) = (1-\lambda_t)V^{t+1}(\mathbf{s}) + \sum_k \lambda_t \cdot \lambda_{k,t} \left[r_k[\mathbb{1}_k(\mathbf{x}) + u_{\phi k}^{ty}] + V^{t+1}(\mathbf{x} - \mathbb{1}_k(\mathbf{x}) \cdot \mathbf{e}_k^x, \mathbf{y} - \sum_l a_{lk} u_{lk}^{ty} \cdot \mathbf{e}_l^y), \right], \tag{3.4.1}$$

subject to the constraints:

$$a_{lk} u_{lk}^{ty} \leq \mathbb{1}(y_l), \quad \forall k, l, \tag{3.4.2a}$$

$$a_{lk} u_{lk}^{ty} = a_{lk} u_{\phi k}^{ty}, \quad \forall k, l, \tag{3.4.2b}$$

$$u_{\phi k}^{ty} \leq 1 - \mathbb{1}(x_k) \ \forall k, \tag{3.4.2c}$$

$$u^{ty}_{\phi k} \geq 1 - \mathbb{1}(x_k) - \sum_l a_{lk}(1 - \mathbb{1}(y_l)) \ \forall k. \tag{3.4.2d}$$

Constraint (3.4.2a) ensures the pooled capacity is used on leg l only if its available, Constraint (3.4.2b) coordinates the decision for pooled capacities across legs, Constraint (3.4.2c) ensures pooled capacity is utilized after the partitioned capacity is exhausted and constraint (3.4.2d) forces the pooled capacity to be used if it is available and the partitioned capacity has been exhausted.

Let $\beta^1 = \{\beta^{1t}_{lk}, \forall t, \forall l \in L, \forall k \in K\}$, $\beta^2 = \{\beta^{2t}_{\phi k} \geq 0, \forall t, \forall k \in K\}$ be the Lagrange multipliers associated with constraints (3.4.2b), (3.4.2d), respectively. Let $\beta = \{\beta^1, \beta^2\}$. Relaxing the constraints, we obtain the following separable value function:

Proposition 3.4.1. *Given* β*:*

$$V^{t,\beta}(\mathbf{x}, \mathbf{y}) = -\sum_{t' \geq t} \sum_k \lambda_{t'} \lambda_{t'k} \beta^{2t'}_{\phi k} + \sum_k v^{t,\beta}_k(x_k) + \sum_l v^{t,\beta}_l(y_l),$$

$$v^{t,\beta}_k(x_k) = (1 - \lambda_t \lambda_{k,t}) v^{t+1,\beta}_k(x_k) + \lambda_t \lambda_{k,t} \Big\{ \mathbb{1}(x_k)[r_k + \beta^{2t}_{\phi k}] + u^{ty}_{\phi k}[r_k + \beta^{2t}_{\phi k} - \sum_l a_{lk} \beta^{1t}_{lk}] + v^{t+1,\beta}_k(x_k - u^{tx}_{\phi k}) \Big\}, \tag{3.4.3}$$

$$v^{t,\beta}_l(y_l) = (1 - \lambda_t) v^{t+1,\beta}_l(y_l) + \sum_k \lambda_t \lambda_{k,t} \Big\{ \mathbb{1}(y_l) a_{lk} u^{ty}_{lk} \beta^{1t}_{lk} + a_{lk}(1 - \mathbb{1}(y_l)) \beta^{2t}_{\phi k} + v^{t+1,\beta}_l(y_l - a_{lk} u^{ty}_{lk}) \Big\}. \tag{3.4.4}$$

Proof: We prove by induction. It is easy to check that it holds for the last time step, $t = H$. Assume it holds at $t + 1$. Then, at time step t:

$$V^t(\mathbf{x}, \mathbf{y}) = \begin{array}{l} (1-\lambda_t)\Big[-\sum_{t' \geq t+1} \sum_k \lambda_{t'} \lambda_{t'k} \beta^{2t'}_{\phi k} + \sum_k v^{t+1,\beta}_k(x_k) + \sum_l v^{t+1,\beta}_l(y_l) \Big] \\ \quad + \sum_k \lambda_t \lambda_{kt} \Big[r_k[\mathbb{1}(x_k) + u^{ty}_{\phi k}] - \sum_{t' \geq t+1} \sum_{k'} \lambda_{t'} \lambda_{t'k'} \beta^{2t'}_{\phi k'} \\ \quad + v^{t+1,\beta}_k(x_k - \mathbb{1}(x_k)) + \sum_{k' \neq k} v^{t+1,\beta}_{k'}(x_{k'}) + \sum_l v^{t+1,\beta}_l(y_l - a_{lk} u^{ty}_{lk}) \Big] \\ + \sum_{l,k} \lambda_t \lambda_{kt} \beta^{1t}_{lk} a_{lk} (u^{ty}_{lk} - u^{ty}_{\phi k}) + \sum_k \lambda_t \lambda_{kt} \beta^{2t}_{\phi k} \Big[u^{ty}_{\phi k} - (1 - \mathbb{1}(x_k)) + \sum_l a_{lk}(1 - \mathbb{1}(y_l)) \Big] \end{array}, \tag{3.4.5a}$$

$$= \begin{array}{l} -\sum_{t' \geq t} \sum_k \lambda_{t'} \lambda_{t'k} \beta^{2t'}_{\phi k} + (1-\lambda_t) \sum_k v^{t+1,\beta}_k(x_k) + \sum_k \lambda_t \lambda_{kt} \Big[\mathbb{1}(x_k)[r_k + \beta^{2t}_{\phi k}] + \\ \quad u^{ty}_{\phi k}[r_k + \beta^{2t}_{\phi k} - \sum_l a_{lk} \beta^{1t}_{lk}] + v^{t+1,\beta}_k(x_k - \mathbb{1}(x_k)) + \sum_{k' \neq k} v^{t+1}_{k'}(x_{k'}) \Big] \\ \sum_l \Big[(1-\lambda_t) v^{t+1,\beta}_l(y_l) + \sum_k \lambda_t \lambda_{k,t} \Big\{ \mathbb{1}(y_l) a_{lk} u^{ty}_{lk} \beta^{1t}_{lk} + a_{lk}(1 - \mathbb{1}(y_l)) \beta^{2t}_{\phi k} + \\ \qquad v^{t+1,\beta}_l(y_l - a_{lk} u^{ty}_{lk}) \Big\} \Big] \end{array}, \tag{3.4.5b}$$

$$= \begin{array}{l} -\sum_{t' \geq t} \sum_k \lambda_{t'} \lambda_{t'k} \beta^{2t'}_{\phi k} + \sum_k v^{t+1,\beta}_k(x_k)[1 - \lambda_t + \sum_{k' \neq k} \lambda_t \lambda_{k't}] \\ + \sum_k \lambda_t \lambda_{kt} \Big[\mathbb{1}(x_k)[r_k + \beta^{2t}_{\phi k}] + u^{ty}_{\phi k}[r_k + \beta^{2t}_{\phi k} - \sum_l a_{lk} \beta^{1t}_{lk}] + v^{t+1,\beta}_k(x_k - \mathbb{1}(x_k)) \Big] \\ \sum_l \Big[(1-\lambda_t) v^{t+1,\beta}_l(y_l) + \sum_k \lambda_t \lambda_{k,t} \Big\{ \mathbb{1}(y_l) a_{lk} u^{ty}_{lk} \beta^{1t}_{lk} + a_{lk}(1 - \mathbb{1}(y_l)) \beta^{2t}_{\phi k} + \\ \qquad v^{t+1,\beta}_l(y_l - a_{lk} u^{ty}_{lk}) \Big\} \Big] \end{array}, \tag{3.4.5c}$$

$$
\begin{aligned}
& - \sum_{t' \geq t} \sum_k \lambda_{t'} \lambda_{t'k} \beta_{\phi k}^{2t'} + \sum_k v_k^{t+1,\beta}(x_k)[1 - \lambda_t + \lambda_t(1 - \lambda_{kt})] \\
= \; & + \sum_k \lambda_t \lambda_{kt} \left[\mathbb{1}(x_k)[r_k + \beta_{\phi k}^{2t}] + u_{\phi k}^{ty}[r_k + \beta_{\phi k}^{2t} - \sum_l a_{lk} \beta_{lk}^{1t}] + v_k^{t+1,\beta}(x_k - \mathbb{1}(x_k)) \right], \\
& \sum_l \left[(1 - \lambda_t) v_l^{t+1,\beta}(y_l) + \sum_k \lambda_t \lambda_{k,t} \Big\{ \mathbb{1}(y_l) a_{lk} u_{lk}^{ty} \beta_{lk}^{1t} + a_{lk}(1 - \mathbb{1}(y_l)) \beta_{\phi k}^{2t} + \right. \\
& \left. \qquad v_l^{t+1,\beta}(y_l - a_{lk} u_{lk}^{ty}) \Big\} \right]
\end{aligned} \tag{3.4.5d}
$$

$$
= - \sum_{t' \geq t} \sum_k \lambda_{t'} \lambda_{t'k} \beta_{\phi k}^{2t'} + \sum_k v_k^{t,\beta}(x_k) + \sum_l v_l^{t,\beta}(y_l). \square. \tag{3.4.5e}
$$

The above relaxation provides a separable value function which is a valid upper bound. To obtain tight upper bound, we minimize with respect to the Lagrange multipliers.

$$
V^0(\mathbf{x}^*, \mathbf{y}^*) \leq \min_\beta \max_{\mathbf{x},\mathbf{y}} V^{0,\beta}(\mathbf{x}, \mathbf{y}). \tag{3.4.6}
$$

However, to compute the optimal quotas w.r.t to the above relaxed objective, we still need to solve an allocation problem at $t = 0$. We do so by solving a linear program. Let $c_k^m = \min_{l \in L_k} c_l$ and $\gamma = \{\gamma_k^p : \gamma_k^p \in \{0, 1\}, \forall k \in K, p \in \{0, ..., c_k^m\}\}$ i.e., γ_k^p is a binary variable, which is 1 if product k is assigned a partitioned capacity p. Similarly, let $\theta = \{\theta_l^q : \theta_l^q \in \{0, 1\}, \forall l \in L, q \in \{0, ..., c_l\}\}$. We then write $x_k = \sum_{p=0}^{c_k^m} \gamma_k^p \cdot p$ such that $\sum_{p=0}^{c_k^m} \gamma_k^p = 1$, and $y_l = \sum_{q=0}^{c_l} \theta_l^q \cdot q$ such that $\sum_{q=0}^{c_l} \theta_l^q = 1$. Let ξ, ξ^c be defined as follows:

$$
\xi = \left\{ \begin{aligned} & (\gamma, \theta) : \sum_{p=0}^{c_k^m} \gamma_k^p = 1, \gamma_k^p \in \{0, 1\}, \forall k \in K, \sum_{q=0}^{c_l} \theta_l^q = 1, \theta_l^q \in \{0, 1\}, \\ & \qquad \sum_k a_{lk} (\sum_{p=0}^{c_k^m} \gamma_k^p \cdot p) + \sum_{q=0}^{c_l} \theta_l^q \cdot q \leq c_l, \forall l \in L \end{aligned} \right\} \tag{3.4.7}
$$

$$\xi^c = \left\{ \begin{array}{r} (\gamma,\theta) : \sum_{p=0}^{c_k^m} \gamma_k^p = 1, \gamma_k^p \in [0,1], \forall k \in K, \sum_{q=0}^{c_l} \theta_l^q = 1, \theta_l^q \in [0,1], \\ \sum_k a_{lk} (\sum_{p=0}^{c_k^m} \gamma_k^p \cdot p) + \sum_{q=0}^{c_l} \theta_l^q \cdot q \le c_l, \forall l \in L \end{array} \right\} \tag{3.4.8}$$

The inequality (3.4.6) can be written as:

$$V^0(\mathbf{x}^*, \mathbf{y}^*) \le \min_{\beta} \max_{\mathbf{x},\mathbf{y}} V^{0,\beta}(\mathbf{x}, \mathbf{y}) \tag{3.4.9a}$$

$$= \min_{\beta} \max_{\mathbf{x},\mathbf{y}} \left\{ -\sum_{t' \ge 0} \sum_k \lambda_{t'} \lambda_{t'k} \beta_{\phi k}^{2t'} + \sum_k v_k^{0,\beta}(x_k) + \sum_l v_l^{0,\beta}(y_l) \right\}, \tag{3.4.9b}$$

$$= \min_{\beta} \left\{ -\sum_{t' \ge 0} \sum_k \lambda_{t'} \lambda_{t'k} \beta_{\phi k}^{2t'} + \max_{\mathbf{x},\mathbf{y}: A\mathbf{x}+\mathbf{y} \le \mathbf{c}} \left\{ \sum_k v_k^{0,\beta}(x_k) + \sum_l v_l^{0,\beta}(\hat{y}) \right\} \right\}, \tag{3.4.9c}$$

$$= \begin{array}{l} \min_{\beta} \Bigg[-\sum_{t' \ge 0} \sum_k \lambda_{t'} \lambda_{t'k} \beta_{\phi k}^{2t'} + \\ \max_{\gamma,\theta \in \xi} \left\{ \sum_k \sum_{p=0}^{c_k^m} \gamma_k^p v_k^{0,\beta}(x_k = p) + \sum_l \sum_{q=0}^{c_l} \theta_l^q v_l^{0,\beta}(y_l = q) \right\} \Bigg] \end{array}, \tag{3.4.9d}$$

$$\le \begin{array}{l} \min_{\beta} \Bigg[-\sum_{t' \ge 0} \sum_k \lambda_{t'} \lambda_{t'k} \beta_{\phi k}^{2t'} + \\ \max_{\gamma,\theta \in \xi^c} \left\{ \sum_k \sum_{p=0}^{c_k^m} \gamma_k^p v_k^{0,\beta}(x_k = p) + \sum_l \sum_{q=0}^{c_l} \theta_l^q v_l^{0,\beta}(y_l = q) \right\} \Bigg] \end{array}. \tag{3.4.9e}$$

For a given β, we solve the above linear program and iteratively update the value of β using the sub-gradient descent method until convergence.

3.4.2 Deterministic Linear Program

The standard deterministic linear program (DLP-S) involves computing the number of request w_k^d that will be accepted against product k subject to capacity and mean demand constraints:

$$DLP\text{-}S = \max_{w_k^d} \quad \sum_k r_k w_k^d, \tag{3.4.10a}$$

$$\text{subject to} \quad \sum_k a_{ik} w_k^d \leq c, \quad \forall i \in L, \tag{3.4.10b}$$

$$w_k^d \leq \sum_t \lambda_t \lambda_{kt}, \quad \forall k \in K. \tag{3.4.10c}$$

We define deterministic linear programming with pooling as follows. Let x_k^d be the partitioned capacity, let y^d be the pooled capacity. Let z_k^d be the number of request accepted for product k against the pooled capacity. The DLP with pooling is given by:

$$DLP = \max_{x_k^d, y^d, z_k^d} \quad \sum_k r_k (x_k^d + z_k^d), \tag{3.4.11a}$$

$$\text{subject to} \quad \sum_k a_{ik} x_k^d + y^d \leq c, \quad \forall i \in L, \tag{3.4.11b}$$

$$\sum_k a_{ik} z_k^d \leq y^d, \quad \forall i \in L, \tag{3.4.11c}$$

$$x_k^d + z_k^d \leq \sum_t \lambda_t \lambda_{kt}, \quad \forall k \in K, \tag{3.4.11d}$$

$$0 \leq x_k^d, z_k^d \leq c. \tag{3.4.11e}$$

We observe that the DLP-P reduces to the standard deterministic linear program.

Lemma 3.2. *DLP with pooling is equivalent to the standard DLP.*

Proof: Given an optimal solution $\hat{w}_k^d$ to DLP-S, a feasible solution to the DLP with pooling can be obtained by setting $y^d_= 0$, $z_k^d = 0 \ \forall k \in K$ and $x_k^d = \hat{w}_k^d$. Similarly, given an optimal solution $\hat{x}_k^d, \hat{y}^d, \hat{z}_k^d$ to the DLP with pooling, a feasible solution to DLP-S can be obtained by $w_k^d = \hat{x}_k^d + \hat{z}_k^d$. $\square$.

3.4.3 Standard NRM Lagrange Relaxation

We now show that the Lagrange relaxation for the standard NRM problem by Topaloglu (2009) provides a valid upper bound. In the previous LR relaxation, we relax the constraint (3.4.2b) as well as the constraint (3.4.2d). Instead, we initially retain the constraint (3.4.2b) and set $\beta_{\phi k}^{2t} = 0$ i.e., we essentially drop the constraint (3.4.2d). Then the following result holds:

Lemma 3.3. *If* $\beta_{\phi k}^{2t} = 0, \forall t, \forall k \in K$*, at optimality* $\mathbf{x} = 0$.

Proof: We prove by induction. For some product k, let $x_k = 1$ and $x_j = 0, \ \forall j \neq k$. Consider the value functions $V^{t,\beta^2=0}(x_k = 1, \mathbf{x}_{-k}, \mathbf{y})$ and the value function obtained by reassigning x_k to the pooled capacity: $V^{t,\beta^2=0}(x_k = 0, \mathbf{x}_{-k}, \mathbf{y}+a_k)$. We drop $\mathbf{x}_{-k}$ terms for notational convenience as they are set to 0. For the reassigned state, we set $u_{\phi k}^{ty} = 1$ i.e., the request against the partitioned resource that is reassigned to the pooled capacity is always accepted. At the last time step, it is easy to check that $V^{H,\beta^2=0}(x_k = 0, \mathbf{y} + a_k) \geq$

$V^{H,\beta^2=0}(x_k = 1, \mathbf{y})$. Assume it is true for time step $t + 1$. Then at time t:

$$V^{t,\beta^2=0}(x_k = 1, \mathbf{y}) = \begin{array}{l} (1-\lambda_t)V^{t+1,\beta^2=0}(x_k = 1, \mathbf{y}) + \lambda_t \cdot \lambda_{k,t}\Big[r_k + V^{t+1,\beta^2=0}(x_k = 0, \mathbf{y})\Big] \\ + \sum_{j \neq k} \lambda_t \cdot \lambda_{j,t} \left[r_j[\mathbb{1}_j(\mathbf{x}) + u^{ty}_{\phi j}] + V^{t+1,\beta^2=0}(x_k = 1, \mathbf{y} - \sum_l u^{ty}_{lj} \cdot \mathbf{e}^y_l) \right] \end{array}. \tag{3.4.12a}$$

$$V^{t,\beta^2=0}(x_k = 0, \mathbf{y} + a_k) = \begin{array}{l} (1-\lambda_t)V^{t+1,\beta^2=0}(x_k = 0, \mathbf{y} + a_k) + \lambda_t \cdot \lambda_{k,t}\Big[r_k + V^{t+1,\beta^2=0}(x_k = 0, \mathbf{y})\Big] \\ + \sum_{j \neq k} \lambda_t \cdot \lambda_{j,t} \left[r_j[\mathbb{1}_j(\mathbf{x}) + u^{ty}_{\phi j}] + V^{t+1,\beta^2=0}(x_k = 0, \mathbf{y} + a_k - \sum_l u^{ty}_{lj} \cdot \mathbf{e}^y_l) \right] \end{array}. \tag{3.4.12b}$$

Comparing term-wise the result follows. □

Given that the optimal partitioned capacity is 0, the resultant value function under only-pooling is given by:

$$V^{t,\beta^2=0}(\mathbf{x} = \mathbf{0}, \mathbf{y}) = (1-\lambda_t)V^{t+1,\beta^2=0}(\mathbf{x} = \mathbf{0}, \mathbf{y}) + \sum_k \lambda_t \cdot \lambda_{k,t} \left[r_k u^{ty}_{\phi k} + V^{t+1,\beta^2=0}(\mathbf{x} = 0, \mathbf{y} - \sum_l u^{ty}_{lk} \cdot \mathbf{e}^y_l) \right], \tag{3.4.13}$$

subject to the constraints:

$$u^{ty}_{lk} \leq \mathbb{1}(y_l), \quad \forall k, l, \tag{3.4.14a}$$

$$u^{ty}_{lk} = u^{ty}_{\phi k}, \quad \forall k, l. \tag{3.4.14b}$$

Let β^{1t}_{lk} be the Lagrange multipliers associated with constraint (3.4.14b). Then the following lemma holds:

Train	Class	Average Fare	Number of Products	Capacity	Bookings-to-Capacity ratio
1	1A	2398.0	33	65	1.13
1	2A	1364.4	35	361	1.25
1	3A	992.3	35	816	1.27
2	1A	3255.5	21	24	1.26
2	2A	1962.2	23	172	1.15
2	3A	1538.2	25	890	1.03

Table 3.1: IR passenger train data set.

Lemma 3.4. *Given β^{1t}_{lk} and $\beta^{2t}_{\phi k} = 0$, the value function satisfies:*

$$V^{t,\beta^1,\beta^2=0}(\mathbf{0},\mathbf{y}) = \sum_{t'\geq t}\sum_{k} \lambda_{t'}\lambda_{t'k}u^{t'y}_{\phi k}[r_k - \sum_{l}\beta^{1t'}_{lk}] + \sum_{l} v^{t,\beta^1}_{l}(y_l),$$

$$\textit{where } v^{t,\beta^1}_{l}(y_l) = (1-\lambda_t)v^{t+1,\beta^1}_{l}(y_l) + \sum_{k}\lambda_t\lambda_{k,t}\left\{u^{ty}_{lk}\beta^{1t}_{lk} + v^{t+1,\beta^1}_{l}(y_l - u^{ty}_{lk})\right\}.$$

Proof: The proof follows from Topaloglu (2009). □.

Thus, Topaloglu's relaxation (2009) is a valid upper bound for our problem.

Proposition 3.4.2.

$$V^0(\mathbf{x}^*,\mathbf{y}^*) \leq V^{t,\beta^2=0}(\mathbf{0},\mathbf{y}) \leq \min_{\beta^1} V^{t,\beta^1,\beta^2=0}(\mathbf{0},\mathbf{y}) \leq DLP.$$

Proof: The first two inequalities follow from the observation that we either drop or relax a subset of constraints. The last inequality follows from the result by Topaloglu (2009), who show that their relaxation is tighter than the DLP-S.

3.5 Numerical Experiments

We now evaluate the three upper bounds and their corresponding revenues. Additionally, we consider the heuristic allocation used within IR, which allocates 10% of the capacity to pooling, with the remaining capacity distributed among products based on their mean demand.

The dataset used is from two passenger trains: Train 1, which has eleven stations, and Train 2, which has eight stations. Each train offers three classes: First AC (1A), Second AC (2A), and Third AC (3A). Class 1A consists of premium seats sold at a higher price point than the other classes. Class 3A has the lowest prices among the three classes but is typically allocated the highest seat capacity due to higher demand.

We have data for 15 departures for each train. For each departure, we have aggregated bookings for each class and the allocated capacity for that class. We observe that IR sells more tickets than the available capacity to account for cancellations. We compute the average bookings-to-capacity ratio for each class, which we use to inform the various demand settings under which we evaluate the algorithms.

For each train, class, table 3.1 provides the number of products (unique origin-destination pairs), average fares, total capacity, and the average bookings-to-capacity ratio (the average is across the 15 departures and the products in that class). On average, the bookings-to-capacity ratio ranges between 1 and 1.3, indicating that demand generally exceeds capacity.

To simulate demand, we use a metric similar to bookings-to-capacity ratio from the literature called the *load factor* (Topaloglu (2009)) defined as follows:

$$\alpha = \frac{\sum_t \sum_k \sum_l a_{lk} \lambda_t \lambda_{kt}}{\sum_l c_l}.$$

We explore five load factors, $\alpha \in \{1.0, 1.1, 1.2, 1.3\}$, which approximately reflect the bookings-to-capacity ratios observed in the data. For the planning horizon, we consider $H \in \{50, 100, 200, 300\}$.

The parameter space is thus defined as follows: $Train \in \{1, 2\}$, $Class \in \{1A, 2A, 3A\}$, $\alpha \in \{1.0, 1.1, 1.2, 1.3\}$, and $H \in \{50, 100, 200, 300\}$, resulting in 96 problem instances. For arrival rates, we define weights $w_{k,t}, \forall k \in K, t \in \{1, ..., H\}$ as follows:

$$w_{k,t} = \begin{cases} \text{Uniform}[0, 100], & \text{with probability } 0.5 \\ 0.0, & \text{with probability } 0.5 \end{cases}$$

The arrival rates $\lambda_{k,t}$ are computed as $\lambda_{k,t} = \frac{w_{k,t}}{\sum_k w_{k,t}}$. Additionally, we set $\lambda_t = 1$, $\forall t$. The capacities are configured to satisfy the load factor α.

3.5.1 Upper bound

We denote the upper bound obtained using our relaxation as LR, the upper bound obtained using Topaloglu (2009) as TOP, and the upper bound obtained using deterministic linear programming as DLP. For each problem instance, we set the number of products and their rewards based on the IR dataset. We conduct six simulations for the arrival rate and report the runtime, as well as the percentage gaps between the upper-bound solutions.

- The percentage improvement in the upper bound offered by TOP over DLP: $\Delta_{TOP}^{DLP} = 100 \times \frac{DLP-TOP}{DLP}$.
- The percentage improvement in the upper bound offered by LR over DLP: $\Delta_{LR}^{DLP} = 100 \times \frac{DLP-LR}{DLP}$.
- The percentage improvement in the upper bound offered by LR over TOP: $\Delta_{LR}^{TOP} = 100 \times \frac{TOP-LR}{TOP}$.

Table 3.2 shows the performance results for Train 1 and Class 2A. We observe that our relaxation is scalable, requiring less than 600 seconds even for larger instances with $H = 300$. Although

Train	Class	α	Horizon	TOP (time)	LR (time)	Δ_{TOP}^{DLP}	Δ_{LR}^{DLP}	Δ_{LR}^{TOP}
1	2A	1.0	50	25.6 ± 3.8	198.8 ± 1.4	1.6 ± 0.0	4.0 ± 0.4	2.4 ± 0.3
1	2A	1.0	100	43.6 ± 0.1	249.2 ± 3.8	1.0 ± 0.1	2.6 ± 0.0	1.6 ± 0.1
1	2A	1.0	200	100.2 ± 1.8	402.5 ± 12.8	0.5 ± 0.0	1.7 ± 0.1	1.2 ± 0.0
1	2A	1.0	300	180.8 ± 7.6	549.3 ± 10.8	0.4 ± 0.0	1.3 ± 0.0	0.9 ± 0.0
1	2A	1.1	50	23.0 ± 0.1	188.7 ± 2.7	1.8 ± 0.1	4.5 ± 0.1	2.8 ± 0.0
1	2A	1.1	100	44.6 ± 1.4	248.4 ± 5.4	0.8 ± 0.1	2.4 ± 0.2	1.6 ± 0.1
1	2A	1.1	200	98.3 ± 1.1	373.1 ± 4.5	0.4 ± 0.0	1.5 ± 0.1	1.1 ± 0.1
1	2A	1.1	300	166.7 ± 1.7	512.3 ± 3.9	0.3 ± 0.0	1.3 ± 0.1	1.0 ± 0.1
1	2A	1.2	50	23.5 ± 0.1	187.3 ± 0.3	2.1 ± 0.2	4.7 ± 0.2	2.7 ± 0.1
1	2A	1.2	100	42.5 ± 0.1	231.2 ± 1.2	1.1 ± 0.1	2.6 ± 0.1	1.4 ± 0.1
1	2A	1.2	200	96.7 ± 1.0	365.6 ± 19.0	0.6 ± 0.0	1.6 ± 0.0	1.1 ± 0.0
1	2A	1.2	300	163.3 ± 0.5	481.3 ± 5.4	0.3 ± 0.1	1.3 ± 0.0	1.0 ± 0.0
1	2A	1.3	50	23.3 ± 0.1	184.1 ± 1.4	2.5 ± 0.1	5.0 ± 0.3	2.6 ± 0.2
1	2A	1.3	100	42.3 ± 0.3	228.2 ± 8.8	1.4 ± 0.3	2.7 ± 0.0	1.3 ± 0.4
1	2A	1.3	200	95.8 ± 2.5	331.8 ± 7.3	0.7 ± 0.0	1.7 ± 0.0	1.0 ± 0.0
1	2A	1.3	300	154.9 ± 2.5	454.5 ± 6.8	0.4 ± 0.1	1.3 ± 0.1	0.9 ± 0.0
Average						1.0	2.5	1.5

TABLE 3.2: Upper bound comparison: Train 1, Class 2A (mean ± standard deviation).

it takes longer than the TOP relaxation, it provides tighter upper bounds across all problem instances.

The performance gains of TOP and LR over DLP are more pronounced for smaller horizons. For longer horizons, DLP generally provides relatively good upper bounds, so the additional improvements we observe are smaller. On average, TOP offers a 1% improvement over DLP, while LR provides an additional 1.5% improvement over TOP.

3.5.2 Revenue

For revenues, we consider the following solutions:

- LR-Rev: The revenue obtained using a feasible allocation derived from the LR solution.
- DLP-Rev: The revenue obtained using a feasible allocation derived from the DLP solution. For both LR-Rev and DLP-Rev, we obtain a feasible allocation by rounding down any fractional solution.
- Pooling-Rev: Revenue obtained using only-pooling strategy.
- Partition-Rev: Revenue obtained using the only-partition solution. Recall that only-partitioning is computationally feasible and can be computed efficiently.
- IR-Heuristic-Rev: Revenue obtained by allocating 10% of capacity to pooling and assigning the remaining capacity to each product based on its mean demand. This is currently used a rule-of-thumb within IR.

We simulate each solution 100 times and compute the average revenue and standard deviations. We then report the percentage gap between the revenue obtained using the LR relaxation and the other solutions, while also highlighting whether the differences in average revenues are statistically significant at the 95% confidence level.

- The percentage improvement in revenue offered by LR over Partition-Rev, $\delta_{PA}^{LR} = 100 \times \frac{\text{LR-Rev}-\text{Partition-Rev}}{\text{LR-Rev}}$.
- The percentage improvement in revenue offered by LR over Pooling-Rev, $\delta_{PO}^{LR} = 100 \times \frac{\text{LR-Rev}-\text{Pooling-Rev}}{\text{LR-Rev}}$.
- The percentage improvement in revenue offered by LR over IR-Heuristic-Rev, $\delta_{IR}^{LR} = 100 \times \frac{\text{LR-Rev}-\text{IR-Heuristic-Rev}}{\text{LR-Rev}}$.
- The percentage improvement in revenue offered by LR over DLP-Rev, $\delta_{DLP}^{LR} = 100 \times \frac{\text{LR-Rev}-\text{DLP-Rev}}{\text{LR-Rev}}$.

Table 3.3 presents revenue comparisons for Train 1, Class 2A. On average, the Lagrange Relaxation (LR) solution consistently outperforms all other solutions across different horizons and load factors. Note that, in most problem instances, the improvements offered by LR are statistically significant at the 95% confidence level.

With respect to the planning horizon, as H increases (e.g.,$H = 300$), the performance gains from the LR solution diminish. This trend suggests that for longer horizons, or when the load factor is high, other solutions—such as Deterministic Linear Programming (DLP) or the Indian Railways heuristic—perform relatively well because the likelihood of unsold inventory decreases.

The results indicate that while the LR solution is particularly effective in scenarios with shorter horizons and moderate load factors, the advantages taper off as the problem becomes more relaxed (i.e., with more time or capacity available).

Overall, the LR solution demonstrates significant improvements across a wide range of conditions. This is likely because, unlike DLP, it accounts for varying arrival rates. Additionally, unlike the only-partition or only-pooling solutions, it balances the allocation between partitioned and pooled capacities, ensuring that the right amount of resources are protected while minimizing the likelihood of unsold inventory.

3.6 Conclusion

In this paper, we consider a novel resource allocation policy used by Indian Railways. We provide a dynamic programming-based formulation and discuss the underlying tradeoffs between resource partitioning and pooling. We develop scalable upper-bound solutions and numerically demonstrate that they are tighter than well-known upper-bound solutions. Additionally, we show the corresponding revenues obtained using our solution are higher than other benchmark solutions.

Train	Class	α	Horizon	δ^{LR}_{PA}	δ^{LR}_{PO}	δ^{LR}_{IR}	δ^{LR}_{DLP}
1	2A	1.00	50	16.98*	3.39*	35.57*	44.81*
1	2A	1.00	100	8.36*	2.11*	17.77*	23.4*
1	2A	1.00	200	4.04*	1.8*	10.13*	14.08*
1	2A	1.00	300	2.25*	0.78	6.08*	9.64*
1	2A	1.10	50	16.54*	3.7*	35.53*	40.6*
1	2A	1.10	100	6.98*	3.12*	16.92*	22.71*
1	2A	1.10	200	2.6*	1.43*	10.37*	14.26*
1	2A	1.10	300	0.51	1.43*	5.03*	8.42*
1	2A	1.20	50	13.19*	2.49*	35.29*	39.61*
1	2A	1.20	100	6.6*	2.86*	18.55*	24.38*
1	2A	1.20	200	2.13*	1.76*	9.25*	12.42*
1	2A	1.20	300	-0.40	1.7*	4.79*	8.05*
1	2A	1.30	50	12.93*	3.4*	35.61*	40.44*
1	2A	1.30	100	5.63*	2.6*	15.87*	21.6*
1	2A	1.30	200	0.57	1.68*	7.18*	10.37*
1	2A	1.30	300	-1.0*	2.08*	4.12*	6.76*
Average				6.2	2.3	16.8	21.4

TABLE 3.3: Revenue comparison: Train 1, Class 2A.

3.7 Appendix

3.7.1 Proofs

Lemma 3.1:

- If $\lambda_{k,t} \in \{0,1\}, \forall k \in K, \ \forall t$, then only-partitioning is optimal.
- If the horizon is sufficiently long, ($H > \max_{l \in L} \max_{t:\lambda_t \lambda_{kt} > 0} \{\frac{c_l}{\lambda_t \lambda_{kt}}\}$), then only-partitioning is optimal.

Proof: Consider the standard DLP:

$$DLP\text{-}S = \max_{w_k^d} \quad \sum_k r_k w_k^d, \tag{3.7.1a}$$

$$\text{subject to} \quad \sum_k a_{ik} w_k^d \leq c, \quad \forall i \in L, \tag{3.7.1b}$$

$$w_k^d \leq \sum_t \lambda_t \lambda_{kt}, \quad \forall k \in K, \tag{3.7.1c}$$

$$\tag{3.7.1d}$$

Since $\mathbf{A}$ is totally unimodular, if $\lambda_{k,t} \in \{0,1\}$, the solution of the above the linear program is integral. We can then define $x_k = w_k^d, \forall k \in K$ and $y_l = 0, \forall l \in L$ which is a feasible solution to the value function formulation. Therefore, only-partitioning is optimal.

For sufficiently long horizon, $H > \max_{l \in L} \max_{t:\lambda_t \lambda_{kt} > 0} \{\frac{c_l}{\lambda_t \lambda_{kt}}\}$, it is easy to check that the constraint $w_k^d \leq \sum_t \lambda_t \lambda_{kt}$ becomes redundant. We again obtain integral solution and a similar argument as above follows. $\square$.

3.7.2 Numerical Results

Upper bounds and revenues for train 1, class 1A, 3A and train 2, class 1A, 2A, 3A.

Train	Class	α	Horizon	TOP (time)	LR (time)	Δ^{DLP}_{TOP}	Δ^{DLP}_{LR}	Δ^{TOP}_{LR}
1	1A	1.0	50	27.4 ± 4.6	182.4 ± 6.5	1.6 ± 0.1	3.3 ± 0.3	1.7 ± 0.2
1	1A	1.0	100	40.0 ± 1.4	231.7 ± 7.6	0.9 ± 0.1	2.0 ± 0.2	1.2 ± 0.2
1	1A	1.0	200	92.4 ± 0.9	371.8 ± 0.4	0.4 ± 0.1	1.2 ± 0.1	0.8 ± 0.2
1	1A	1.0	300	142.1 ± 3.8	511.2 ± 2.9	0.3 ± 0.0	0.9 ± 0.1	0.7 ± 0.1
1	1A	1.1	50	22.6 ± 1.8	172.6 ± 3.5	1.9 ± 0.0	3.8 ± 0.3	2.0 ± 0.3
1	1A	1.1	100	40.0 ± 0.8	229.4 ± 10.5	0.8 ± 0.1	1.9 ± 0.1	1.1 ± 0.1
1	1A	1.1	200	91.8 ± 3.9	350.1 ± 11.1	0.4 ± 0.0	1.1 ± 0.1	0.7 ± 0.1
1	1A	1.1	300	147.8 ± 3.4	472.7 ± 5.1	0.3 ± 0.1	1.0 ± 0.1	0.7 ± 0.0
1	1A	1.2	50	22.1 ± 0.4	170.3 ± 5.6	2.0 ± 0.2	4.3 ± 0.2	2.3 ± 0.3
1	1A	1.2	100	39.0 ± 1.2	218.8 ± 6.4	0.9 ± 0.0	2.0 ± 0.1	1.1 ± 0.1
1	1A	1.2	200	88.1 ± 1.5	332.1 ± 8.3	0.5 ± 0.1	1.3 ± 0.0	0.7 ± 0.1
1	1A	1.2	300	143.4 ± 6.0	444.4 ± 6.8	0.4 ± 0.1	1.0 ± 0.1	0.6 ± 0.2
1	1A	1.3	50	21.8 ± 0.4	164.9 ± 6.7	1.9 ± 0.3	4.2 ± 0.2	2.3 ± 0.1
1	1A	1.3	100	39.8 ± 1.7	209.8 ± 5.3	1.1 ± 0.2	2.1 ± 0.2	1.0 ± 0.1
1	1A	1.3	200	86.9 ± 1.7	313.4 ± 3.2	0.5 ± 0.1	1.3 ± 0.1	0.7 ± 0.1
1	1A	1.3	300	140.4 ± 5.8	426.0 ± 11.2	0.3 ± 0.0	1.0 ± 0.0	0.7 ± 0.1
Average						0.9	2.0	1.1

Table 3.4: Upper bound comparison: Train 1, Class 1A.

Train	Class	α	Horizon	TOP (time)	LR (time)	Δ^{DLP}_{TOP}	Δ^{DLP}_{LR}	Δ^{TOP}_{LR}
1	3A	1.0	50	24.0 ± 0.5	183.8 ± 6.9	1.9 ± 0.2	4.3 ± 0.4	2.4 ± 0.4
1	3A	1.0	100	40.8 ± 0.6	232.6 ± 5.3	1.1 ± 0.0	2.5 ± 0.2	1.5 ± 0.2
1	3A	1.0	200	92.5 ± 1.5	340.2 ± 3.3	0.6 ± 0.0	1.6 ± 0.1	1.1 ± 0.1
1	3A	1.0	300	150.7 ± 1.7	474.5 ± 8.8	0.4 ± 0.0	1.3 ± 0.0	0.9 ± 0.0
1	3A	1.1	50	23.2 ± 0.6	175.9 ± 7.5	2.0 ± 0.1	4.5 ± 0.3	2.5 ± 0.2
1	3A	1.1	100	41.8 ± 1.2	220.4 ± 5.7	1.0 ± 0.1	2.5 ± 0.2	1.5 ± 0.2
1	3A	1.1	200	90.4 ± 1.1	324.5 ± 7.6	0.5 ± 0.0	1.8 ± 0.0	1.3 ± 0.0
1	3A	1.1	300	146.0 ± 1.0	443.5 ± 12.0	0.3 ± 0.0	1.3 ± 0.1	1.0 ± 0.1
1	3A	1.2	50	23.4 ± 1.0	172.7 ± 14.4	2.0 ± 0.2	5.0 ± 0.1	3.0 ± 0.3
1	3A	1.2	100	41.3 ± 1.5	211.0 ± 16.3	1.1 ± 0.1	2.7 ± 0.2	1.6 ± 0.1
1	3A	1.2	200	88.5 ± 1.3	306.3 ± 3.6	0.5 ± 0.0	1.8 ± 0.1	1.3 ± 0.1
1	3A	1.2	300	141.8 ± 1.8	411.1 ± 2.8	0.4 ± 0.0	1.3 ± 0.1	0.9 ± 0.0
1	3A	1.3	50	22.9 ± 0.3	168.9 ± 4.8	2.6 ± 0.4	5.4 ± 0.2	2.9 ± 0.3
1	3A	1.3	100	39.6 ± 2.3	191.6 ± 7.4	1.4 ± 0.2	2.9 ± 0.1	1.5 ± 0.3
1	3A	1.3	200	86.8 ± 1.3	288.7 ± 2.8	0.7 ± 0.1	1.8 ± 0.1	1.1 ± 0.2
1	3A	1.3	300	138.7 ± 0.5	398.1 ± 12.8	0.5 ± 0.1	1.4 ± 0.0	0.9 ± 0.1
Average						1.1	2.6	1.6

Table 3.5: Upper bound comparison: Train 1, Class 3A.

Train	Class	α	Horizon	TOP (time)	LR (time)	Δ_{TOP}^{DLP}	Δ_{LR}^{DLP}	Δ_{LR}^{TOP}
2	1A	1.0	50	19.2 ± 4.9	152.5 ± 8.5	2.0 ± 0.1	3.4 ± 0.5	1.4 ± 0.6
2	1A	1.0	100	22.0 ± 0.3	190.6 ± 2.5	1.0 ± 0.1	1.9 ± 0.3	0.9 ± 0.2
2	1A	1.0	200	41.8 ± 1.2	265.6 ± 7.2	0.6 ± 0.1	1.5 ± 0.2	0.9 ± 0.1
2	1A	1.0	300	73.1 ± 1.4	362.1 ± 7.1	0.3 ± 0.0	1.2 ± 0.1	0.8 ± 0.1
2	1A	1.1	50	13.6 ± 0.1	148.5 ± 1.9	2.0 ± 0.1	3.8 ± 0.5	1.9 ± 0.5
2	1A	1.1	100	22.7 ± 1.6	185.5 ± 7.4	1.1 ± 0.2	2.4 ± 0.4	1.3 ± 0.2
2	1A	1.1	200	43.2 ± 1.2	253.1 ± 2.3	0.6 ± 0.0	1.3 ± 0.1	0.8 ± 0.1
2	1A	1.1	300	71.1 ± 2.7	342.7 ± 12.5	0.3 ± 0.0	1.0 ± 0.1	0.6 ± 0.1
2	1A	1.2	50	13.5 ± 0.3	141.9 ± 5.8	2.0 ± 0.3	3.9 ± 0.7	1.9 ± 0.5
2	1A	1.2	100	21.7 ± 0.4	177.0 ± 3.1	1.0 ± 0.0	2.0 ± 0.1	1.1 ± 0.1
2	1A	1.2	200	41.6 ± 0.9	243.0 ± 7.4	0.5 ± 0.0	1.4 ± 0.1	0.9 ± 0.0
2	1A	1.2	300	71.2 ± 2.2	326.5 ± 4.6	0.3 ± 0.1	1.1 ± 0.1	0.7 ± 0.1
2	1A	1.3	50	13.5 ± 0.3	137.1 ± 2.5	2.1 ± 0.3	3.8 ± 0.3	1.8 ± 0.4
2	1A	1.3	100	21.8 ± 0.5	175.9 ± 3.9	1.0 ± 0.0	2.1 ± 0.1	1.2 ± 0.1
2	1A	1.3	200	41.0 ± 0.8	234.8 ± 3.2	0.5 ± 0.1	1.4 ± 0.2	0.9 ± 0.1
2	1A	1.3	300	66.0 ± 0.6	300.2 ± 4.5	0.3 ± 0.1	1.1 ± 0.1	0.8 ± 0.1
Average						1.0	2.0	1.1

TABLE 3.6: Upper bound comparison: Train 2, Class 1A.

Train	Class	α	Horizon	TOP (time)	LR (time)	Δ_{TOP}^{DLP}	Δ_{LR}^{DLP}	Δ_{LR}^{TOP}
2	2A	1.0	50	14.7 ± 0.3	158.4 ± 4.2	2.2 ± 0.2	3.8 ± 0.8	1.6 ± 1.0
2	2A	1.0	100	24.0 ± 0.4	195.6 ± 5.7	1.3 ± 0.1	2.8 ± 0.6	1.5 ± 0.6
2	2A	1.0	200	45.5 ± 0.4	275.3 ± 4.7	0.8 ± 0.0	1.7 ± 0.2	1.0 ± 0.2
2	2A	1.0	300	78.1 ± 3.7	370.5 ± 5.1	0.4 ± 0.1	1.4 ± 0.3	0.9 ± 0.4
2	2A	1.1	50	14.7 ± 0.4	155.7 ± 5.3	2.3 ± 0.2	4.5 ± 0.5	2.3 ± 0.7
2	2A	1.1	100	24.1 ± 0.2	185.6 ± 5.6	1.2 ± 0.1	2.8 ± 0.5	1.6 ± 0.5
2	2A	1.1	200	44.8 ± 1.1	261.7 ± 7.9	0.7 ± 0.1	2.0 ± 0.0	1.3 ± 0.0
2	2A	1.1	300	76.9 ± 2.6	350.5 ± 3.9	0.4 ± 0.0	1.6 ± 0.4	1.1 ± 0.4
2	2A	1.2	50	14.6 ± 0.1	142.6 ± 9.7	2.3 ± 0.1	4.5 ± 0.3	2.2 ± 0.4
2	2A	1.2	100	24.1 ± 1.2	187.3 ± 1.1	1.2 ± 0.3	2.6 ± 0.2	1.3 ± 0.3
2	2A	1.2	200	44.8 ± 0.8	249.7 ± 4.4	0.6 ± 0.1	1.6 ± 0.2	1.0 ± 0.3
2	2A	1.2	300	75.0 ± 1.5	339.6 ± 5.6	0.4 ± 0.0	1.5 ± 0.2	1.1 ± 0.1
2	2A	1.3	50	15.0 ± 0.7	152.8 ± 8.3	1.9 ± 0.4	4.3 ± 0.3	2.5 ± 0.1
2	2A	1.3	100	23.8 ± 0.1	178.4 ± 5.2	1.2 ± 0.1	2.1 ± 0.1	1.0 ± 0.2
2	2A	1.3	200	43.4 ± 1.2	240.3 ± 3.0	0.6 ± 0.0	1.7 ± 0.2	1.1 ± 0.2
2	2A	1.3	300	70.9 ± 1.9	323.5 ± 6.7	0.4 ± 0.0	1.3 ± 0.1	0.9 ± 0.1
Average						1.1	2.5	1.4

TABLE 3.7: Upper bound comparison: Train 2, Class 2A.

Train	Class	α	Horizon	TOP (time)	LR (time)	Δ_{TOP}^{DLP}	Δ_{LR}^{DLP}	Δ_{LR}^{TOP}
2	3A	1.0	50	15.4 ± 0.5	166.1 ± 6.7	3.1 ± 0.3	4.2 ± 0.6	1.2 ± 0.7
2	3A	1.0	100	25.4 ± 0.3	202.2 ± 4.8	1.5 ± 0.1	2.4 ± 0.5	1.0 ± 0.6
2	3A	1.0	200	49.8 ± 0.6	295.4 ± 2.4	0.7 ± 0.0	2.0 ± 0.1	1.4 ± 0.1
2	3A	1.0	300	85.7 ± 2.7	401.0 ± 1.1	0.4 ± 0.1	1.5 ± 0.3	1.1 ± 0.4
2	3A	1.1	50	15.7 ± 0.8	154.4 ± 7.1	2.6 ± 0.3	3.8 ± 0.6	1.3 ± 0.4
2	3A	1.1	100	25.4 ± 0.3	197.6 ± 2.2	1.4 ± 0.0	2.4 ± 0.1	1.1 ± 0.1
2	3A	1.1	200	48.8 ± 0.5	277.2 ± 7.5	0.7 ± 0.1	1.5 ± 0.1	0.7 ± 0.0
2	3A	1.1	300	83.5 ± 1.8	371.4 ± 4.0	0.5 ± 0.1	1.1 ± 0.0	0.6 ± 0.1
2	3A	1.2	50	15.3 ± 0.7	154.3 ± 6.6	2.3 ± 0.1	3.9 ± 0.3	1.7 ± 0.2
2	3A	1.2	100	25.7 ± 0.9	189.4 ± 3.6	1.2 ± 0.1	2.3 ± 0.1	1.1 ± 0.0
2	3A	1.2	200	48.6 ± 1.3	268.3 ± 10.9	0.6 ± 0.0	1.6 ± 0.1	1.0 ± 0.1
2	3A	1.2	300	82.5 ± 2.7	355.9 ± 5.9	0.4 ± 0.0	1.2 ± 0.0	0.9 ± 0.0
2	3A	1.3	50	15.6 ± 0.8	157.5 ± 6.0	2.1 ± 0.1	4.4 ± 0.2	2.3 ± 0.3
2	3A	1.3	100	25.4 ± 0.5	182.9 ± 12.0	1.0 ± 0.1	2.2 ± 0.1	1.2 ± 0.1
2	3A	1.3	200	48.3 ± 1.8	252.3 ± 3.3	0.6 ± 0.1	1.6 ± 0.1	1.0 ± 0.0
2	3A	1.3	300	78.7 ± 1.0	334.4 ± 1.2	0.4 ± 0.0	1.2 ± 0.1	0.8 ± 0.1
Average						1.2	2.3	1.1

TABLE 3.8: Upper bound comparison: Train 2, Class 3A.

Train	Class	α	Horizon	δ_{PA}^{LR}	δ_{PO}^{LR}	δ_{IR}^{LR}	δ_{DLP}^{LR}
1	1A	1.00	50	15.12*	1.69	36.02*	43.96*
1	1A	1.00	100	9.18*	2.24*	16.05*	24.5*
1	1A	1.00	200	2.94*	1.03*	9.15*	13.37*
1	1A	1.00	300	1.94*	0.92	6.51*	9.69*
1	1A	1.10	50	15.54*	3.28*	37.36*	46.05*
1	1A	1.10	100	7.81*	2.18*	17.85*	22.86*
1	1A	1.10	200	3.18*	1.05*	8.06*	13.01*
1	1A	1.10	300	0.28	1.06*	4.08*	6.55*
1	1A	1.20	50	11.72*	2.2*	31.67*	35.83*
1	1A	1.20	100	4.09*	1.38	14.17*	19.57*
1	1A	1.20	200	0.88	1.39*	6.23*	9.18*
1	1A	1.20	300	-1.03*	1.49*	3.21*	4.94*
1	1A	1.30	50	13.19*	3.64*	37.1*	43.87*
1	1A	1.30	100	6.82*	3.66*	19.38*	24.15*
1	1A	1.30	200	0.26	1.28*	8.05*	10.25*
1	1A	1.30	300	-1.83*	1.53*	2.87*	4.94*
Average				5.6	1.9	16.1	20.8

TABLE 3.9: Revenue comparison: Train 1, Class 1A.

Train	Class	α	Horizon	δ_{PA}^{LR}	δ_{PO}^{LR}	δ_{IR}^{LR}	δ_{DLP}^{LR}
1	3A	1.00	50	14.72*	1.44	35.96*	43.59*
1	3A	1.00	100	8.62*	2.76*	19.24*	26.05*
1	3A	1.00	200	3.28*	0.99	9.59*	13.97*
1	3A	1.00	300	2.02*	1.0*	5.39*	9.72*
1	3A	1.10	50	15.13*	2.69*	41.56*	46.13*
1	3A	1.10	100	6.83*	2.6*	17.87*	25.41*
1	3A	1.10	200	2.66*	1.17*	7.59*	11.87*
1	3A	1.10	300	0.31	1.66*	4.47*	7.99*
1	3A	1.20	50	14.21*	1.93	36.57*	38.78*
1	3A	1.20	100	5.87*	2.58*	14.72*	21.61*
1	3A	1.20	200	1.06	2.09*	8.92*	13.17*
1	3A	1.20	300	-0.52	1.81*	3.84*	7.83*
1	3A	1.30	50	11.48*	3.47*	45.0*	45.52*
1	3A	1.30	100	4.05*	3.02*	16.71*	22.68*
1	3A	1.30	200	0.08	1.53*	6.55*	9.85*
1	3A	1.30	300	-1.48*	2.07*	4.34*	6.99*
Average				5.6	2.1	17.4	21.9

TABLE 3.10: Revenue comparison: Train 1, Class 3A.

Train	Class	α	Horizon	δ_{PA}^{LR}	δ_{PO}^{LR}	δ_{IR}^{LR}	δ_{DLP}^{LR}
2	1A	1.00	50	12.04*	1.74	31.64*	31.85*
2	1A	1.00	100	5.46*	2.92*	13.95*	17.21*
2	1A	1.00	200	1.06*	1.2*	4.87*	7.37*
2	1A	1.00	300	-0.08	0.93*	3.05*	4.9*
2	1A	1.10	50	9.33*	2.51*	31.12*	32.46*
2	1A	1.10	100	4.63*	2.53*	11.73*	14.29*
2	1A	1.10	200	0.45	1.42*	5.92*	7.21*
2	1A	1.10	300	-0.94*	1.41*	3.53*	4.23*
2	1A	1.20	50	7.46*	2.74*	32.4*	33.76*
2	1A	1.20	100	3.46*	2.12*	13.52*	15.72*
2	1A	1.20	200	0.43	1.84*	5.46*	8.09*
2	1A	1.20	300	-1.4*	1.51*	2.89*	3.66*
2	1A	1.30	50	7.95*	2.54*	31.19*	31.48*
2	1A	1.30	100	3.58*	2.54*	11.58*	12.81*
2	1A	1.30	200	-0.68	1.16*	5.87*	6.8*
2	1A	1.30	300	-1.99*	1.94*	2.8*	3.59*
Average				3.2	1.9	13.2	17.7

TABLE 3.11: Revenue comparison: Train 2, Class 1A.

Train	Class	α	Horizon	δ_{PA}^{LR}	δ_{PO}^{LR}	δ_{IR}^{LR}	δ_{DLP}^{LR}
2	2A	1.00	50	14.15*	3.22*	26.28*	33.0*
2	2A	1.00	100	6.39*	1.98*	12.87*	16.18*
2	2A	1.00	200	3.21*	1.0*	7.67*	10.03*
2	2A	1.00	300	1.24*	0.40	4.5*	6.17*
2	2A	1.10	50	10.02*	2.23*	24.12*	25.26*
2	2A	1.10	100	4.47*	1.83*	13.59*	14.26*
2	2A	1.10	200	1.08*	1.36*	7.11*	8.48*
2	2A	1.10	300	-0.14	1.33*	4.15*	4.51*
2	2A	1.20	50	9.97*	1.58	30.31*	31.19*
2	2A	1.20	100	5.03*	1.54*	13.93*	16.83*
2	2A	1.20	200	1.72*	1.72*	7.2*	8.31*
2	2A	1.20	300	-1.32*	1.66*	4.01*	3.92*
2	2A	1.30	50	10.25*	1.96*	31.94*	35.46*
2	2A	1.30	100	4.03*	2.22*	15.96*	18.19*
2	2A	1.30	200	-0.69	1.66*	7.18*	8.35*
2	2A	1.30	300	-1.89*	1.78*	3.09*	4.26*
Average				4.2	1.7	13.3	15.2

TABLE 3.12: Revenue comparison: Train 2, Class 2A.

Train	Class	α	Horizon	δ_{PA}^{LR}	δ_{PO}^{LR}	δ_{IR}^{LR}	δ_{DLP}^{LR}
2	3A	1.00	50	16.43*	2.28*	27.71*	33.64*
2	3A	1.00	100	8.99*	2.01*	18.19*	20.36*
2	3A	1.00	200	2.79*	1.14*	7.91*	11.05*
2	3A	1.00	300	1.08*	0.71	3.4*	5.3*
2	3A	1.10	50	14.0*	2.76*	29.61*	30.92*
2	3A	1.10	100	5.09*	1.75*	14.73*	15.48*
2	3A	1.10	200	2.2*	1.45*	6.41*	7.69*
2	3A	1.10	300	0.23	1.16*	3.13*	4.54*
2	3A	1.20	50	12.13*	1.73*	29.65*	30.52*
2	3A	1.20	100	5.4*	1.56*	16.3*	18.29*
2	3A	1.20	200	1.78*	2.01*	7.7*	8.85*
2	3A	1.20	300	0.25	1.9*	4.64*	5.41*
2	3A	1.30	50	10.06*	1.7*	30.04*	33.1*
2	3A	1.30	100	5.38*	1.82*	15.77*	22.47*
2	3A	1.30	200	0.79	2.05*	6.67*	8.78*
2	3A	1.30	300	-0.28	2.57*	5.07*	5.46*
Average				5.4	1.8	14.2	16.4

TABLE 3.13: Revenue comparison: Train 2, Class 3A.

Bibliography

Adelman, Daniel. 2007. Dynamic bid prices in revenue management. *Operations Research*, 55 (4), 647-661.

Ahmed, Asrar, Milind G Sohoni, Chaithanya Bandi. 2022. Parameterized approximations for the two-sided assortment optimization. *Operations Research Letters*, 50 (4), 399-406.

Anderson, Ross, Itai Ashlagi, David Gamarnik, Yash Kanoria. 2014. A dynamic model of barter exchange. *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1925-1933.

Aouad, Ali, Daniela Saban. 2020. Online assortment optimization for two-sided matching platforms. *Available at SSRN 3712553*, .

Arnosti, Nick, Ramesh Johari, Yash Kanoria. 2014. Managing congestion in decentralized matching markets. *Proceedings of the Fifteenth ACM Conference on Economics and Computation*. 451-451.

Arora, SR, MC Puri, Kanti Swarup. 1977. The set covering problem with linear fractional functional. *Indian Journal of Pure Applied Mathematics*, 8 (5), 578-588.

Ashlagi, Itai, Anilesh K Krishnaswamy, Rahul Makhijani, Daniela Saban, Kirankumar Shiragur. 2019. Assortment planning for two-sided sequential matching markets. *arXiv preprint arXiv:1907.04485*, (`https://arxiv.org/abs/1907.04485`).

Avadhanula, Vashist, Jalaj Bhandari, Vineet Goyal, Assaf Zeevi. 2016. On the tightness of an LP relaxation for rational optimization and its applications. *Operations Research Letters*, 44 (5), 612-617.

Beal, Logan DR, Daniel C Hill, R Abraham Martin, John D Hedengren. 2018. Gekko optimization suite. *Processes*, 6 (8), 106.

Benson, Harold P. 2007. A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem. *European Journal of Operational Research*, 182 (2), 597-611.

Bertsimas, Dimitris, Sanne De Boer. 2005. Simulation-based booking limits for airline revenue management. *Operations Research*, 53 (1), 90-106.

Boros, Endre, Peter L Hammer. 2002. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123 (1-3), 155-225.

Borrero, Juan S, Colin Gillen, Oleg A Prokopyev. 2017. Fractional 0–1 programming: Applications and algorithms. *Journal of Global Optimization*, 69 (1), 255-282.

Bront, Juan José Miranda, Isabel Méndez-Díaz, Gustavo Vulcano. 2009. A column generation algorithm for choice-based network revenue management. *Operations Research*, 57 (3), 769-784.

Caro, Felipe, Victor Martínez-de Albéniz, Paat Rusmevichientong. 2014. The assortment packing problem: Multiperiod assortment planning for short-lived products. *Management Science*, 60 (11), 2701-2721.

Charnes, Abraham, William W Cooper. 1962. Programming with linear fractional functionals. *Naval Research logistics quarterly*, 9 (3-4), 181-186.

Chen, Danny Z, Ovidiu Daescu, Yang Dai, Naoki Katoh, Xiadong Wu, Jinhui Xu. 2000. Optimizing the sum of linear fractional functions and applications. *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*. Association for Computing Machinery, Society for Industrial and Applied Mathematics, 707-716.

Chen, Xi, Chao Shi, Yining Wang, Yuan Zhou. 2021. Dynamic assortment planning under nested logit models. *Production and Operations Management*, 30 (1), 85-102.

Ciancimino, A, G Inzerillo, Stefano Lucidi, Laura Palagi. 1999. A mathematical programming approach for the solution of the railway yield management problem. *Transportation science*, 33 (2), 168-181.

Davis, James M, Guillermo Gallego, Huseyin Topaloglu. 2014. Assortment optimization under variants of the nested logit model. *Operations Research*, 62 (2), 250-273.

Depetrini, Daniele, Marco Locatelli. 2011. Approximation of linear fractional-multiplicative problems. *Mathematical Programming*, 128 (1), 437-443.

Désir, Antoine, Vineet Goyal, Jiawei Zhang. 2022. Capacitated assortment optimization: Hardness and approximation. *Operations Research*, 70 (2), 893-904.

Dickerson, John P, Ariel D Procaccia, Tuomas Sandholm. 2013. Failure-aware kidney exchange. *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*. 323-340.

Dinkelbach, Werner. 1967. On nonlinear fractional programming. *Management Science*, 13 (7), 492-498.

Drezner, Zvi, Siegfried Schaible, David Simchi-Levi. 1990. Queueing-location problems on the plane. *Naval Research Logistics (NRL)*, 37 (6), 929-935.

Dutta, Goutam, Priyanko Ghosh. 2012. A passenger revenue management system (rms) for a national railway in an emerging asian economy. *Journal of Revenue and Pricing Management*, 11 487-499.

Feldman, Jacob, Huseyin Topaloglu. 2015. Bounding optimal expected revenues for assortment optimization under mixtures of multinomial logits. *Production and Operations Management*, 24 (10), 1598-1620.

Freund, Roland W, Florian Jarre. 2001. Solving the sum-of-ratios problem by an interior-point method. *Journal of Global Optimization*, 19 (1), 83-102.

Gale, David, Lloyd S Shapley. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69 (1), 9-15.

Gallego, Guillermo, Huseyin Topaloglu, et al. 2019. *Revenue management and pricing analytics*, vol. 209. Springer.

Golrezaei, Negin, Hamid Nazerzadeh, Paat Rusmevichientong. 2014. Real-time optimization of personalized assortments. *Management Science*, 60 (6), 1532-1551.

Gopalakrishnan, Raja, Narayan Rangaraj. 2010. Capacity management on long-distance passenger trains of indian railways. *Interfaces*, 40 (4), 291-302.

Hammer, I., P.L. Ivănescu, P.L. Hammer, P.I. Ivănescu, S. Rudeanu. 1968. *Boolean Methods in Operations Research and Related Areas*. Econometrics and Operations Research, Springer-Verlag. URL https://books.google.co.in/books?id=7EPxAAAAMAAJ.

Han, Shaoning, Andres Gomez, Oleg A Prokopyev. 2020. Fractional 0-1 programming and submodularity. *arXiv preprint arXiv:2012.07235*, .

Hansen, Pierre, Marcus V Poggi de Aragão, Celso C Ribeiro. 1990. Boolean query optimization and the 0-1 hyperbolic sum problem. *Annals of Mathematics and Artificial Intelligence*, 1 (1-4), 97-109.

Hansen, Pierre, Marcus V Poggi de Aragão, Celso C Ribeiro. 1991. Hyperbolic 0–1 programming and query optimization in information retrieval. *Mathematical Programming*, 52 (1), 255-263.

Jiang, Xiushan, Xiqun Chen, Lei Zhang, Ruifeng Zhang. 2015. Dynamic demand forecasting and ticket assignment for high-speed rail revenue management in china. *Transportation Research Record*, 2475 (1), 37-45.

Kunnumkal, Sumit, Victor Martínez-de Albéniz. 2019. Tractable approximations for assortment planning with product costs. *Operations Research*, 67 (2), 436-452.

Kuno, Takahito. 2002. A branch-and-bound algorithm for maximizing the sum of several linear ratios. *Journal of Global optimization*, 22 (1), 155-174.

Kuno, Takahito. 2005. A revision of the trapezoidal branch-and-bound algorithm for linear sum-of-ratios problems. *Journal of Global Optimization*, 33 (2), 215-234.

Li, Guang, Paat Rusmevichientong, Huseyin Topaloglu. 2015. The d-level nested logit model: Assortment and price optimization problems. *Operations Research*, 63 (2), 325-342.

Li, Han-Lin. 1994. A global approach for general 0–1 fractional programming. *European Journal of Operational Research*, 73 (3), 590-596.

Matsui, Tomomi. 1996. Np-hardness of linear multiplicative programming and related problems. *Journal of Global Optimization*, 9 (2), 113-119.

Megiddo, Nimrod. 1979. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4 (4), 414-424. URL `http://www.jstor.org/stable/3689226`.

Méndez-Díaz, Isabel, Juan José Miranda-Bront, Gustavo Vulcano, Paula Zabala. 2014. A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Applied Mathematics*, 164 246-263.

Ministry of Railways, India. 2019. Indian railwaysannual report & accounts. `https://indianrailways.gov.in/railwayboard/uploads/directorate/stat_econ/Year_Book/Indian%20Railways%20Annual%20Report%20%26%20Accounts%20English%202018-19.pdf`.

Nemhauser, George L, Laurence A Wolsey, Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14 (1), 265-294.

Page-Tickell, Rebecca, Elaine Yerby. 2020. *Conflict and shifting boundaries in the gig economy: An interdisciplinary analysis*. Emerald Group Publishing.

Pardalos, Panos M. 1996. *Continuous Approaches to Discrete Optimization Problems*. Springer US, Boston, MA, 313-325.

Prokopyev, Oleg A, Hong-Xuan Huang, Panos M Pardalos. 2005. On complexity of unconstrained hyperbolic 0–1 programming problems. *Operations Research Letters*, 33 (3), 312-318.

Rao, MR. 1971. Cluster analysis and mathematical programming. *Journal of the American statistical association*, 66 (335), 622-626.

Roth, Alvin E. 1984. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 92 (6), 991-1016.

Roth, Alvin E. 1991. A natural experiment in the organization of entry-level labor markets: regional markets for new physicians and surgeons in the united kingdom. *The American Economic Review*, 415-440.

Roth, Alvin E, Tayfun Sönmez, M Utku Ünver. 2004. Kidney exchange. *The Quarterly Journal of Economics*, 119 (2), 457-488.

Roth, Alvin E, Tayfun Sönmez, M Utku Ünver. 2007. Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *The American Economic Review*, 97 (3), 828-851.

Rusmevichientong, Paat, Zuo-Jun Max Shen, David B Shmoys. 2009. A ptas for capacitated sum-of-ratios optimization. *Operations Research Letters*, 37 (4), 230-238.

Rusmevichientong, Paat, David Shmoys, Chaoxu Tong, Huseyin Topaloglu. 2014. Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management*, 23 (11), 2023-2039.

Sen, Alper, Alper Atamtürk, Philip Kaminsky. 2018. A conic integer optimization approach to the constrained assortment problem under the mixed multinomial logit model. *Operations Research*, 66 (4), 994-1003.

Skiscim, Christopher C, Susan W Palocsay. 2001. Minimum spanning trees with sums of ratios. *Journal of Global Optimization*, 19 (2), 103-120.

Smith, Barry C, John F Leimkuhler, Ross M Darrow. 1992. Yield management at american airlines. *interfaces*, 22 (1), 8-31.

Sönmez, Tayfun, M Utku Ünver. 2010. Course bidding at business schools. *International Economic Review*, 51 (1), 99-123.

Talluri, Kalyan, Garrett Van Ryzin. 1998. An analysis of bid-price controls for network revenue management. *Management science*, 44 (11-part-1), 1577-1593.

Talluri, Kalyan, Garrett Van Ryzin. 2004. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50 (1), 15-33.

Tawarmalani, Mohit, Shabbir Ahmed, Nikolaos V Sahinidis. 2002. Global optimization of 0-1 hyperbolic programs. *Journal of Global Optimization*, 24 (4), 385-416.

Topaloglu, Huseyin. 2009. Using lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research*, 57 (3), 637-649.

Torrico, Alfredo, Margarida Carvalho, Andrea Lodi. 2020. Provable guarantees for general two-sided sequential matching markets. *arXiv preprint arXiv:2006.04313*, (`https://arxiv.org/abs/2006.04313`).

Tyson, Gareth, Vasile C Perta, Hamed Haddadi, Michael C Seto. 2016. A first look at user activity on tinder. *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 461-466.

Ünver, M Utku. 2010. Dynamic kidney exchange. *The Review of Economic Studies*, 77 (1), 372-414.

Van Ryzin, Garrett, Gustavo Vulcano. 2008. Simulation-based optimization of virtual nesting controls for network revenue management. *Operations research*, 56 (4), 865-880.

Williamson, Elizabeth Louise. 1992. Airline network seat inventory control: Methodologies and revenue impacts. Ph.D. thesis, Massachusetts Institute of Technology.

Wu, Tai-Hsi. 1997. A note on a global approach for general 0–1 fractional programming. *European Journal of Operational Research*, 101 (1), 220-223.

Yan, Zhenying, Xiaojuan Li, Qi Zhang, Baoming Han. 2020. Seat allocation model for high-speed railway passenger transportation based on flexible train composition. *Computers & Industrial Engineering*, 142 106383.

You, Peng-Sheng. 2008. An efficient computational approach for railway booking problems. *European Journal of Operational Research*, 185 (2), 811-824.

Zhang, Rui, Saied Samiedaluie, Dan Zhang. 2022. Product-based approximate linear programs for network revenue management. *Operations Research*, 70 (5), 2837-2850.