UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

The Rank Pricing Problem
A Mixed-Integer Linear Optimization Approach

El Problema de Tarificación Basado en Preferencias
Un Enfoque desde la Optimización Lineal Entera Mixta

Dª Concepción Domínguez Sánchez

2021

Programme de Doctorat en Sciences

Programa de Doctorado en Matemáticas

PhD Dissertation

The Rank Pricing Problem

A mixed-integer linear optimization approach

El Problema de Tarificación basado en Preferencias

Un enfoque desde la optimización lineal entera mixta

Author
Concepción Domínguez Sánchez

Supervisors
Prof. Dr. *Bernard Fortz*
Prof. Dr. *Martine Labbé*
Prof. Dr. *Alfredo Marín*

2021

To my mother

Resumen

Esta tesis está dedicada a un estudio en profundidad del Problema de Tarificación basado en Preferencias (del inglés, *Rank Pricing Problem* (RPP)) y dos generalizaciones. El RPP es un problema de optimización combinatoria que tiene como objetivo fijar el precio de los productos de una compañía para maximizar su beneficio. En él, intervienen clientes *unit-demand*, es decir, clientes que están interesados en varios de los productos de la empresa, pero pretenden comprar como mucho uno de ellos. Los clientes tienen un presupuesto fijo y clasifican los productos que les interesan formando un *ranking* del *mejor* al *peor*. Cuando la compañía fije los precios, cada cliente comprará su producto preferido de entre los que se puede permitir. En el RPP, asumimos que los productos se ofertan en cantidad ilimitada, lo cual encaja si consideramos que la compañía tiene suficientes productos para satisfacer la demanda, o cuando los productos se pueden producir rápidamente con un coste despreciable (por ejemplo, los bienes digitales). El RPP está clasificado como un problema de tarificación de multi-productos con clientes *unit-demand* y un modelo de consumo del cliente basado en sus preferencias.

El RPP se introdujo en Rusmevichientong et al. (2006). Los autores utilizaron datos obtenidos a través de una página web que ofrecía recomendaciones de vehículos a los consumidores. En esta página, los clientes potenciales escribían un presupuesto para su vehículo, y ordenaban las características de los vehículos de acuerdo a la importancia que les daban y a sus preferencias. Tras esto, la página web creaba una lista con varios de los vehículos de la compañía ordenados según las preferencias de los clientes y ajustados a su presupuesto. Estas listas fueron las usadas por Rusmevichientong et al. (2006) para optimizar los precios de los vehículos de la compañía. El trabajo que han realizado Rusmevichientong et al. (2006) e investigadores posteriores ha girado en torno al estudio de la complejidad del problema y al desarrollo de métodos de resolución no exactos, es decir, métodos que buscan *buenas soluciones* del problema, pero no necesariamente la óptima. Así, se ha probado que el RPP es uno de los conocidos como problemas *duros*, se han propuesto algoritmos basados en heurísticas, y también algoritmos de resolución que dan una cota del error cometido. En algunos casos, se han estudiado también casos particulares con propiedades interesantes, como añadir *escalas de precios* (o sea, restricciones que ordenan los precios de algunos productos antes de fijarlos, lo que tiene sentido si consideramos por ejemplo dos versiones de un producto, una básica y una premium). Sin embargo, no se ha llevado a cabo ningún estudio del RPP de métodos exactos para este problema de optimización combinatoria previo a este trabajo. Dentro de la Programación Matemática, nosotros hemos centrado nuestro trabajo en el desarrollo de métodos exactos para el problema. En concreto, nos centramos en dar formulaciones enteras-mixtas, linealizarlas y/o reforzarlas si corresponde, y desarrollar algoritmos de resolución basados

en dichas formulaciones que proporcionan una solución óptima.

Esta tesis doctoral consta de cuatro capítulos. El primero de ellos es un capítulo de introducción al problema y a los conceptos matemáticos presentes en la tesis, mientras que los tres siguientes se centran en cada uno de los problemas estudiados: el RPP y dos generalizaciones, una en la que se considera que los clientes pueden tener empates entre varios productos en su *ranking*, y otra en la que existe un número limitado de copias de cada producto en venta.

El Capítulo 1 tiene dos secciones diferenciadas. La primera sección contiene los conceptos fundamentales que se emplean a lo largo de la tesis. Aquí explicamos, pues, conceptos clave que aparecen en el trabajo como la importancia de las desigualdades válidas o en qué consiste la descomposición de Benders. También damos resultados teóricos tales como el Lema de Farkas o el Teorema de Dualidad Fuerte, en los que nos apoyamos a lo largo de la tesis. La segunda sección constituye una presentación de los resultados existentes sobre el RPP anteriores a este trabajo. Comienza con la definición del problema, seguida de una revisión de la literatura del RPP. Después se incluyen dos apartados que revisan trabajos relacionados con el RPP. El primero incluye problemas de tarificación en los que la decisión del consumidor está basada en la utilidad de los productos, mientras que el segundo contiene problemas de otras ramas de la optimización que tienen en común que la decisión de compra del consumidor también se basa en un *ranking* de los productos. Estos problemas han sido relevantes, por ejemplo, porque han motivado la introducción de la descomposición de Benders como una técnica de resolución adecuada para el RPP, o porque nos han servido para dar resultados de complejidad de un subproblema que surge en el Capítulo 4.

El segundo capítulo se centra en el estudio del RPP. En primer lugar, explicamos que el problema puede ser visto como un problema binivel: la compañía actúa como primer nivel (o *líder*) maximizando su beneficio; y el segundo nivel lo componen los clientes o *seguidores* que, forma independiente, buscan maximizar sus preferencias. Por esto, la primera formulación que presentamos es una formulación binivel no lineal con múltiples seguidores independientes, que procedemos a transformar en una formulación de un nivel utilizando resultados de programación lineal. A continuación, proponemos otra formulación, esta vez directamente en un nivel. Como la función objetivo de ambas formulaciones (que es la misma) es no lineal, dedicamos la siguiente sección a linealizarla de dos maneras, utilizando dos conjuntos distintos de variables continuas. Así, obtenemos cuatro modelos lineales enteros-mixtos. Después nos centramos en reforzar las cotas superiores dadas por la relajación lineal de los modelos. Para ello presentamos dos conjuntos de desigualdades válidas diseñados para acotar los valores de las variables continuas que intervienen en las funciones objetivo, reforzando así directamente las cotas dadas por las relajaciones lineales. Estos conjuntos tienen un número exponencial de desigualdades. En vez de incorporarlas todas a las formulaciones, optamos por diseñar un algoritmo de separación que determina cuáles son las más violadas por la solución óptima de la relajación. Nuestro método resulta ser muy eficiente en la separación de las desigualdades, lo que deriva en grandes mejoras de las cotas superiores. Tras la separación, añadimos las desigualdades válidas a la formulación dinámicamente, mediante un procedimiento de ramificación y cortes. Para finalizar, incluimos un apartado con técnicas de preprocesamiento que reducen el tamaño de las instancias del problema, y comparamos las formulaciones y los algoritmos teórica y computacionalmente.

Además, nos dimos cuenta de que las restricciones asociadas a las variables binarias de uno de nuestros modelos constituyen un caso particular de un Problema de Empaquetamiento de Conjuntos, por lo que decidimos realizar un análisis poliédrico de este subproblema de empaquetamiento. Unas de las mejores desigualdades que definen facetas de un problema de empaquetamiento de conjuntos, que además tienen relevancia práctica, son las obtenidas a partir de *cliques* (subgrafos maximales completos) del grafo intersección asociado. Por tanto, comenzamos identificando el grafo intersección asociado a nuestro problema, y llevamos a cabo un estudio en profundidad centrado en la caracterización de todos sus cliques, agrupándolos en dos familias y definiéndolos además mediante una expresión parametrizada común.

El Capítulo 3 está dedicado al estudio del Problema de Tarificación basado en Preferencias con Empates (RPPT por sus siglas en inglés, *Rank Pricing Problem with Ties*). En esta extensión del problema, asumimos que los clientes pueden expresar su indiferencia entre productos de su interés mediante empates en su lista de preferencia. Desde el punto de vista de la programación binivel, esto implica que el segundo nivel –dado por la decisión de compra de los consumidores– ya no tiene solución óptima única. En el caso de que un cliente esté interesado en dos productos por igual, parece natural asumir que, si se puede permitir ambos, comprará el más barato de los dos. Esto equivale a elegir una posición pesimista del problema del segundo nivel en el marco binivel. Aunque no abordaremos el problema desde la perspectiva binivel, esta caracterización tiene interés por cuanto muestra el incremento de la dificultad que constituye esta suposición.

En este capítulo presentamos una nueva formulación con variables de tres índices e introducimos dos métodos de resolución. En el primero, comenzamos proponiendo una formulación con variables de dos índices, para luego proyectar la anterior en esta, obteniendo un conjunto exponencial de desigualdades válidas que la refuerzan. La matriz asociada a las restricciones de los problemas de separación de dichas desigualdades posee la propiedad de los unos consecutivos, lo que nos permite reducir nuestros problemas de separación a problemas de flujo a coste mínimo y resolverlos mediante un algoritmo existente.

Alternativamente, resolvemos la formulación de tres índices siguiendo un esquema basado en la descomposición de Benders que aprovecha la separabilidad del problema en un problema *master* y varios subproblemas. Así, obtenemos un problema *master* con un conjunto de restricciones cuya separación se puede realizar resolviendo subproblemas lineales asociados a los clientes, y conseguimos identificar un subconjunto pequeño (polinómico) de restricciones del mismo que nos permite obtener una formulación *master* reducida válida para el RPPT, el Modelo de Benders. En la implementación de la descomposición de Benders, resolvemos el Modelo de Benders en vez del *master*, añadiendo desigualdades del llamado *método lazy*, es decir, resolviendo el problema mediante el algoritmo de ramificación y cortes pero añadiendo las desigualdades solo cuando se encuentran soluciones enteras en el árbol. Al contrario que en el método clásico, en el que el *master* se resuelve desde el principio cada vez que se añade un conjunto de desigualdades nuevo, aquí el problema solo se resuelve una vez, y las restricciones necesarias se añaden iterativamente. Finalmente, llevamos a cabo experimentos computacionales exhaustivos para evaluar la efectividad de los métodos de resolución.

El último capítulo de la tesis comprende el estudio del Problema de Tarificación Capacitado basado en Preferencias o *Capacitated Rank Pricing Problem* (CRPP) con envidia. En esta extensión, hemos asumido *precios de reserva* en los clientes que reflejan lo que están dispuestos a pagar por cada producto, en vez de un solo presupuesto por consumidor. No obstante, la principal diferencia es que en el CRPP la compañía tiene un número limitado de productos y puede no ser capaz de satisfacer la demanda de todos los clientes. Esta es una hipótesis realista que se da, por ejemplo, en compañías que venden productos hechos a mano, en la industria del lujo o en compañías que venden bienes no materiales (experiencias, servicios, entradas para eventos, etc.).

Cuando el suministro es limitado, se pueden dar dos soluciones dependiendo de la postura de la empresa. Si su objetivo principal es la maximización del beneficio, consideraremos factibles soluciones con clientes envidiosos. En estas soluciones con envidia, hay clientes que se pueden permitir un producto y les interesa, pero la compañía no se lo puede proporcionar porque está agotado, por lo que optan por comprar otro producto que les interesa menos (o por no comprar ninguno). Aunque generalmente una solución con envidia da mayor beneficio que una sin envidia, la compañía puede evitar las soluciones con envidia por distintos motivos, como prevenir el malestar de los clientes a largo plazo, fidelizándolos. Estas dos variantes del CRPP tienen características muy distintas. La primera de ellas es que, mientras que en la versión sin envidia se respetan totalmente las preferencias de los clientes, en caso de envidia las preferencias solo se respetan si el producto no se agota en la solución final (y esto requiere nuevas restricciones que modelicen las preferencias). Otra diferencia quizás menos intuitiva es que en el caso sin envidia un vector de precios determina unívocamente la asignación de productos a los clientes, por lo que una tarificación deriva directamente en una solución factible única del problema. Sin embargo, en la versión con envidia la asignación de clientes no está unívocamente determinada por la tarificación, luego se tienen que dar ambas para caracterizar una solución factible. De hecho, para comprender mejor el problema, comenzamos nuestro estudio analizando la complejidad del subproblema dado por la asignación de los productos a los clientes (asumiendo una tarificación fija) en el caso con envidia. En esta sección, reducimos un problema de emparejamiento NP-duro llamado *problema del matrimonio estable con empates y listas incompletas* a nuestro problema, demostrando así que es NP-completo. El caso general con envidia es, por tanto, más difícil de resolver que el otro, y además técnicas utilizadas anteriormente como la descomposición de Benders o las técnicas de preprocesamiento no se pueden aplicar a esta versión. Las cualidades que tienen las soluciones con envidia, y el hecho de que fueran inherentes del caso con capacidades (ya que las versiones con suministro ilimitado siempre dan soluciones sin envidia), fueron las que nos hicieron decidirnos por el estudio de la versión con envidia del CRPP.

La primera formulación propuesta para el CRPP es una formulación lineal entera mixta con un conjunto de variables de decisión de tres índices. En ella utilizamos dos conjuntos de variables ya empleados en el caso sin capacidades, más un tercer conjunto que sirve para modelizar la capacidad, pues indica cuándo un producto se ha agotado a cierto precio. Junto a esta formulación se introducen varias familias de desigualdades válidas basadas en las restricciones de capacidad. También presentamos una segunda formulación reducida que solo emplea variables con dos índices, y varios conjuntos de desigualdades para reforzarla. Proyectando la formulación de tres índices en la reducida mediante el Lema de Farkas, obtenemos el primer conjunto parametrizado de desigualdades válidas. De este conjunto, separamos un subconjunto de tamaño polinómico que domina a un conjunto de desigualdades propuesto para el RPP en el Capítulo 2. Además, hacemos uso de las restricciones de capacidad para desarrollar un segundo conjunto de desigualdades (de tamaño exponencial en este caso), cada una de las cuales involucra a un subconjunto de clientes interesado en un mismo producto. A continuación, se describen siete algoritmos basados en las dos formulaciones que emplean el método de ramificación y cortes para la separación de las desigualdades. Finalmente, comparamos los algoritmos en el estudio computacional, donde se muestra que las desigualdades válidas propuestas tienen un impacto directo tanto en el tiempo de resolución de ambos modelos como en la reducción de la cota de la relajación lineal.

Publicaciones y colaboraciones

El contenido original de los Capítulos 2, 3 y 4 ha sido enviado a revistas internacionales para su publicación. Concretamente, el contenido del Capítulo 2 es fruto de la colaboración con las profesoras Carmen Galé y Herminia Calvete de la Universidad de Zaragoza, y fue publicado en la revista *Computers & Operations Research* bajo el título *The rank pricing problem: Models and branch-and-cut algorithms* (Calvete et al. (2019)). El segundo capítulo está basado en los resultados del artículo *The rank pricing problem with ties*, publicado en la revista *European Journal of Operational Research* (Domínguez et al. (2021b)). Y el Capítulo 4 se basa en el contenido del manuscrito titulado *The Capacitated Rank Pricing with envy*, que ha sido enviado recientemente a la revista *Computers & Operations Research* y ya ha recibido críticas positivas (Domínguez et al. (2021a)).

Abstract

This doctorate is entirely devoted to an in-depth study of the *Rank Pricing Problem* (RPP) and two generalizations. The RPP is a combinatorial optimization problem which aims at setting the prices of a series of products of a company to maximize its revenue. This problem is specified by a set of unit-demand customers, that is, customers interested in a subset of the products offered by the company which intend to buy at most one of them. To do so, they count on a fixed budget and they rank the products of their interest from the *best* to the *worst*. Once the prices are established by the company, they will purchase their highest-ranked product among the ones they can afford. In the RPP, it is assumed an unlimited supply of products, which is consistent with a company having enough copies of a product to satisfy the demand, or with a setting where the products can be produced quickly at negligible cost (e.g. digital goods). It is hence classified as a multi-product pricing problem with unit-demand customers and a ranking-based consumer choice model.

The RPP was introduced in Rusmevichientong et al. (2006). They leveraged data collected through a website offering car recommendations to customers, with the aim of optimizing the price of the vehicles of the company. In this website, potential customers could define a budget for their vehicle, and order the different features of the vehicles according to their preferences and the importance they attached to them. Afterwards, the website offered a list of vehicles ordered with respect to the preferences of the customers and adjusted to their budget. The work carried out in Rusmevichientong et al. (2006) and in subsequent papers centered on the study of the complexity of the problem, and in the development of inexact resolution methods, i.e. methods that seek for a "good solution" of the problem, but not necessarily the optimal one. Thus, it has been proved that the RPP belongs to the family of NP-hard problems, and heuristics and approximation algorithms (with performance bounds) have been proposed. The properties of particular cases like the RPP with price-ladder (an ordering in the prices of the products made before the prices are established that is consistent with having e.g. two versions of a product, a basic one and a premium one) have also been studied. Yet no article before our work has derived exact methods for this combinatorial problem. Within the field of Mathematical Programming, our work focuses on the introduction of mixed-integer formulations, their linearization and/or strengthening when necessary, and the development of algorithms based on such formulations that provide an optimal solution.

This dissertation consists of four chapters. Chapter 1 introduces the RPP problem and the mathematical concepts present in the work, whereas each of the next three chapters tackles the resolution of each of the problems of study: the RPP and two generalizations, one that considers ties in the preference lists of the customers, and a second one that

assumes a limited number of copies of the products for sale.

Chapter 1 has two distinguishable sections. The first section contains the main concepts used throughout the thesis. Here we detail the key concepts that appear in the work, such as the importance of developing valid inequalities or what a Benders decomposition entails. We also provide theoretical results used throughout the work, like the Farkas' Lemma of the Strong Duality Theorem. The second section constitutes a presentation of the existing results on the RPP prior to this work. It begins with the problem's definition, followed by a literature review of the RPP. Then, we include two subsections to look over problems related to the RPP. The first subsection comprises pricing problems in which the customers' decision purchase is based on the utility of the products. Conversely, the second one contains problems from other optimization areas with the common denominator of customers whose purchasing decision is based on a ranking of the products. These problems have proven relevant, for instance, in the motivation for introducing Benders decomposition as a plausible resolution technique for the RPP, and have been useful to derive complexity results of a subproblem that arises in Chapter 4.

Chapter 2 is devoted to the study of the RPP. The RPP can be modeled as a bilevel program, in which the first level (or leader) is the company intending to maximize its profit, and the second level (or follower) is given by the customers, who independently aim at purchasing their highest-ranked item. Therefore, the first formulation we develop is a bilevel nonlinear one with multiple independent followers. We then transform this model into a single-level formulation using results from linear programming. Next, we also formulate the problem straightforwardly as a single-level one. Since the two single-level models had a nonlinear objective function (which was the same for both), we devote the next section to two linearizations of the models by means of two different sets of continuous variables. These linearizations result in four mixed-integer linear models. We then focus on tightening the upper bounds given by the linear relaxations of the models. To do so, we develop two sets of valid inequalities that include an exponential number of inequalities. Instead of incorporating all of them into the formulations, we design a polynomial procedure to determine which ones are most violated by the optimal solutions of the relaxations. Our method is proved to be very efficient in the separation of the inequalities, which in turn leads to great improvements in the upper bounds. After separating them, we add them dynamically into our models in a branch-and-cut framework. Finally, we include a section with preprocessing techniques designed to reduce the size of the RPP instances, and we compare our formulations and algorithms theoretically and computationally.

Furthermore, we exploit the fact that the constraints associated to the binary variables of one of our single-level linear models constitute a special case of a Set Packing Problem (SPP). We therefore carry out a polyhedral analysis of this set packing subproblem. One of the best known facet-defining inequalities of an SPP, and also those of practical relevance when solving the problem, are the ones determined by the maximal complete subgraphs (cliques) of the associated intersection graph. Thus, we start by identifying the graph associated to our specific subproblem and carry out an in-depth study to characterize all its cliques, grouping them in several families but also finding a way to represent all of them in a common parametrized expression. As well as deriving strong valid inequalities for the models, this characterization enables us to prove that the original models are already tight, since they contain mainly inequalities which are facet-defining in the corresponding subproblems.

Chapter 3 is dedicated to the *Rank Pricing Problem with Ties* (RPPT), an extension of the RPP where we consider that customers can express indifference among products in their preference list. From the bilevel point of view, this implies that the second level problem (the one given by the customers' purchasing decision) no longer has a unique optimal solution. If we suppose that a customer is equally interested in two products, it seems natural that he would purchase the cheapest one when he can afford both. This is equivalent to choosing a *pessimistic solution* of the lower level problem in the bilevel framework. Although we will not approach the resolution of the RPPT from the bilevel perspective, this characterization is of interest in terms of showing the increase in the difficulty that this assumption constitutes.

We present a new three-index integer formulation for the RPPT and introduce two resolution approaches. In the first one, we begin by introducing a formulation with two-index variables. We then project out the three-index formulation onto the latter one, deriving an exponential set of valid inequalities that strengthen it. The matrix associated to the constraints of the separation problems of these inequalities possesses the consecutive ones property, and this allows us to reduce our separation problems to min-cost flow problems and solve them using an existing algorithm.

Alternatively, we solve the three-index formulation following a Benders decomposition approach that leverages the separability of the problem into a master problem and several subproblems. We obtain a master problem with a set of constraints whose separation can be done by solving linear subproblems associated to each customer, and we are able to identify a small (polynomial) subset of constraints from the previous set to obtain a reduced master formulation, the Benders Model, that is valid for the RPPT. For the implementation of the Benders decomposition, we solve the Benders Model instead of the master problem, adding valid inequalities in the so-called *lazy fashion*, that is, solving the master problem using a branch-and-bound solver and checking for violated inequalities only at each integer solution generated in the branching tree. As opposed to the classical approach, where the master problem is solved from scratch each time a new set of constraints is added, in this approach the problem is only solved once, and constraints are added iteratively when needed. We finally carry out extensive computational experiments to assess the performance of the resolution approaches.

The last chapter of the thesis is devoted to a generalization of the problem that we have named the *Capacitated Rank Pricing Problem* (CRPP) *with envy*. For this generalization, we have considered reservation prices of customers for the different products that reflect their willingness to pay, instead of a single budget per customer. However, the main difference is that, in the CRPP, the company has a limited supply of products and might not be able to satisfy all the customers' requests. This is a realistic assumption that we can find in many companies. For instance, companies selling hand-made or customized products, in the luxury industry, or in companies offering all sorts of events tickets or other kinds of non-material goods (experiences, services), to name but a few.

When the supply is limited, two possible solutions can arise depending on the stance of the company. If the main objective is the maximization of the profit, then we will consider as feasible solutions those with envious customers. In these solutions with envy, some customers can afford a product and are interested in it, but the company cannot provide it because it has sold all the copies to other customers, so they end up purchasing a different (and less preferred) product (or none). Although an solution with envy usually provides higher benefit than an envy-free one, a company might decide to consider envy-free solutions for different reasons, such as preventing discomfort among long-term customers, thus ensuring fidelity. These two variants of the CRPP have very different characteristics. One of them is that, while in the envy-free setting the preferences are fully respected, when there is envy the preferences of the customers are respected if and only if the product is not sold out in the final solution (and this calls for a different set of constraints to model the preferences). Another perhaps less intuitive feature is that in the envy-free case a vector of prices univocally determines the allocation of the products to the customers, so a unique feasible solution for the problem can be straightforwardly derived from a pricing. However, in the envy setting, an allocation of the products needs to be provided, along with a vector of prices, to characterize a feasible solution. In fact, in order to get a deeper understanding of the problem, we begin by analyzing the complexity of the allocation subproblem in the envy setting. We reduce a particular NP-hard matching problem called the *Stable Marriage problem with Ties and Incomplete Lists* to our problem, proving in this way that it is NP-complete. The general envy case is hence more difficult to solve than the envy-free one, and previously developed procedures like Benders decomposition or our preprocessing techniques do not apply to this extension. These particular features that appear when we allow for envy (and the fact that they do not occur in the RPP with unlimited supply, which always has envy-free solutions) were the main reason we decided to focus on the CRPP with envy.

The first formulation we derive for the CRPP is a mixed-integer linear formulation with a set of three-index variables. We use two sets of binary variables previously used for the unlimited case, and a third set to model the capacity, representing when a product is sold out at a certain price. Along with the formulation, we introduce several families of valid inequalities based on the capacity constraints that reinforce the model. We then present a second formulation with only sets of two-index variables, and several sets of inequalities to strengthen it as well. We begin by projecting out the three-index formulation onto the reduced one by means of Farkas' Lemma, obtaining a parametrized set of valid inequalities and separating a strong polynomial-size family that dominates an exponential set introduced for the uncapacitated version of the problem in Chapter 2. We also take advantage of the constraints concerning the capacity to derive an entirely new set of valid inequalities (of an exponential size in this case), each of them involving a subset of the customers interested in a given product. Afterwards, we develop seven resolution algorithms based on the formulations, the separation of the inequalities and their inclusion in branch-and-cut fashions. We then compare them in the computational study, where we show that the valid inequalities proposed for both models have a direct impact in the reduction of the linear relaxation bound and the total resolution times.

Publications and collaborations

The original content from Chapters 2, 3 and 4 has been submitted to international journals for publication. Specifically, the work gathered in Chapter 2 is the result of a collaboration with Prof. Carmen Galé and Prof. Herminia Calvete from the University of Zaragoza, and was published in the journal *Computers & Operations Research* under the name *The rank pricing problem: Models and branch-and-cut algorithms* (Calvete et al. (2019)). The second chapter is based on the results included in the article *The rank pricing problem with ties*, recently published in the *European Journal of Operational Research* (Domínguez et al. (2021b)). And Chapter 4 corresponds with the content of a manuscript entitled *The Capacitated Rank Pricing with envy*, which has been recently sent to the journal *Computers & Operations Research* and has already received positive reviews (Domínguez et al. (2021a)).

Acknowledgments

First of all, I would like to express my gratitude to my supervisors, Prof. Alfredo Marín, Prof. Martine Labbé and Prof. Bernard Fortz. I feel very lucky to have had the chance to work with such smart, witty and kind people, and learning from them and collaborating with them these past four years has been an absolute joy. Alfredo is the reason I decided to do a PhD in the first place, since it was only after enjoying researching with him for my undergraduate and master thesis that I considered this path as a possibility. He has been incredibly supportive when I most needed it, both at a professional and at a personal level, and I truly appreciate it. Martine has encouraged me to always go a step further and to think out of the box. She also found me the position in Lille that allowed me to do research within the Inocs team and enjoy a great year there before I was awarded a grant in Brussels. Merci beaucoup, Martine. And thank you as well to Bernard for being ready to help with any issue I had.

This work has also given me the opportunity to discover Lille and Brussels, to travel and do conferences in beautiful places and to live experiences that have largely contributed to my personal growth and that I treasure. Thank you to Luce Brotcorne and Frédéric Semet for welcoming me in the Inocs family. Thank you to all the people that I was lucky to meet along the journey and that have been part of lots of meaningful conversations, gatherings, conferences, long walks, parties, vintage markets, concerts, trips and much more. Julia, Émilie, Léo, Mathieu, Paula, Vanessa, Palma, Víctor, Fränk, Renaud. You have been my friends and my family when I was abroad.

And thank you to my family for their unconditional support and love. To Marina and José Antonio, already family at this point. To my aunts and uncles, cousins, to Victoria, to my father and brother, for always being there for me. To my mum, always ready to help with everything she could. To my sister, who always tells me what I need to hear and gives me the best advice. And to Pedro, for knowing me so well and for all his support during the good and the not so good times, his wise words, and his love.

The research of the author has been developed with the financial support of a research contract in Centrale Lille - Inria and by the "Fonds de la Recheche Scientifique" under a FRIA grant. It has also been partially supported by the projects MTM2015-65915-R (MINECO, Spain) and PID2019-110886RB-I00 (MICINN, Spain). They allowed me to live in Lille and Brussels, carry out two stays at the University of Murcia and to attend many conferences and research visits.

Contents

Chapter 1

Introduction

This dissertation is devoted to the study of decision problems from a mathematical programming perspective within the frame of Operations Research, a discipline concerned with improving decision-making. The first part of the chapter summarizes the concepts from the field that will prove useful for the non expert reader, since they are the foundations upon which we build the results in the remaining chapters. This background material comprises definitions and concepts within the fields of Linear, Integer and Bilevel Optimization, as well as a brief section on Complexity Theory. The second part constitutes an overview of the Rank Pricing Problem, our main problem of study. It includes the introduction of the problem, a state of the art with the main results available prior to this work and a section concerning related problems.

1.1 Concepts

1.1.1 Linear Optimization

In this section, we touch on the subject of linear optimization as a first step towards the introduction of integer optimization. For further details, the reader is referred to Bertsimas and Tsitsiklis (1997); Wolsey (1998).

Linear Programming is a branch of Operations Research that describes and solves mathematical problems whose requirements can be modeled by means of linear relationships. A linear program (LP) (in the canonical form) can be stated by means of formulation

$$\begin{aligned} \max_{x} \quad & cx \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in \mathbb{R}^n_+, \end{aligned} \tag{1.1}$$

with $0 \in \mathbb{R}^n_+$, $c \in \mathbb{R}^n$, $A_{m\times n} = (a_{ij})$, $a_{ij} \in \mathbb{R}\ \forall i, j$, $b \in \mathbb{R}^m$. x is the vector of *decision variables*, cx is the (linear) *objective function* and the linear inequalities $Ax \leq b$ are called *constraints*, a_{ij} being the coefficients of the constraints, and b the vector of independent terms. A vector x^* satisfying all the constraints is called a *feasible solution*. If x^* maximizes the objective function (it satisfies $cx^* \geq cx\ \forall x$), then it is an *optimal solution*, and the value cx^* is called the *optimal value*. The set of all feasible solutions

$\{x \in \mathbb{R}^n_+ : Ax \leq b\}$ is called the *feasible region.* This region is a polyhedron, and a polytope when it is bounded. Problem (1.1) is *unfeasible* if the feasible region is empty (because there is no vector x satisfying all the constraints), and *unbounded* when $\forall h \in \mathbb{R}$ there exists x in the feasible region such that $cx > h$. Throughout the thesis, we will use the terms formulation and model indistinctly.

Our main goal is to understand how to solve linear optimization problems, since they are the basis for solving more complicated problems like integer or bilevel ones. In an LP, either there is an optimal solution, or the problem is unbounded, or it is unfeasible. Thus, in order to solve an LP, the first question to address is whether it is feasible or unfeasible, and bounded or unbounded. Furthermore, for feasible bounded LPs, it is essential to know how to characterize optimality. The third step is to obtain an efficient mechanism to derive optimal solutions.

Farkas' Lemma can be viewed as a characterization of the feasibility of a problem, and it can also help us determine if it is bounded. It has several (equivalent) variants, here we discuss one of them:

Lemma 1.1 (Farkas, Farkas (1902))**.** *Let A be a $m \times n$ matrix, and $b \in \mathbb{R}^n$. Then exactly one of the following statements holds:*

(1) $\exists x \in \mathbb{R}^n : Ax \leq b,\ x \geq 0$ or

(2) $\exists y \in \mathbb{R}^m : y^t A \geq 0,\ y^t b < 0,\ y \geq 0$.

If (1) holds, then there exists a feasible solution of (1.1), so the problem is feasible. On the other hand, deriving a vector y that satisfies (2) serves as a certificate of unfeasibility. There is also a variant of Farkas' Lemma which refers to feasibility:

Lemma 1.2. *Let A be a $m \times n$ matrix, and $b \in \mathbb{R}^n$. Then the following two statements are equivalent:*

(1) $\exists x \in \mathbb{R}^n : Ax \leq b,\ x \geq 0$ or

(2) $\forall y \in \mathbb{R}^m : y \geq 0,\ y^t A \geq 0,$ it follows $y^t b \geq 0$.

We use this variant as a method to project out variables and obtain valid inequalities in several chapters of the thesis.

Farkas' Lemma is also useful for the next step, since it is used in the proof of duality results. Duality theory constitutes a characterization of optimality for LPs. It deals with the relationship between a primal problem (any linear problem like (1.1)) and another LP, called the *dual.* For a given primal problem of the form (1.1), the corresponding dual problem is

$$\begin{aligned} \min_u \quad & b^t u \\ \text{s.t.} \quad & A^t u \geq c, \\ & u \in \mathbb{R}^m_+. \end{aligned} \tag{1.2}$$

Theorem 1.3 (Weak duality)**.** *Let x be a feasible solution to* (1.1) *and u a feasible solution to* (1.2)*. Then $c^t x \leq u^t b$.*

As a consequence, if the primal problem is unbounded, then the dual is infeasible, and vice versa. Another corollary is that when $c^t x = u^t b$, then x and u are optimal solutions to the primal and dual problems, respectively. The converse is the strong duality theorem, a central result on duality theory:

Theorem 1.4 (Strong duality). *If an LP has an optimal solution, so does its dual, and the optimal values of both problems are equal.*

Duality theory leads to algorithms to derive optimal solutions for LPs such as the *simplex method*. First, let us characterize a polyhedron P in terms of its extreme points and extreme rays:

Definition 1.5 (Extreme point). Let $P \in \mathbb{R}^n$ be a non-empty polyhedron. Then x is an *extreme point* (vertex) of P if there are no two points $x_1, x_2 \in P$ and $\lambda \in (0,1)$ such that $x = \lambda x_1 + (1-\lambda)x_2$. The vertices of a polyhedron P are its zero-dimensional faces.

Definition 1.6 (Extreme ray). Let $P = \{x \in \mathbb{R}^n : Ax \leq b\} \neq \emptyset$ be a nonempty polyhedron. Then $r \in P^0$ is a ray of P, where $P^0 = \{r \in \mathbb{R}^n : Ar \leq 0\}$. A ray r of P is an *extreme ray* if there are no two rays $r_1, r_2 \in P^0$ with $r_1 \neq \mu r_2$ for any $\mu \geq 0$, such that $r = \lambda r_1 + (1-\lambda) r_2$ for $\lambda \in (0,1)$.

Theorem 1.7 (Minkowski-Weyl Theorem, Minkowski (1896); Weyl (1950)). *The polyhedron $P \neq \emptyset$ can be represented as*

$$P = \{x \in \mathbb{R}^n : x = \sum_{k \in K} \lambda_k x^k + \sum_{j \in J} \mu_j r^j \text{ with } \sum_{k \in K} \lambda_k = 1, \lambda_k \geq 0 \ \forall k \in K, \mu_j \geq 0 \ \forall j \in J\}, \tag{1.3}$$

where $\{x^k\}_{\{k \in K\}}$ is the set of extreme points of P and $\{r^j\}_{\{j \in J\}}$ is the set of extreme rays of P.

In essence, resolution algorithms are based on the fact that, if the feasible region has an extreme point and the LP has a bounded optimal solution, then there exists an optimal solution of the LP that is an extreme point. However, the number of extreme points can be exponential, so a complete enumeration of the extreme points of the feasible set is generally not an option. The simplex method, given by Dantzig, searches through the set of extreme points, which are basic feasible solutions, to find an optimal one in an organized way. The algorithm begins in a random vertex, and at each step it moves to an adjacent vertex in the polytope, always seeking to increase the objective value. The algorithm is finite once we define a way to break the ties in the pivoting rules used to select the extreme points, avoiding cycles.

Still to this day, there is no proof that the simplex method runs in polynomial time (see Schrijver (1998)). The complexity of the method is intimately related to the Hirsch Conjecture, formulated by Warren M. Hirsch in a letter addressed to George Dantzig in 1957. The Hirsch Conjecture establishes that the diameter of the graph of a polytope of dimension d with m faces is bounded by $m - d$ (the diameter of a polytope is the maximum distance between its vertices, and such distance is measured as the number of edges from a shortest path between them). Thus, if the diameter of any polytope were polynomial in d and m, it is possible that, with a certain pivot rule, the simplex method runs in polynomial time for a problem with d variables and m constraints. On the contrary, if this diameter is exponential on n and d, then the complexity of the simplex method will necessarily be exponential as well. Recently, Francisco Santos has obtained a counterexample of the Hirsch Conjecture (Santos (2012)).

1.1.2 A note on Complexity Theory

Here we present a brief introduction of some of the definitions used in this thesis related to the field of Complexity Theory. For more details on the subject, we refer the reader to Garey and Johnson (1978, 1979); Sipser (2012); Conforti et al. (2014).

When developing an algorithm, an important aspect is the time it takes to produce a solution to our problem. This time is a function that measures the number of computational necessary arithmetic operations, and is normally associated with the size of the instance input, which is the space required to encode the given data (also a function, for instance the number of bits required for binary encoding). An algorithm solves a problem in *polynomial time* if this time is polynomially bounded by the size of the input. In such case, usually the algorithm is called *polynomial-time algorithm* (or simply polynomial algorithm), and the problem belongs to complexity class P.

Another important complexity class encompasses *nondeterministic polynomial-time* (NP) decision problems. A *decision problem* is a problem that can be posed as a "yes-or-no" question on the set of inputs. Intuitively, NP problems are those whose positive answer can be certified in polynomial time. Moreover, a problem Q (not necessarily a decision problem) is *NP-hard* if there exists a reduction for any problem Q' in NP to Q by means of a polynomial-time algorithm. For this, there must exist two polynomial algorithms, one to produce an instance I of Q from a given instance I' of Q', and another to produce a solution of I from a given solution of I'. Finally, a decision problem belongs to class *NP-complete* if it is both in NP and NP-hard.

Because the problem of whether class NP is equal to P is still unsolved, NP-completeness serves as a way to indicate that a problem can be reduced to the "hardest" problems of NP, the ones that most likely do not belong to P. Naturally, it also implies that, unless P=NP, there is no polynomial-time algorithm to solve the problem. For instance, Integer Programming (that we introduce in the following) is NP-hard (Papadimitriou (1981)). For some of the NP-complete or NP-hard problems, there exist resolution algorithms known as *pseudo-polynomial time algorithms*, in the sense that they are bounded by a polynomial in the input length and the magnitude of the largest number in the given instance. These problems are *NP-complete in the weak sense* or weakly NP-complete, and can be considered tractable as long as the instances do not contain "exponentially large" numbers. On the contrary, if no such algorithms can be developed unless P=NP, then the problems belong to the special case named *strongly NP-complete.* In the following section, we see that the pricing problems we tackle in this thesis are NP-complete in the strong sense. Furthermore, in Chapter 4, we derive an NP-completeness result for an assignment problem that appears as a subproblem of the problem at hand.

1.1.3 Integer Optimization

In the LP (1.1), all the variables are continuous. When all the variables are integer, we obtain an integer program (IP):

$$\begin{aligned} \max_{x} \quad & cx \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in \mathbb{Z}^n_+. \end{aligned} \tag{1.4}$$

If only a subset of the variables is integer, we obtain a mixed-integer (linear) program (MILP). The most common integer variables found in many applications (and throughout this thesis) are binary variables. The importance of Integer Programming lies in its adequacy to model a wide range of practical problems in fields such as telecommunications, facility location, production planning, scheduling, pricing, etc., and even in areas such as set theory, graph theory or number theory. We present here two methods that have been most successful in solving IPs and that will be used throughout this entire work, namely the branch-and-bound and branch-and-cut algorithms, along with some of the essential properties of integer programs. Benders decomposition for mixed-integer programs is addressed in an independent subsection. More extensive explanations on these topics can be found in Wolsey (1998); Nemhauser and Wolsey (1999); Conforti et al. (2014).

In an IP, the *feasible region* $S = \{x \in \mathbb{Z}^n_+ : Ax \leq b\}$ becomes a discrete set of points in $\mathbb{R}^n$, hence integer optimization is often called *combinatorial* or *discrete*. This may induce us to think that solving an integer problem is a simple task, when in reality it is not in general. Indeed, complete enumeration, although possible in theory, is not practical for medium and big-sized problems. Algorithms developed to solve integer problems usually take advantage on the tools available for linear problems and study the linear relaxation $\max\{cx : Ax \leq b, x \in \mathbb{R}^n_+\}$, which results from removing the integrality constraints on the variables of the IP. One widely-used algorithm that has been proved efficient for solving IPs and MILPs is the *branch-and-bound* algorithm. In broad terms, the branch-and-bound procedure searches for an optimal solution by iteratively *branching*, i.e. imposing linear constraints that partition the feasible region into subsets (and organizing the resulting problems as nodes in a tree). If the partition is made restricting the range of a variable (e.g. fixing a binary variable to either 0 or 1), then the strategy is known as variable branching. The enumeration tree is afterwards pruned using several methods. Apart from pruning by integrality or by infeasibility (when the resulting problem is integral or infeasible, resp.), it is essential the pruning by *bound*, when the optimal value of the subproblem is smaller than or equal to the best known lower bound on the value of the IP.

The ability to prune the tree is essential in the good performance of the branch-and-bound procedure, and the pruning process is based on the tightness of the bounds obtained. Good lower bounds can be obtained using heuristics. However, the formulation of the IP or the MILP plays an crucial role in obtaining tight upper bounds. Clearly, different integer formulations lead to linear relaxations with different feasible regions (all of which contain the same integer feasible points). In an ideal situation, the feasible region of the linear relaxation corresponds with the convex hull of S, $conv(S)$. In this case, all the extreme points have integer components, and thus solving the linear relaxation of the relaxed

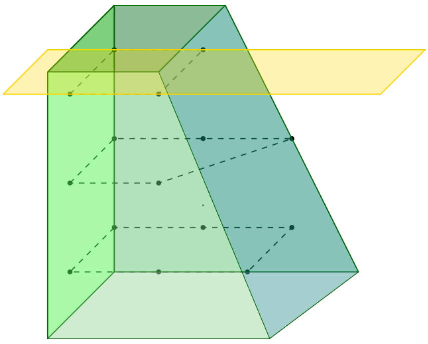

Figure 1.1: A linear relaxation polytope in $\mathbb{R}^3$ and a valid inequality

problem (which is found in an extreme point) leads to an integer solution which is also a solution to the IP. This is called a *perfect formulation*, and the feasible region obtained $conv(S)$ is an *integral polyhedron.* For instance, the integrality constraints of an IP can be relaxed when the matrix A is totally unimodular and the vector b is integer:

Definition 1.8 (Totally Unimodular matrix)**.** A matrix A is *totally unimodular* (TU) if every square submatrix of A has determinant +1,-1 or 0.

Proposition 1.9. *The linear program* $\{\max_x cx : Ax \leq b, x \in \mathbb{R}^n_+\}$ *has an integral optimal solution for all integer vectors* b *for which it has a finite optimal value if and only if* A *is TU.*

However, finding a description of the convex hull of the set of integer feasible points by means of inequalities is only ideal in theory, because the description often has such a large number of inequalities that is computationally intractable (for instance, in Balas and Pulleyblank (1983)). Besides, instead of computing the complete convex hull, it suffices to tighten the polyhedron in the neighborhood of an optimal solution in order to obtain stronger bounds that speed up the algorithm. A common way to tighten the polyhedron that results from the linear relaxation consists in adding valid inequalities to the formulation.

Definition 1.10 (Valid inequality)**.** An inequality $\pi x \leq \pi_0$ is *valid* for an IP in the form (1.4) if it is satisfied by every point of its feasible region S. We say that the inequality is *proper* when there exists $\bar{x} \in \mathbb{R}^n_+$ such that $A\bar{x} \leq b$ and $\pi\bar{x} > \pi_0$. A valid inequality that is violated by an optimal fractional solution x^* of the linear relaxation of the IP is a *cutting plane* separating x^* from S.

Proper valid inequalities are often called just valid inequalities, and when they originate a cutting plane they are also known as *cuts*. Figure 1.1 shows a set of feasible points of an IP (in black) and a green polytope from a linear relaxation of the IP. A valid inequality is shown in yellow. Explanations and examples on how to obtain valid inequalities for several problems can be found in Wolsey (1998).

Some valid inequalities are known as *general-purpose inequalities* because they can be derived for general configurations of IP programs. Such is the case of *Gomory's fractional cuts* (Gomory (1963)), other special types of *split inequalities* (equivalent to Gomory's cuts and to *mixed-integer rounding inequalities* Nemhauser and Wolsey (1990)) like *Chvátal inequalities* (Chvàtal (1973)) or *lift-and-project cuts* (Balas et al. (1993)), and *intersec-*

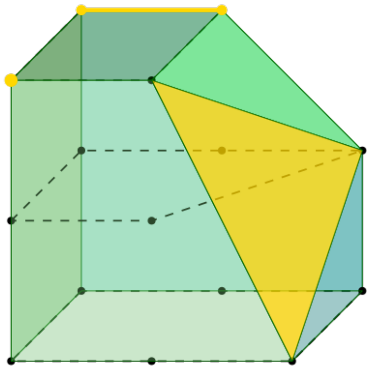

Figure 1.2: Convex hull of the feasible set of points of Figure 1.1 and three faces: vertex, edge and facet

tion cuts (Balas (1971, 1972)), among others. There also exist valid inequalities useful for problems with an underlying combinatorial structure, e.g. *cover* and *flow cover* inequalities that appear e.g. for the 0,1 knapsack problem or in IP formulations that model fixed charges (Conforti et al. (2014)). Other times, like in this dissertation, when we speak of valid cuts we refer to valid inequalities tailored to the specific problem. Many of these cuts are *strong* or *tight*, in the sense that they define nonempty faces or even facets of the convex hull of the feasible points, like the ones featured in Figure 1.2.

Definition 1.11 (Dominated inequality)**.** Let $\mathcal{B}$ be the polytope given by the convex hull of the set of feasible (integer) solutions S of (1.4). An inequality $\pi x \leq \pi_0$ is *dominated* by another inequality $\gamma x \leq \gamma_0$ when $\mathcal{B} \cap \{x : \gamma x \leq \gamma_0\} \subset \mathcal{B} \cap \{x : \pi x \leq \pi_0\}$. If an inequality is not dominated by any other, we say that it is *undominated* or *maximal*.

All undominated inequalities for S define a nonempty face of $\mathcal{B}$. The strongest maximal inequalities are those that induce facets of $\mathcal{B}$, i.e. faces of maximal dimension $n-1$ (of a full dimensional polyhedron, which we assume in what follows).

Definition 1.12 (Facet)**.** An inequality $\pi x \leq \pi_0$ is a *facet* of a polytope $\mathcal{B}$ of dimension n if:

1. *(validity)* every $x \in \mathcal{B}$ satisfies $\pi x \leq \pi_0$; and
2. *(maximality)* there exist n affinely independent points $x_i \in \mathcal{B}$ satisfying $\pi x_i = \pi_0$, $i = 1, \dots, n$.

One way to obtain tighter upper bounds in a branch-and-bound scheme consists in incorporating cutting planes in the tree nodes after the branching step, obtaining a *branch-and-cut* algorithm. When there are a small number of them, valid inequalities can be added as constraints to the original formulation or in the root node of the branching tree. However, as the number of them increases, a more sophisticated method is usually required to prevent running time to get out of control. A *separation procedure* is then a useful tool to determine which cuts to include at each step in order to obtain the best possible performance. The branch-and-cut algorithm is one of the most successful methods for solving integer programs to date (Conforti et al. (2014)).

Other techniques to derive formulations with a small integrality gap are *reformulation-linearization techniques* (RLT) (Sherali and Adams (1998)) and are based on a reformulation of the problem in a higher dimensional space (the *integrality gap* is the difference between the objective value of the linear relaxation of the problem and the optimal value). RLT are equally useful to derive strong valid inequalities. And *reformulation and relaxation techniques* are resolution techniques broadly used that take advantage of special structures arising when either the set of variables or the set of constraints can be partitioned into two subsets, one of them called a *complicating* subset. Some of the classical approaches include *Lagrangian relaxation* (Geoffrion (1974)), *Dantzig-Wolfe reformulation* (Dantzig and Wolfe (1960)) and *Benders decomposition* for mixed-integer programs (Benders (1962)). Other ways to fasten the resolution include *preprocessing techniques* to reduce the size of an instance (e.g. Escudero et al. (2009)) and *heuristic algorithms* to find good solutions of the primal and dual problems quickly (e.g. van Roy and Erlenkotter (1982)).

Benders decomposition

The Benders decomposition (Benders (1962)) is a decomposition approach widely used for solving mixed-integer problems. Consider the MILP

$$\begin{aligned} \max_{x,y} \quad & cx + hy \\ \text{s.t.} \quad & Ax + Gy \leq b, \\ & x \in \mathbb{Z}_+^{n_1}, \\ & y \in \mathbb{R}_+^{n_2}, \end{aligned} \tag{1.5}$$

with $c \in \mathbb{R}^{n_1}$, $h \in \mathbb{R}^{n_2}$, $A_{m\times n_1} = (a_{ij})$, $G_{m\times n_2} = (g_{ij})$, $b \in \mathbb{R}^m$. The first step in the Benders decomposition is to reformulate problem (1.5) into:

$$\begin{aligned} \max_{x} \quad & cx + z \\ \text{s.t.} \quad & z \leq \phi(x), \\ & x \in \mathbb{Z}_+^{n_1}, \\ & z \in \mathbb{R}_+, \end{aligned} \tag{1.6}$$

where $\phi(x)$ is defined as the optimal value of

$$\begin{aligned} \max_{y} \quad & hy \\ \text{s.t.} \quad & Gy \leq b - Ax, \\ & y \in \mathbb{R}_+^{n_2}. \end{aligned} \tag{1.7}$$

Problem 1.7 can be infeasible, unbounded or have a bounded optimal value. To determine it, we make use of the Minkowski-Weyl theorem to obtain a representation of the dual polyhedron in terms of its extreme points and extreme rays. The dual of (1.7) is:

$$\begin{aligned} \min_{u} \quad & u(b - Ax) \\ \text{s.t.} \quad & uG \geq h, \\ & u \in \mathbb{R}_+^{m}. \end{aligned} \tag{1.8}$$

Let $\{u^k\}_{\{k \in K\}}$ be the set of extreme points of the feasible set of the dual subproblem (1.8), $D := \{u \in \mathbb{R}^m_+ : uG \geq h\}$, and let $\{r^j\}_{\{j \in J\}}$ be the set of extreme rays of $D^0 := \{u \in \mathbb{R}^m_+ : uG \geq 0\}$.

Theorem 1.13 (Benders)**.** *Problem 1.6 can be reformulated as*

$$\max_x \quad cx + z \tag{1.9a}$$
$$s.t. \quad z \leq u^k(b - Ax) \text{ for all } k \in K, \tag{1.9b}$$
$$s.t. \quad r^j(b - Ax) \geq 0 \text{ for all } j \in J, \tag{1.9c}$$
$$x \in \mathbb{Z}^{n_1}_+, \tag{1.9d}$$
$$z \in \mathbb{R}. \tag{1.9e}$$

Model (1.9) is known as a *Benders reformulation* of problem (1.5) and is a mixed-integer reformulation with a very large (usually exponential) number of constraints. Therefore, the common approach, called *Benders decomposition*, consists in solving a relaxation of (1.9) where constraints from (1.9b) and (1.9c) *–Benders' cuts–* are added dynamically. The relaxed main problem is then known as the *master problem*, whereas the linear problem (1.7) solved to provide inequalities that strengthen the master problem is the *Benders subproblem*.

At each iteration, we solve the master problem with a subset $K_i \subseteq K$ of inequalities from (1.9b) and a subset $J_i \subseteq J$ of inequalities from (1.9c). The solution (x^*, z^*) obtained is used to solve the Benders subproblem (or, equivalently, its dual). If $D = \emptyset$, then the Benders subproblem is either infeasible or unbounded, and thus the master problem is either infeasible or unbounded as well. Otherwise, $D \neq \emptyset$ and the Benders subproblem is infeasible or has a bounded optimal value $cx^* + z^*$. If it is infeasible, then by duality theory (1.8) is unbounded. In this case, there exists an extreme ray r^j, $j \in J \setminus J_i$ such that $r^j(b - Ax^*) < 0$, so we can add the *feasibility cut* $r^j(b - Ax) \geq 0$ to the master problem. Alternatively, by strong duality of linear problems the optimal value $\phi(x^*)$ of the Benders problem is equal to the optimal value of its dual. Thus, we solve (1.8) to obtain an optimal value $\phi(x^*) = u^k(b - Ax^*)$, with $k \in K$. If $\phi(x^*) < z^*$, then $k \in K \setminus K_i$ and we add the *optimality cut* $z \leq u^k(b - Ax)$ to the master. Finally, if $\phi(x^*) = z^*$ then (x^*, y^*) is an optimal solution of (1.5), where y^* is optimal for the Benders subproblem.

Although not always (see e.g. Fischetti et al. (2016)), the advantage of Benders decomposition in many cases relies on the fact that the Benders subproblem is *separable*, i.e. can be decomposed into many Benders subproblems of smaller size. Decisive issues are the separation procedure developed to select Benders' cuts (researched in Magnanti and Wong (1981); Fischetti et al. (2010), among others) and the overall resolution algorithm used to incorporate them and solve the master problem. The interested reader can consult Nemhauser and Wolsey (1999); Conforti et al. (2014) for a more extensive review on the topic. The details of our implementation of Benders decomposition are described in Chapter 3.

1.1.4 Bilevel Optimization

Bilevel Programming problems are defined as mathematical optimization problems with a hierarchical structure, where a first level player or leader wishes to optimize his problem knowing that the follower or second level player will react optimally to it. The leader is assumed to have complete knowledge of the follower problem, whereas the follower observes the decisions of the leader and selfishly optimizes his own strategy. Formally, bilevel problems are optimization problems where the constraints of a first optimization problem translate the fact that some of the variables are optimal solutions to a nested optimization problem. The first level is given by the first objective function and its constraints; and the nested optimization problem, characterized by its own objective function and constraints, captures the problem of the second level. Let $c, c' \in \mathbb{R}^{n_1}$, $h, h' \in \mathbb{R}^{n_2}$, $A_{m\times n_1} = (a_{ij})$, $a_{ij} \in \mathbb{R}\ \forall i, j$, $G_{m\times n_2} = (g_{ij})$, $g_{ij} \in \mathbb{R}\ \forall i, j$, $b, b' \in \mathbb{R}^m$, $x \in \mathbb{R}^{n_1}_+$, $y \in \mathbb{R}^{n_2}_+$. The general formulation of a linear bilevel problem is:

$$\max_{x\in X} \quad cx + hy \tag{1.10a}$$
$$\text{s.t.} \quad Ax + Gy \leq b, \tag{1.10b}$$
$$y \in \arg\max_{y\in Y} \quad c'x + h'y, \tag{1.10c}$$
$$\text{s.t.} \quad A'x + G'y \leq b'. \tag{1.10d}$$

(1.10a) and (1.10b) constitute the objective function and constraints of the leader problem, whereas (1.10c) and (1.10d) are the objective function and constraints of the follower problem. $X \in \mathbb{R}^{n_1}$ (resp. $Y \in \mathbb{R}^{n_2}$) represents the set of constraints related only to the x-variables (resp. y-variables), such as upper or lower bounds, or integrality constraints for the linear integer and mixed-integer bilevel programs.

If we remove constraint (1.10c), we obtain the relaxed problem associated with (1.10). Thus, the *constraint region* is $\Omega = \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : x \in X, y \in Y, Ax + Gy \leq b, A'x + G'y \leq b'\}$. For a given vector $\bar{x} \in X$, the *lower-level feasible set* is defined by $\Omega(\bar{x}) = \{y \in Y : A'\bar{x} + G'y \leq b'\}$, and the *lower-level (rational) reaction set* is $R(\bar{x}) = \{y \in Y : y \in \arg\max \{c'\bar{x} + h'\hat{y} : \hat{y} \in \Omega(\bar{x})\}\}$.

Every $y \in R(\bar{x})$ is known as a rational response, and the second level problem must be solvable for global minima. Much research has been devoted to the case where it is convex. Furthermore, R is a so-called point-to-set mapping from $\mathbb{R}^{n_1}$ into the power set of $\mathbb{R}^{n_2}$ denoted by $R : \mathbb{R}^{n_1} \to 2^{\mathbb{R}^{n_2}}$. We denote some element of $R(\bar{x})$ by $y(\bar{x})$.

Lastly, the *induced region* $IR = \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} : x \in X, Ax + Gy \leq b, y \in R(x)\}$ corresponds to the feasible set of the leader. This set is usually nonconvex and it can even be disconnected or empty.

Bilevel Programming problems are difficult to solve, mainly because they are in general non convex and non differentiable. Even the linear setting (with linear objective functions and constraints) is difficult (see Jeroslow (1985) for the NP-hardness result). Due to these intrinsic difficulties, current resolution methods focus on particular cases easier to solve. The majority of the resolution algorithms are based on the reformulation of the bilevel problem into a single-level problem by various methods as a starting point, followed by the determination of necessary and sufficient optimality conditions and solution algorithms. One of the most used methods, applicable in the optimistic position, is called the Kuhn-Tucker approach. It consists in replacing the lower level problem by its Karush-Kunh-Tucker conditions, resulting in a single-level reformulation. The resulting problem is a mathematical program with equilibrium constraints. Then a branch-and-bound strategy is applied to deal with the complementarity constraints. Some references that have proposed necessary and sufficient optimality conditions are those by Bard (1984); Ishizuka (1988); Dempe (1992a, 1992b); Outrata (1993); Savard and Gauvin (1994); Vicente and Calamai (1995); Dempe et al. (2010); Dempe et al. (2014); Aboussoror and Adly (2018). The approach we follow in Chapter 2 begins by characterizing optimality of the lower-level by means of duality theory. Then we use integer programming techniques to manipulate the complementarity constraints that arise, obtaining a single-level mixed-integer linear formulation. This approach has been used in the previous literature for instance in Paruchuri et al. (2008), García-Herreros et al. (2016).

Optimistic versus pessimistic position of bilevel programming

If the optimal solution of the lower-level problem is not uniquely determined, i.e. $R(\bar{x})$ consists of at least two points for some $\bar{x} \in \mathbb{R}^{n_1}$, there exist at least two different modeling approaches to attack a bilevel problem: the optimistic and the pessimistic. In the optimistic setting, the follower selects, among all optimal solutions, the one that is best suited by the leader. Formally, in optimistic bilevel programming it is assumed that, if the reaction set $R(x)$ is not a singleton, the leader is allowed to select the element $y \in \Omega(x)$ that is most beneficial for him.

Definition 1.14 (Optimistic solution)**.** A point $(x^*, y^*) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ is an *optimistic solution* for problem (1.10) if

$$\begin{aligned}
&x^* \in X,\\
&Ax^* + Gy^* \leq b,\\
&y^* \in R(x^*),\\
&cx^* + hy^* \geq cx^* + hy \qquad \forall y \in R(x^*),
\end{aligned}$$

and $\phi_o(x^*) \geq \phi_o(x)\ \forall x \in X$, where $\phi_o(x) = \max_y\{cx + hy : y \in R(x)\}$.

In the pessimistic case, however, the follower can select any optimal solution, so the approach of the leader is forced to trying to limit the damage the follower can make in the worst-case scenario (with respect to the leader). In this way, the optimistic strategy corresponds with a cooperative behavior between the leader and the follower, whereas the pessimistic approach reflects an aggressive follower behavior.

Definition 1.15 (Pessimistic solution)**.** A point $(x^*, y^*) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ is a *pessimistic solution* for (1.10) if

$$\begin{aligned}
&x^* \in X,\\
&Ax^* + Gy^* \leq b,\\
&y^* \in R(x^*),\\
&cx^* + hy^* \leq cx^* + hy \qquad \forall y \in R(x^*),
\end{aligned}$$

and $\phi_p(x^*) \leq \phi_p(x)\ \forall x \in X$, where $\phi_p(x) = \min_y\{cx + hy : y \in R(x)\}$.

As already pointed out in Bialas and Karwan (1984), the optimistic position can occur under suitable assumptions like, for instance, when the follower participates in the profit of the leader. In the literature, most of the results we find so far are devoted to the optimistic approach. Besides, in the optimistic case the existence of solutions is guaranteed when the constraint region Ω is non empty. This is in general not true for the pessimistic case:

an example of a pessimistic bilevel problem without optimal solution can be found in Dempe (2002), which is a useful resource to expand on the topic of pessimistic bilevel programming. In Chapters 2 and 4 of this dissertation, the bilevel problems studied are viewed from an optimistic position; and in Chapter 3, the second level players choose a pessimistic solution in case of multiple optimal solutions.

1.2 The Rank Pricing Problem

The fundamental problem that will be studied in this thesis is the Rank Pricing Problem (RPP). In the RPP, a company is interested in setting the price of a series of products so as to maximize its revenue. Customers are endowed with a positive budget and are unit-demand, i.e. they are interested in a subset of the products offered by the company, but intend to buy at most one of them. To do so, they rank the subset of products they find relevant (this ranking is strict). Once the company has established the prices of the products, each customer purchases his most desired product (the highest-ranked one) below his budget. The supply of products is assumed to be unlimited, or high enough so that customers can always purchase what they prefer. This setting is consistent with companies selling a product line or different different versions of the same products that differ in their features, so typically customers only intend to buy one of them. Electrical appliances (like dishwashers or irons), electronic devices (like laptops or mobile phones) or digital goods (e-books, digital subscriptions, templates) are a few examples of this type of products. In the RPP, no cost is associated to the production, so a negligible cost of production is assumed.

Throughout the thesis, two new variants of the RPP that have not been considered in previous literature are proposed. The Rank Pricing Problem with Ties (RPPT) is a generalization that features customers whose ranked list of products can include ties. Such ties represent the indifference of a customer regarding the characteristics of two (or more) products. If a customer can afford several products and is indifferent between them, naturally his purchase decision is based on the price: he purchases the least expensive product. The second generalization is called the Capacitated Rank Pricing Problem because it includes a capacity on the number of copies of each product for sale, i.e. a limited supply of products. In this extension, reservation prices for each customer and product in his ranked list are considered instead of the positive budgets.

1.2.1 State of the art

The RPP as stated here was introduced in Rusmevichientong et al. (2006). The motivation behind it was the availability of online customer data collected through a choice advisory website developed by General Motors (GM). This website recommended vehicles to customers based on their preferences regarding the features of the vehicles, and their budgets. The system then recorded several data from the customers: a budget (explicitly specified by the customer) and the resulting list of recommended vehicles for each customer, sorted according to the degree to which each vehicle fulfills the client's requirements. As a result, in the Rank Pricing consumer choice model proposed each customer is characterized by a budget, a ranked list of products and a choice rule that leverages this ranking-list structure of the data in the previously described manner. The objective is the optimization of GM's vehicle prices. They show that the RPP is NP-complete in the strong sense and introduce a heuristic approximation algorithm with performance bounds when the prices are constrained by a price ladder (an ordering of prices defined beforehand). They then apply their algorithms to the real data set. Their analysis provides insights into the current pricing strategy and suggests improvements that may increase the profit of the firm. Rusmevichientong et al. (2006) introduce an additional problem to the RPP with the same profit-maximization objective but a different customer choice mechanism. According to the Min Pricing choice model, the customer chooses the least expensive product from his list that meets the budget constraints (without taking into account any order). In Rusmevichientong (2003) there is a third consumer choice model, the Max Pricing, where customers are assumed to purchase the most expensive product under their budget. The resulting problems are called the Min Pricing problem and the Max Pricing problem.

Aggarwal et al. (2004) study a modification of the RPP that includes a price-ladder constraint (an ordering on prices defined beforehand) and reservation prices for each customer-product pair (instead of a uniform budget per customer). These reservation prices reflect how much they are willing and able to spend on each product, commonly known as their willingness to pay (WTP). They present a polynomial time approximation scheme (PTAS) and a 4-approximation algorithm for this problem when the reservation prices are consistent, i.e. higher-ranked products are assigned non-decreasing reservation prices (a uniform budget is a particular case of consistent reservation prices). They first show that the RPP with price-ladder can be reduced to the Max Pricing problem, and then introduce a PTAS for the latter problem. The 4-approximation algorithm is developed for the Max Pricing problem with a price-ladder constraint assuming a limited number of copies of each product. Briest and Krysta (2006) take up the work in Aggarwal et al. (2004) and prove that both algorithms are close to best possible. They also analyze the hardness of approximation of a variety of unit-demand pricing models under different assumptions on the selection rules, the capacity of the supply and the prices of the products. They consider the three problems described, namely the RPP, the Min Pricing and the Max Pricing. Briest (2008) considers the Min Pricing problem with unlimited supply on the uniform-budget case, and derive hardness-approximability results.

1.2.2 Related pricing problems

The rise of e-commerce in the 2000s decade resulted in practical applications for algorithmic pricing, a branch of pricing problems where the customers' choice rule is assumed to be known by the company. This led to the proliferation of pricing optimization problems that featured different customer choice behavior models. Within the reservation price framework, where the customers choices are entirely based on their reservation prices, a widely-used choice rule is the maximization of the customers utility. Utility-based models assume that customers have a reservation price for each product of their interest. Thus, once the pricing strategy is established, customers purchase the product that maximizes their utility, which is the difference between the price of the product and their reservation price (customers only purchase if the utility is non negative). In this sense, utility-based models assume that customers follow compensatory (or rational) decision processes, where high levels on some attributes of a product (such as the price) can compensate for low levels on other aspects. In contrast, ranking-based models in the pricing context can accommodate to customers following both compensatory and non-compensatory decision processes (like a lexicographic rule), as long as the price of the products is not considered as an attribute when taking decisions.

From the extensive literature, we cite a few research works on pricing problems with utility-based customers that include exact models. Dobson and Kalish (1988) tackle the problem of positioning and pricing a product line with two different objectives, maximizing the total profit (for firms oriented to pricing) and maximizing the total (also called maximum, buyer) welfare (for a non-profit organization). They give mathematical formulations and describe new heuristics algorithms. Dobson and Kalish (1993) show that the maximum welfare is equivalent to the uncapacitated plant location problem (PLP) and develop a heuristic for the total profit that outperforms (in time) the one in Dobson and Kalish (1988). Shioda et al. (2011) formulate the product pricing problem with utility-maximizing customers as a mixed-integer programming problem, also designing heuristics and yielding valid cuts. Heilporn et al. (2010) and Fernandes et al. (2016) study the connection between the utility-maximization problem and different Network Pricing Problems (NPP) with connected toll arcs (where the set of tolls constitutes a path). Myklebust et al. (2016) take on the work presented in Dobson and Kalish (1988, 1993) and propose improvements to their heuristics based on a study of the structure of the underlying mixed-integer programming problem.

In the case of a limited supply of products, Guruswami et al. (2005) study the problem of pricing in the utility-maximization context while being envy-free regarding the customers' valuation/reservation price for each product. In the limited-supply setting, the envy-freeness is a fairness criterion which guarantees that customers always purchase the product that maximizes their preferences among the ones below their reservation prices. When there is unlimited supply, the company can always serve customers and therefore they purchase according to their selection rule, so any pricing is envy-free. Guruswami et al. (2005) introduced this problem and derived logarithmic approximations (in the number of customers) for two important customer profiles: unit-demand customers and single-minded customers. Single-minded customers are only interested in one bundle offered by the company, and they will purchase it if their reservation price is below the price of the sum of products of the bundle. Single-minded customers have been thoroughly studied in auction mechanism design. In contrast with algorithmic pricing, the mechanism design framework aims at determining the best strategy when the agents implicated hold their information privately. In this context, Goldberg and Hartline (2001) and Goldberg et al. (2006) relate auction mechanism design with algorithmic pricing. They do so by introducing the notion of competitive auctions, that achieve a profit that is a constant fraction of optimal on every input. These auctions are truthful, so the best strategy for the bidders amounts to truthfully revealing their private valuations and can be found using algorithmic pricing. Goldberg and Hartline (2001) extend their results to the case of multiple items, stating the optimal pricing problem as an integer programming problem. Balcan et al. (2008) and Balcan et al. (2005) apply these results to the problem of auctioning a digital good and to the problem of item-pricing in unlimited-supply combinatorial auctions. In algorithmic pricing problems, following the research of Guruswami et al. (2005), Hartline and Koltun (2005) develop efficient approximation algorithms for the unit-demand and single-minded customer settings in the unlimited-supply context, and the single-minded envy-free setting with limited supply of products. Khandekar et al. (2009) give stronger hardness of approximation bounds for this problem when the buyers are interested in subsets of size at most two. A recent paper that includes mixed-integer linear formulations for the product pricing problem with single-minded customers is Bucarey et al. (2021). Figure 1.3 shows a diagram that summarizes the contributions made in the different pricing problems with utility-based customers presented.

1.2.3 Other optimization problems with ranking-based customer preferences

Pricing optimization problems in combination with ranking-based customer preferences are scarce in the literature. However, the modelization of the customers selection rule by means of a ranked list of preferences appears in many related fields. One close problem is the Product Line Design (PLD). This problem aims at selecting a subset of products to be produced (generally from a bigger given set) in order to maximize the company's revenue. The PLD problem has been traditionally studied under two different customer choice rules. In the probabilistic choice behavior, each customer (or type of customer) probabilistically chooses from the available options. Some references are those by Chen and Hausman (2000), Schön (2010a, 2010b) and Kraus and Yano (2003), among others. In the first-choice rule, customers deterministically select the product from the offered line that maximizes their utility. Some references are those by Green and Krieger (1985, 1993), McBride and Zufryden (1988), Dobson and Kalish (1988) and Belloni et al. (2008). A very recent work by Bertsimas and Mišić (2019) studies the PLD problem under a ranking-based customer choice model. It introduces a new mixed-integer formulation, theoretically analyzes it, and presents a solution approach based on Benders decomposition that significantly outperforms the previous results.

Another related field is Discrete Location, a branch focused on finding the location of establishments that best satisfies the demand of customers. In particular, Hanjoul and Peeters (1987) introduce the Simple (or Uncapacitated) Plant Location Problem with Order, in which a firm wants to select the number and places of a series of facilities to open so as to maximize the revenue, and the clients to be allocated have a ranking on the list of potential sites. Hansen et al. (2004) and Cánovas et al. (2007) build up on the problem presented in Hanjoul and Peeters (1987), the first ones deriving formulations from the bilevel perspective and the second ones introducing some valid inequalities as well as a very effective preprocessing, along with a computational study to show the efficiency of their approach. Hemmati and Smith (2016) relate multi-product pricing, facility location and bilevel optimization. These authors propose a mixed-integer bilevel programming approach for a competitive prioritized set covering problem. This model can be applied to the introduction of new products in a competitive market and to the competitive facility location problem. In both cases each customer has an ordered product (facility) preference list which represents the relative utility of each product (facility). Other works deal with a particular ordering of the facilities through the concept of closest assignment. Espejo et al. (2012) give a thorough review and comparison of the different closest assignment constraints encountered in the location literature, and study their generalization in the case of ties between distances.

Although the optimality criteria differ from the pricing literature, bipartite matching problems with preferences also model the customers' choice by means of a ranked list

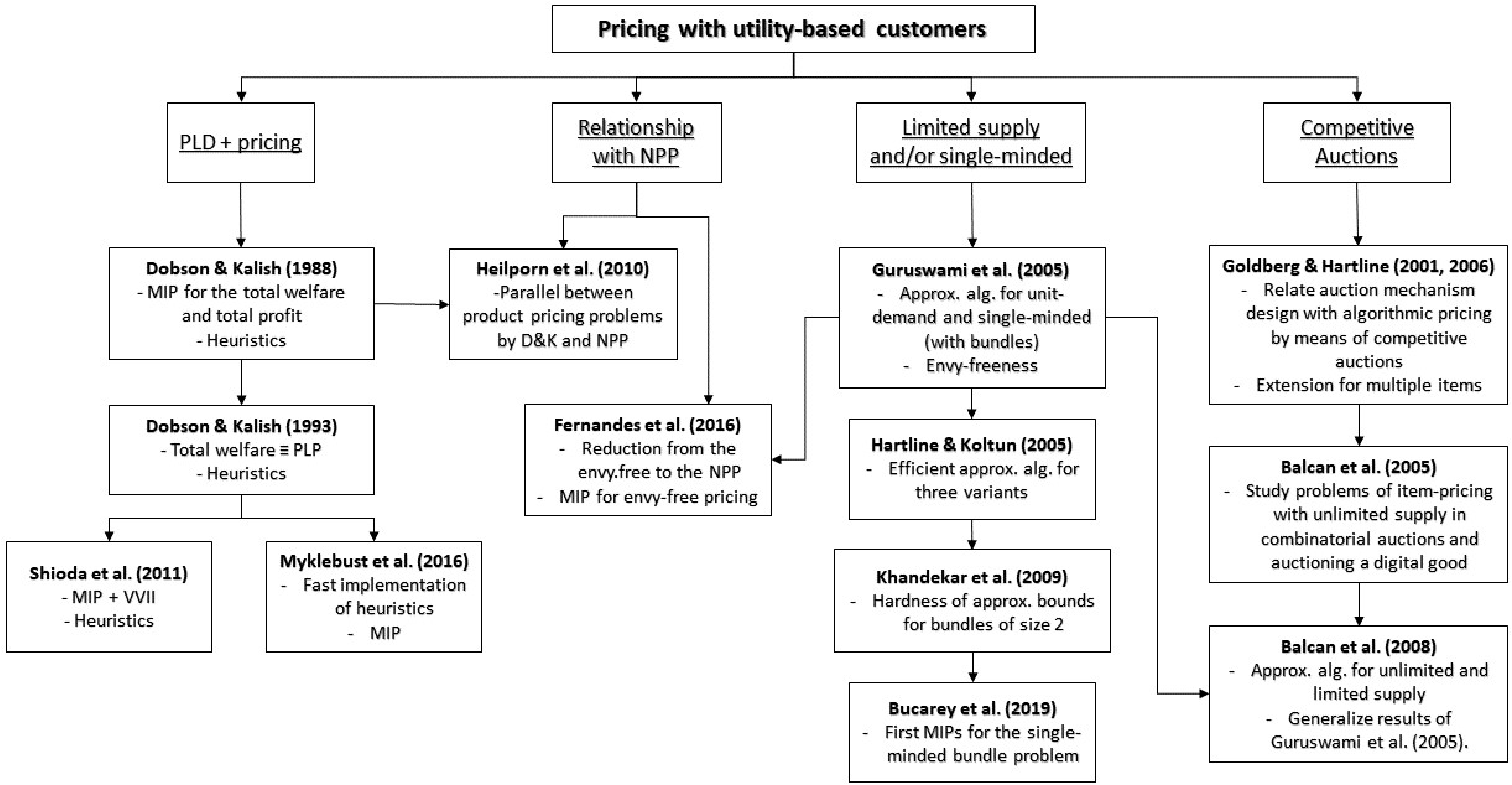

Figure 1.3: Contributions to pricing problems with utility-maximizing customers and related problems

of preferences. In particular, very well-known problems like the Stable Marriage (SM) problem include preference lists as the agents' choice. The first integer formulations were introduced by Vande Vate (1989) and by Gusfield and Irving (1989). An extension of Vande Vate's model to include incomplete lists of preferences was given by Rothblum (1992). More recently, extensions of these models have been introduced by Kwanashie and Manlove (2014) and Delorme et al. (2019) to tackle a one-to-many generalization of the SM problem, namely the Hospital-Residents (HR) problem, as well as the Stable Marriage with Ties (SMT) and the Hospital-Residents with Ties (HRT) generalizations. An in-depth review on structural and algorithmic results on matching problems with preferences can be found in Manlove (2013).

Chapter 2

The Rank Pricing Problem

In this chapter, we focus on the Rank Pricing Problem as stated in Rusmevichientong et al. (2006), that is, a ranking-based pricing model with unit-demand customers with a positive budget and unlimited supply of products. For a detailed literature review on the RPP, see Chapter 1. The previous literature focuses on complexity results and approximation algorithms. To the best of our knowledge, no exact optimization models have been proposed in the literature to deal with it so far. To address the RPP, we present a nonlinear bilevel formulation in which the company acts as a leader and determines the prices of the products. Once the prices are fixed, each customer, which acts as a follower, solves his own optimization problem. Besides, a non-linear single level formulation is proposed, based on the fact that a customer purchases the highest-ranked product among all the products he can afford. We linearize the formulations by means of two types of auxiliary variables and derive new valid inequalities. These inequalities are separated and included into the models through the development of a branch-and-cut algorithm. We also take advantage of the fact that some of its constraints constitute a special case of the Set Packing Problem and other properties of the problem in order to strengthen the formulations. We develop some preprocessing techniques to be applied to the instances before solving them. Finally, we present the results of our computational analysis, in which we compare the formulations and show that the results obtained reduce the computational effort when obtaining optimal solutions.

The remainder of the chapter is organized as follows. Section 2.1 is devoted to a bilevel formulation for the RPP; in Section 2.2, the RPP is formulated directly as a single level nonlinear model; Section 2.3 includes two linearizations that apply to both models and the development of other valid inequalities to strengthen their linear relaxations through the implementation of branch-and-cut algorithms; in Section 2.4 some families of clique inequalities associated to a subset of constraints are studied attending to the formulations; Section 2.5 includes some preprocessing results; and Section 2.6 is devoted to testing the performance of the models by means of a computational study.

2.1 Notation and bilevel formulation

Let $K = \{1, \ldots, |K|\}$ be the set of customers and $I = \{1, \ldots, |I|\}$ the set of products. Each customer $k \in K$ has a subset of *acceptable* products $I^k \subseteq I$ so that k would rather not make any purchase than buy a product $i \notin I^k$. Similarly, we say that a customer k is acceptable for a product i if customer k is a potential buyer of product i. The set of acceptable customers for i is $K_i := \{k \in K : i \in I^k\}$. Without loss of generality, we assume that customers are interested in at least one product from the company, i.e. $I^k \neq \emptyset \; \forall k \in K$, and that each product is acceptable for at least one customer, that is, $K_i \neq \emptyset \; \forall i \in I$. Otherwise, the customer and/or the product can be removed from the optimization process.

Every customer k makes a ranked list of the products in I^k according to his preferences, usually named his *list of preferences*. To describe this ranking in the bilevel formulation, we set a *preference value* $s_i^k > 0$ for each customer $k \in K$ and each of his acceptable products $i \in I^k$, where $s_i^k > s_j^k$ if customer k prefers product i over product j. We also assume that for each customer all preferences are strict, so that he never likes two products the same.

Finally, each customer k is endowed with a fixed budget. In order to keep notation consistent in the formulation, and given that different customers may have the same budget, we define set $M = \{1, \ldots, |M|\}$ as the set of indices that refer to the different budgets of the customers, and $B = (b^m)_{m \in M}$ as the (ordered) set of different budgets, so that $b^{m_1} < b^{m_2}$ if $m_1 < m_2$. To describe a customer's budget, we define a function $\sigma : K \to M$ such that $\sigma(k) = m$ if the budget of customer k is the m-th smallest budget b^m. We say that a customer k_1 is *richer* than k_2 if $\sigma(k_1) > \sigma(k_2)$, and the *richest* customers are those whose budget is $b^{|M|}$. Since it will be useful in following sections, we set $b^0 = 0$.

The RPP aims at establishing the prices of a set of products sold by a company so as to maximize its revenue, the sum of the prices of all items sold. Each customer purchases his most preferred product among the ones he can afford. Note that if a customer cannot afford any product, he will not purchase anything. Therefore, the company, acting as the upper level decision maker, decides on the prices of the products, $p_i \geq 0$, $i \in I$. At the lower level of the hierarchy, the customers decide which product to purchase. For this purpose, we introduce binary variables x_i^k, $k \in K$, $i \in I^k$, for every customer's purchase decision, that is, $x_i^k = 1$ if and only if customer k buys product i. The bilevel formulation for the RPP is:

$$(\text{BNLM}^p) \qquad \max_{p} \quad \sum_{k \in K} \sum_{i \in I^k} p_i x_i^k \tag{2.1a}$$

$$\text{s.t.} \quad p_i \geq 0 \quad \forall i \in I, \tag{2.1b}$$

where $\forall k \in K$, x^k is an optimal solution of

$$\max_{x^k} \quad \sum_{i \in I^k} s_i^k x_i^k \tag{2.1c}$$

$$\text{s.t.} \quad \sum_{i \in I^k} x_i^k \leq 1, \tag{2.1d}$$

$$\sum_{i \in I^k} p_i x_i^k \leq b^{\sigma(k)}, \tag{2.1e}$$

$$x_i^k \in \{0, 1\} \quad \forall i \in I^k, \tag{2.1f}$$

where constraint (2.1d) forces customer k to buy one product or none and (2.1e) establishes that customer k only buys a product if he can afford it. (BNLM^p) is a nonlinear bilevel problem with multiple independent followers. Notice that the unlimited supply assumption guarantees that each customer solves a problem involving only the upper level variables and his own variables, thus they are independent followers.

The following result proves that (BNLM^p) is well-posed, in the sense that the lower level problem has a unique optimal solution for a given vector of prices.

Proposition 2.1. *The lower level optimization problems of formulation* (BNLM^p) *have a unique optimal solution.*

Proof. The objective function of the lower level problem of (BNLM^p) for a given customer k is $\sum_{i \in I^k} s_i^k x_i^k$, with coefficients $s_i^k > 0$, $s_i^k \neq s_j^k$ $\forall i, j \in I^k$, $i \neq j$. Constraints (2.1d) ensure that at most one x-variable can take value one. If $p_i > b^{\sigma(k)}$ $\forall i \in I^k$, then the optimal solution is given by $x_i^k = 0$ $\forall i \in I^k$. Otherwise, the optimal solution is $x_i^k = 1$ for the unique product i such that $s_i^k = \max\left\{s_j^k : j \in I^k, p_j \leq b^{\sigma(k)}\right\}$, $x_j^k = 0$ for all $j \in I^k : j \neq i$. $\square$

It is worth noticing that, although we have focused on the unit-demand case, this formulation and the following ones also apply if a customer k is interested in purchasing d^k copies of the same product and his budget represents the maximum amount he is willing to pay per copy. Indeed, without loss of generality, it suffices to replace the customer with d^k customers with such budget and the same list of preferences. Alternatively, we can replace the objective function by $\sum_{k \in K} d^k \sum_{i \in I^k} p_i x_i^k$.

The following illustrative example facilitates the understanding of the RPP.

Example 2.2. *Table 2.1 shows the preference matrix and the vector of budgets of an instance of the RPP with 10 customers and 5 products. Each entry of the matrix shows the pair (t_i^k, s_i^k), where t_i^k is the ranking that the customer has over the product, and s_i^k equals the associated preference value. If product i is the most preferred product for customer k, then $t_i^k = 1$ and $s_i^k = |I| = 5$; if j is the second most preferred product for k, then the entry in the matrix is $(2, 4)$,* et cetera. *In this example, $|M| = 7$ and $b^1 = 16$, $b^2 = 25$, $b^3 = 31$, ..., $b^7 = 53$. Furthermore, for instance, for customer $k = 7$, $I^7 = \{1, 2, 3, 5\}$, and $\sigma(7) = 2$ because he has the second lowest budget. After solving this RPP, we obtain that an optimal solution is provided setting the prices indicated in the last row of Table 2.1. Taking into account these prices and the preferences, the customers purchase the product whose preference is marked with an asterisk in the preference matrix. For instance, customer 4 can only afford products 2, 4 and 5, and he purchases product 5 (for less than his budget) because it is his preferred one among them; whereas customer 7 purchases product 2, his least preferred one, because it is the only one in his list of preferences that he can afford.*

The fact, already observed in Rusmevichientong et al. (2006), that an optimal solution of (BNLM^p) exists such that $p_i \in B$ $\forall i \in I$, suggests us to define new binary variables

	Product 1	Product 2	Product 3	Product 4	Product 5	Budgets
Customer 1	-	(3,3)	(1,5)*	-	(2,4)	53
Customer 2	(2,4)*	-	(1,5)	-	-	40
Customer 3	(1,5)*	-	(2,4)	(4,2)	(3,3)	40
Customer 4	(4,2)	(3,3)	(5,1)	(2,4)	(1,5)*	38
Customer 5	(1,5)	-	(3,3)	-	(2,4)*	32
Customer 6	(4,2)	(3,3)	(1,5)	(5,1)	(2,4)*	31
Customer 7	(1,5)	(4,2)*	(2,4)	-	(3,3)	25
Customer 8	(1,5)	-	(3,3)	(2,4)*	-	25
Customer 9	-	-	(2,4)	(1,5)*	-	25
Customer 10	(2,4)	(1,5)*	-	(4,2)	(3,3)	16
Optimal prices	40	16	53	25	31	

Table 2.1: Preference matrix, vector of budgets and an optimal solution to an instance of the RPP with 10 customers and 5 products

v_i^m, $i \in I$, $m \in M$ representing the candidate prices of products, that is, $v_i^m = 1$ if and only if the price of product i is set to b^m in a feasible solution. Since each product i has only one price, only one binary variable v_i^m can take value 1 $\forall i \in I$. Therefore, the price of product i can be expressed as $p_i = \sum_{m \in M} b^m v_i^m$. Furthermore, the set of indices m of candidate prices b^m at which k can afford i is defined as $M^k = \{1, \ldots, \sigma(k)\}$.

We can now reformulate the problem replacing p_i variables by v_i^m variables, replacing constraints (2.1b) with the following constraints in order to ensure products have at most one price:

$$\sum_{m \in M} v_i^m \leq 1 \quad \forall i \in I, \tag{2.2a}$$

$$v_i^m \in \{0,1\} \quad \forall i \in I, m \in M, \tag{2.2b}$$

and replacing constraints (2.1e) of the lower level problem with

$$x_i^k \leq \sum_{m \in M^k} v_i^m \quad \forall k \in K, i \in I^k. \tag{2.2c}$$

The resulting bilevel formulation is:

$$(\text{BNLM}^v) \qquad \max_{v} \quad \sum_{k \in K} \sum_{i \in I^k} \left(\sum_{m \in M^k} b^m v_i^m \right) x_i^k \tag{2.3a}$$

$$\text{s.t.} \quad \sum_{m \in M} v_i^m \leq 1 \quad \forall i \in I, \tag{2.3b}$$

$$v_i^m \in \{0,1\} \quad \forall i \in I, m \in M, \tag{2.3c}$$

where $\forall k \in K$, x^k is an optimal solution of

$$\max_{x^k} \quad \sum_{i \in I^k} s_i^k x_i^k \tag{2.3d}$$

$$\text{s.t.} \quad \sum_{i \in I^k} x_i^k \leq 1, \tag{2.3e}$$

$$x_i^k \leq \sum_{m \in M^k} v_i^m \quad \forall k \in K, i \in I^k, \tag{2.3f}$$

$$x_i^k \in \{0, 1\} \quad \forall i \in I^k. \tag{2.3g}$$

Now, we focus on the lower level problem of a customer k for a fixed integer vector of prices v. The fact that the matrix corresponding to the feasible set of each lower level problem of (BNLM^v) is TU and the RHS of the constraints are integer enables us to relax the integrality constraints (2.1f) (see Chapter 1). The lower level problem can be further simplified taking into account that once the leader variables v_i^m are known, a subset of x-variables is determined. If we consider the subset $I^{k*} = \{i \in I^k : \sum_{m \in M^k} v_i^m = 1\}$, variables $\{x_i^k, k \in K, i \in I^k \setminus I^{k*}\}$ are automatically settled to 0 since customer k cannot afford these products. Hence, constraints (2.2c) can be eliminated and the lower level problem can be formulated as

$$\begin{aligned} \max_{x^k} \quad & \sum_{i \in I^{k*}} s_i^k x_i^k \\ \text{s.t.} \quad & \sum_{i \in I^{k*}} x_i^k \leq 1, \\ & x_i^k \geq 0 \quad i \in I^{k*}. \end{aligned}$$

For each customer k, the dual problem of the lower level problem is

$$\begin{aligned} \min_{u^k} \quad & u^k \\ \text{s.t.} \quad & u^k \geq s_i^k \quad i \in I^{k*}, \\ & u^k \geq 0. \end{aligned}$$

By duality theory, x^k and u^k are optimal solutions to the primal and dual problems, respectively, if and only if

$$\begin{aligned} \sum_{i \in I^{k*}} s_i^k x_i^k = u^k, \\ \sum_{i \in I^{k*}} x_i^k \leq 1, \\ u^k \geq s^k \quad \forall i \in I^{k*}, \\ x_i^k, u^k \geq 0. \end{aligned}$$

Thus, the resultant formulation after substitution of u^k is

$$\text{(BNLM)} \qquad \max_{v,x} \quad \sum_{k \in K} \sum_{i \in I^k} \left(\sum_{m \in M^k} b^m v_i^m \right) x_i^k \tag{2.4a}$$

$$\text{s.t.} \quad \sum_{m \in M} v_i^m \leq 1 \quad \forall i \in I, \tag{2.4b}$$

$$\sum_{i \in I^k} x_i^k \leq 1 \quad \forall k \in K, \tag{2.4c}$$

$$x_i^k \leq \sum_{m \in M^k} v_i^m \quad \forall k \in K, i \in I^k, \tag{2.4d}$$

$$\sum_{j \in I^k} s_j^k x_j^k \geq s_i^k \sum_{m \in M^k} v_i^m \quad \forall k \in K, i \in I^k, \tag{2.4e}$$

$$v_i^m, x_i^k \in \{0,1\} \quad \forall k \in K, i \in I^k, m \in M, \tag{2.4f}$$

where the objective function (2.4a) is the same as in model (BNLM^v) after replacing p_i by $\sum_{m \in M^k} b^m v_i^m$ (since for $v_i^m = 1$ with $m > \sigma(k)$, $x_i^k = 0$). Constraints (2.4b) are the upper level constraints (2.2a) that guarantee that products have at most one price. Constraints (2.4c) and (2.4d) are the lower level constraints (2.1d) and (2.2c), respectively. These constraints ensure that customers purchase at most one product which they can afford. Finally, constraints (2.4e) assure that, if customer k can afford product i, he purchases a product j he likes the same or better than i. Note that constraints (2.4e) affect $i \in I^k$ instead of $i \in I^{k*}$. If $i \in I^k \setminus I^{k*}$ then $\sum_{m \in M^k} v_i^m = 0$ and the constraint always holds. Otherwise, $\sum_{m \in M^k} v_i^m = 1$ and the constraint applies.

2.2 Single level formulation

In this section, we formulate the problem directly as a single level optimization problem. First of all, we introduce some definitions that will be used to represent the preferences of the customers in all the formulations from now on.

Let $k \in K$ be a customer and $i, j \in I^k$ two products. If customer k prefers product i over product j (i.e. $s_i^k > s_j^k$), it is denoted $i \prec_k j$ or $j \succ_k i$. Since preferences are strict, for any given products $i, j \in I^k$, it follows $i \prec_k j$ or $i \succ_k j$. It is also worth noticing that a customer k buys product i if and only if $i \in I^k$, its price is below the customer budget and all the products more preferred than i have a price higher than his budget. In terms of the binary variables x_i^k, v_i^m previously defined:

$$x_i^k = 1 \quad \Leftrightarrow \quad \sum_{m \in M^k} v_i^m = 1 \text{ and } \sum_{m \in M^k} v_j^m = 0 \ \forall j \in I^k : j \prec_k i.$$

Using this notation and decision variables x_i^k, v_i^m, a single level nonlinear formulation with two-index variables is

$$\text{(2INLM)} \qquad \max_{\mathbf{v},\mathbf{x}} \quad \sum_{k \in K} \sum_{i \in I^k} \left(\sum_{m \in M^k} b^m v_i^m \right) x_i^k \tag{2.5a}$$

$$\text{s.t.} \quad \sum_{i \in I^k} x_i^k \leq 1 \quad \forall k \in K, \tag{2.5b}$$

$$\sum_{m \in M} v_i^m \leq 1 \quad \forall i \in I, \tag{2.5c}$$

$$x_i^k + \sum_{m \in M^k} v_j^m \leq 1 \quad \forall k \in K, i \in I^k, j \in I^k : j \prec_k i, \tag{2.5d}$$

$$x_i^k + \sum_{\substack{m \in M: \\ m > \sigma(k)}} v_i^m \leq 1 \quad \forall k \in K, i \in I^k, \tag{2.5e}$$

$$v_i^m, x_i^k \in \{0,1\} \quad \forall k \in K, i \in I^k, m \in M, \tag{2.5f}$$

where constraints (2.4d) have been replaced by (2.5e) using constraints (2.5c). Constraints (2.5d), also called *preference constraints*, are given by the previous reasoning and can be strengthened by means of the following result:

Proposition 2.3. *The following constraints*

$$\sum_{\substack{j \in I^k: \\ j \succ_k i}} x_j^k + \sum_{m \in M^k} v_i^m \leq 1 \qquad \forall k \in K, i \in I^k, \tag{2.6}$$

are valid for (2INLM) and dominate constraints (2.5d).

Proof. First of all, we shall prove the validity of (2.6). We have $\sum_{j \in I^k : j \succ_k i} x_j^k \leq \sum_{j \in I^k} x_j^k \leq 1$ using (2.5b) and $\sum_{m \in M^k} v_i^m \leq \sum_{m \in M} v_i^m \leq 1$ because of (2.5c). Furthermore, provided that product i is within k's budget, i.e. if $\sum_{m \in M^k} v_i^m = 1$, then customer k will not buy any product he likes less than i, so $\sum_{j \in I^k : j \succ_k i} x_j^k = 0$, so (2.6) are valid.

If we change the notation of (2.6) and write $\sum_{i' \in I^k : i' \succ_k j} x_{i'}^k + \sum_{m \in M^k} v_j^m \leq 1$, $\forall k \in K$, $j \in I^k$, we obtain

$$x_i^k + \sum_{m \in M^k} v_j^m \leq \sum_{\substack{i' \in I^k: \\ i' \succ_k j}} x_{i'}^k + \sum_{m \in M^k} v_j^m \leq 1.$$

Therefore, we have proved that (2.6) are stronger than (2.5d). □

Interestingly, we can compare formulations (BNLM) and (2INLM) (with set (2.6) instead of (2.5d)) theoretically.

Set (2.6) is dominated by the following set:

$$\sum_{m \in M^k} v_i^m \leq x_i^k + \sum_{\substack{j \in I^k: \\ j \prec_k i}} x_j^k \qquad \forall k \in K, i \in I^k. \tag{2.7}$$

The justification is clear considering that the RHS of (2.7) is smaller than or equal to $1 - \sum_{\substack{j \in I^k: \\ j \succ_k i}} x_j^k$ due to constraints (2.5b). However, in practice the sets (2.6) and (2.7) perform nearly identically, so throughout the chapter we refer to model (2INLM) with (2.6) instead of (2.7). The reason is that in Section 2.4 we study the Set Packing problem related to a subset of the constraints of (2INLM), but (2.7) are not set packing constraints. In the following, we use (2.7) to prove that formulation (2INLM) is tighter than formulation (BNLM). The result is straightforward noting that the formulations are identical except for sets (2.6) and (2.4e), and considering the following proposition.

Proposition 2.4. *The family of constraints* (2.7) *from* (2INLM) *dominates family* (2.4e) *from* (BNLM).

Proof. Consider a fixed customer $k \in K$ and product $i \in I^k$, and assume that the corresponding inequality from set (2.7) holds. We are going to prove that, in this case, the corresponding inequality from set (2.4e) holds as well. Taking into account that for a given product $j \in I^k$ such that $j \prec_k i$ it holds $s_j^k > s_i^k > 0$ and hence $\frac{s_j^k}{s_i^k} > 1$, we can write:

$$\sum_{m \in M^k} v_i^m \leq x_i^k + \sum_{j \in I^k : j \prec_k i} x_j^k < \frac{s_i^k}{s_i^k} x_i^k + \sum_{j \prec_k i} \frac{s_j^k}{s_i^k} x_j^k + \sum_{j \succ_k i} \frac{s_j^k}{s_i^k} x_j^k = \sum_{j \in I^k} \frac{s_j^k}{s_i^k} x_j^k.$$

□

In Section 2.6, we see that the computational results confirm the theoretical comparison of the formulations obtained in this section.

2.3 Linearizing and strengthening formulations

Formulations (BNLM) and (2INLM) are nonlinear because of the objective functions (2.4a) and (2.5a). Since both objective functions are the same, from now on we refer to (2.5a). In order to linearize it, one approach consists in introducing variables z^k, $k \in K$, representing the profit obtained from customer k. Thus, the objective (2.5a) can be replaced by

$$\max_{v,x,z} \sum_{k \in K} z^k$$

and the following constraints need to be added to the formulation

$$z^k \leq \sum_{m \in M^k} b^m v_i^m + b^{\sigma(k)} \left(1 - x_i^k\right) \quad \forall k \in K, i \in I^k, \tag{2.8a}$$

$$z^k \leq b^{\sigma(k)} \sum_{i \in I^k} x_i^k \quad \forall k \in K, \tag{2.8b}$$

where constraints (2.8a) ensure that if customer k buys product i, $z^k = \sum_{m \in M^k} b^m v_i^m$ and (2.8b) guarantee $z^k = 0$ if customer k does not make any purchase. Constraints (2.8a) can be strengthened taking into account that customer k buys at most one item, obtaining

$$z^k \leq \sum_{m \in M^k} b^m v_i^m + b^{\sigma(k)} \sum_{j \in I^k : j \neq i} x_j^k \quad \forall k \in K, i \in I^k. \tag{2.9}$$

Therefore, we can reformulate problem (2INLM) obtaining a linear model as follows:

$$(2\text{IM}_1) \qquad \max_{\mathbf{v},\mathbf{x},\mathbf{z}} \quad \sum_{k \in K} z^k \tag{2.10a}$$

$$\text{s.t.} \quad \sum_{i \in I^k} x_i^k \leq 1 \quad \forall k \in K, \tag{2.10b}$$

$$\sum_{m \in M} v_i^m \leq 1 \quad \forall i \in I, \tag{2.10c}$$

$$\sum_{\substack{j\in I^k:\\ j\succ_k i}} x_j^k + \sum_{m\in M^k} v_i^m \leq 1 \quad \forall k\in K, i\in I^k, \tag{2.10d}$$

$$x_i^k + \sum_{\substack{m\in M:\\ m>\sigma(k)}} v_i^m \leq 1 \quad \forall k\in K, i\in I^k, \tag{2.10e}$$

$$z^k \leq \sum_{m\in M^k} b^m v_i^m + b^{\sigma(k)} \sum_{j\in I^k: j\neq i} x_j^k \quad \forall k\in K, i\in I^k, \tag{2.10f}$$

$$z^k \leq b^{\sigma(k)} \sum_{i\in I^k} x_i^k \quad \forall k\in K, \tag{2.10g}$$

$$v_i^m, x_i^k \in \{0,1\}, z^k \geq 0 \quad \forall k\in K, i\in I^k, m\in M. \tag{2.10h}$$

The nonlinearity of the objective function (2.5a) can alternatively be handled through the introduction of variables z_i^k, $k\in K$, $i\in I^k$, representing the profit obtained from customer k associated to product i. With these variables, the objective is

$$\max_{\mathbf{v},\mathbf{x},\mathbf{z}} \sum_{k\in K}\sum_{i\in I^k} z_i^k$$

and the following constraints ought to be added to the model:

$$z_i^k \leq \sum_{m\in M^k} b^m v_i^m \quad \forall k\in K, i\in I^k,$$

$$z_i^k \leq b^{\sigma(k)} x_i^k \quad \forall k\in K, i\in I^k.$$

Thus, the resulting model is

$$(2\mathrm{IM}_2) \qquad \max_{\mathbf{v},\mathbf{x},\mathbf{z}} \sum_{k\in K}\sum_{i\in I^k} z_i^k \tag{2.11a}$$

$$\text{s.t.} \quad \sum_{i\in I^k} x_i^k \leq 1 \quad \forall k\in K, \tag{2.11b}$$

$$\sum_{m\in M} v_i^m \leq 1 \quad \forall i\in I, \tag{2.11c}$$

$$\sum_{\substack{j\in I^k:\\ j\succ_k i}} x_j^k + \sum_{m\in M^k} v_i^m \leq 1 \quad \forall k\in K, i\in I^k, \tag{2.11d}$$

$$x_i^k + \sum_{\substack{m\in M:\\ m>\sigma(k)}} v_i^m \leq 1 \quad \forall k\in K, i\in I^k, \tag{2.11e}$$

$$z_i^k \leq \sum_{m\in M^k} b^m v_i^m \quad \forall k\in K, i\in I^k, \tag{2.11f}$$

$$z_i^k \leq b^{\sigma(k)} x_i^k \quad \forall k\in K, i\in I^k, \tag{2.11g}$$

$$v_i^m, x_i^k \in \{0,1\}, z_i^k \geq 0 \quad \forall k\in K, i\in I^k, m\in M. \tag{2.11h}$$

In formulations ($2\mathrm{IM}_1$) and ($2\mathrm{IM}_2$), the values of the z-variables associated to an assignment of prices to products (v-variables) and products to customers (x-variables) are obtained, respectively, by means of constraints (2.10f)-(2.10g) and (2.11f)-(2.11g). Although these constraints suffice to obtain the desired values of the z-variables, they lead to weak linear relaxations. Given the shape of the objective function, this weakness is directly transmitted to the upper bounds in the branch-and-bound method. Furthermore, in (2.10f) (resp. (2.11f)), a bound for z is obtained exclusively from the v-variables, and in (2.10g) (resp. (2.11g)), from the x-variables. These two issues invite to develop stronger constraints on the z-variables.

In what follows, two families of valid inequalities for (2IM$_1$) and (2IM$_2$) are presented. As will be shown in the computational study, they produce the desired improvement in the upper bounds given by the LP relaxation, and they have the particularity of relating the z-variables with both the x- and the v-variables at a time.

Proposition 2.5. *The following inequalities are valid for* (2IM$_1$)*:*

$$z^k \leq \sum_{i\in I^k}\left(b^{r_i^k}x_i^k + \sum_{m=r_i^k+1}^{\sigma(k)}\left(b^m - b^{r_i^k}\right)v_i^m + \sum_{m\in Q_i^k}\left(b^m - b^{r_i^k}\right)\left(x_i^k + v_i^m - 1\right)\right), \qquad (2.12)$$

$\forall k \in K$, *integers* $r_i^k \in \{0,\dots,\sigma(k)\}$ $\forall i \in I^k$ *and subsets* $Q_i^k \subseteq \{1,\dots,r_i^k - 1\}$ $\forall i \in I^k$.

Proof. Notice that in the case $r_i^k = 0$, set Q_i^k must be empty. We aim at proving that constraints (2.12) are valid for (2IM$_1$). Let us assume $x_{i_0}^k = 1$ for some $i_0 \in I^k$, and prove that the sum of the addends corresponding to product i_0 in the RHS of the constraint is greater than or equal to its price. Thus, such sum is

$$b^{r_{i_0}^k} + \sum_{m=r_{i_0}^k+1}^{\sigma(k)}\left(b^m - b^{r_{i_0}^k}\right)v_{i_0}^m + \sum_{m\in Q_{i_0}^k}\left(b^m - b^{r_{i_0}^k}\right)v_{i_0}^m, \qquad (2.13)$$

and we know that $v_{i_0}^{m_0} = 1$ for some $m_0 \leq \sigma(k)$. If $m_0 > r_{i_0}^k$, then $v_{i_0}^m = 0$ $\forall m \in Q_{i_0}^k$ and we get $b^{r_{i_0}^k} + (b^{m_0} - b^{r_{i_0}^k}) = b^{m_0}$, which is exactly the price of i_0. On the other hand, if $m_0 \leq r_{i_0}^k$ we have $v_{i_0}^m = 0$ $\forall m : r_{i_0}^k < m \leq \sigma(k)$, and therefore (2.13) becomes

$$b^{r_{i_0}^k} + \sum_{m\in Q_{i_0}^k}\left(b^m - b^{r_{i_0}^k}\right)v_{i_0}^m.$$

If $m_0 \notin Q_{i_0}^k$, we obtain $b^{r_{i_0}^k}$, which is greater than or equal to b^{m_0} because $r_{i_0}^k \geq m_0$; otherwise, if $m_0 \in Q_{i_0}^k$, then the term becomes $b^{r_{i_0}^k} + (b^{m_0} - b^{r_{i_0}^k}) = b^{m_0}$.

Now, let us suppose $x_{i_0}^k = 0$ for $i_0 \in I^k$. Then the addends corresponding to product i_0 become

$$\sum_{m=r_{i_0}^k+1}^{\sigma(k)}\left(b^m - b^{r_{i_0}^k}\right)v_{i_0}^m + \sum_{m\in Q_{i_0}^k}\left(b^m - b^{r_{i_0}^k}\right)\left(v_{i_0}^m - 1\right).$$

Since $(b^m - b^{r_{i_0}^k}) > 0$ for $m : r_{i_0}^k < m \leq \sigma(k)$ and $(b^m - b^{r_{i_0}^k}) < 0$ for $m \in Q_{i_0}^k$, then the sum is greater than or equal to zero.

Therefore, if $x_i^k = 0$ $\forall i \in I^k$, z^k is bounded from above by a sum of non-negative values. Otherwise, at any feasible solution at most one x-variable can take value 1 for a fixed customer k, say $x_{i_0}^k$. In this case, the upper bound is obtained as the sum of the term corresponding to product i_0 (which has been proved to be greater than or equal to the price assigned to i_0) plus some non-negative addends. $\square$

Remark 2.6. *The family of inequalities* (2.12) *contains all of the previous upper bound constraints on* z^k *of* (2IM$_1$)*. Constraints* (2.10f) *are obtained by, given a customer* $k \in K$ *and a product* $i \in I^k$*, setting* $r_i^k = 0$*,* $r_j^k = \sigma(k)\ \forall j \in I^k \setminus \{i\}$ *and* $Q_j^k = \emptyset\ \forall j \in I^k$ *in* (2.12)*; constraints* (2.10g)*, by, given a customer* $k \in K$*, setting* $r_i^k = \sigma(k)$ *and* $Q_i^k = \emptyset$ $\forall i \in I^k$*.*

Proposition 2.7. *The inequalities of the following family are valid for* (2IM$_2$)*:*

$$z_i^k \leq b^{r_i^k} x_i^k + \sum_{m=r_i^k+1}^{\sigma(k)} \left(b^m - b^{r_i^k}\right) v_i^m + \sum_{m \in Q_i^k} \left(b^m - b^{r_i^k}\right) \left(x_i^k + v_i^m - 1\right), \tag{2.14}$$

$\forall k \in K, i \in I^k$*, any integer* $r_i^k \in \{0, \ldots, \sigma(k)\}$ *and any subset* $Q_i^k \subseteq \{1, \ldots, r_i^k - 1\}$*.*

Proof. First assume that $x_i^k = 1$. This implies $v_i^{m_0} = 1$ for some $m_0 \leq \sigma(k)$. If $m_0 \leq r_i^k$, then $v_i^m = 0\ \forall m : r_i^k < m \leq \sigma(k)$ and (2.14) becomes $z_i^k \leq b^{r_i^k} + \sum_{m \in Q_i^k} (b^m - b^{r_i^k}) v_i^m$. If $m_0 \in Q_i^k$, then the RHS of the constraint is $b^{r_i^k} + (b^{m_0} - b^{r_i^k}) = b^{m_0}$, which is valid as it is the exact price of product i; otherwise, the RHS of the constraint is $b^{r_i^k}$, valid since $b^{r_i^k} \geq b^{m_0}$. If $m_0 > r_i^k$, then $v_i^m = 0\ \forall m \in Q_i^k$ and the inequality we obtain is $z_i^k \leq b^{r_i^k} + (b^{m_0} - b^{r_i^k})$, also valid.

On the other hand, if we assume $x_i^k = 0$, then the inequality holds trivially because its RHS is non negative and $z_i^k = 0$. □

Remark 2.8. *The family of inequalities* (2.14) *contains all of the previous upper bound constraints on* z_i^k *of* (2IM$_2$)*: constraints* (2.11f) *are obtained by setting* $r_i^k = 0$ *and* $Q_i^k = \emptyset$ $\forall k \in K$*,* $i \in I^k$*, whereas constraints* (2.11g) *appear as a result of setting* $r_i^k = \sigma(k)$*,* $Q_i^k = \emptyset$ $\forall k, i \in I^k$*.*

The family of inequalities from Proposition 2.7 (and consequently that of Proposition 2.5) is further improved in Chapter 4. However, we present these sets here because they were introduced in this way in the corresponding paper Calvete et al. (2019). Furthermore, the inequalities presented here are the ones used in the computational tests of Section 2.6.

The number of inequalities of Propositions 2.5 and 2.7 increases exponentially as the number of customers and products grows. However, these inequalities can be efficiently separated and added dynamically to formulations (2IM$_1$) and (2IM$_2$), respectively, in a branch-and-cut mode. Thus, regarding the family of valid inequalities (2.12), and given a fractional optimal solution of the linear relaxation of (2IM$_1$), $(\overline{v}_i^m, \overline{x}_i^k, \overline{z}^k)$, our aim is to find, for each $k \in K$, integers r_i^k and subsets Q_i^k $\forall i \in I^k$ such that the upper bound given by the RHS of the resultant constraint of the family is as tight as possible. As the sum given by the RHS of (2.12) can be decomposed by products and given that $\overline{z}$ is fixed, our problem reduces to

$$\min_{\substack{r \in \{0,\ldots,\sigma(k)\},\\ Q \subseteq \{1,\ldots,r-1\}}} b^r \overline{x}_i^k + \sum_{m=r+1}^{\sigma(k)} (b^m - b^r)\, \overline{v}_i^m + \sum_{m \in Q} (b^m - b^r) \left(\overline{x}_i^k + \overline{v}_i^m - 1\right), \tag{2.15}$$

where $(k, i) \in K \times I^k$ is fixed, and we have denoted r_i^k as r and Q_i^k as Q so as to simplify notation. It is worth noticing that this pair (r, Q) also minimizes the RHS of the corresponding constraint of family (2.14) when given an optimal fractional solution of the linear relaxation of (2IM$_2$), $(\overline{v}_i^m, \overline{x}_i^k, \overline{z}_i^k)$, and fixed $(k,i) \in K \times I^k$. Thus, finding a pair (r,Q) that minimizes (2.15) for a given customer k and product i not only leads to the development of an efficient separation algorithm for the set of valid inequalities (2.12), but also for the set (2.14).

The fact that $(b^m - b^r) \leq 0 \; \forall m \leq r$ implies that, for a given r, $Q^r := \{m \in \{1,\dots,r-1\} : \overline{x}_i^k + \overline{v}_i^m > 1\}$ minimizes (2.15). Therefore, if $W(r)$ is the value of the sum (2.15) when $Q = Q^r$, our problem consists in minimizing $W(r)$ for $r \in \{0,\dots,\sigma(k)\}$.

To do so, we shall study the variation of $W(r)$ as r increases. Given that $Q^{r+1} = Q^r \cup \{r\}$ if $\overline{x}_i^k + \overline{v}_i^r > 1$, $Q^{r+1} = Q^r$ otherwise, for $r < \sigma(k)$ we get

$$
\begin{aligned}
W(r+1) - W(r) = &\left(b^{r+1}\overline{x}_i^k + \sum_{m=r+2}^{\sigma(k)} \left(b^m - b^{r+1}\right)\overline{v}_i^m + \sum_{m\in Q^{r+1}} \left(b^m - b^{r+1}\right)\left(\overline{x}_i^k + \overline{v}_i^m - 1\right)\right) \\
&- \left(b^{r}\overline{x}_i^k + \sum_{m=r+1}^{\sigma(k)} \left(b^m - b^{r}\right)\overline{v}_i^m + \sum_{m\in Q^{r}} \left(b^m - b^{r}\right)\left(\overline{x}_i^k + \overline{v}_i^m - 1\right)\right) \\
= &(b^{r+1} - b^r)\overline{x}_i^k + \sum_{m=r+2}^{\sigma(k)} \left(b^r - b^{r+1}\right)\overline{v}_i^m - \left(b^{r+1} - b^r\right)\overline{v}_i^{r+1} \\
&+ \sum_{m\in Q^{r+1}} \left(b^r - b^{r+1}\right)\left(\overline{x}_i^k + \overline{v}_i^m - 1\right) \\
= &\left(b^{r+1} - b^r\right)\left(\overline{x}_i^k - \sum_{m=r+1}^{\sigma(k)} \overline{v}_i^m + \sum_{m\in Q^{r+1}} \left(1 - \overline{x}_i^k - \overline{v}_i^m\right)\right).
\end{aligned}
\tag{2.16}
$$

First of all, we are going to prove that, when r increases from 0 to $\sigma(k)$, $W(r)$ first decreases and then increases. We can achieve that by proving that $W(r) - W(r-1) \geq 0 \Rightarrow W(r+1) - W(r) \geq 0$. Since $b^{r+1} - b^r > 0 \; \forall r$, it follows from (2.16) that $W(r+1) - W(r) \geq 0 \Leftrightarrow \overline{x}_i^k - \sum_{m=r+1}^{\sigma(k)} \overline{v}_i^m + \sum_{m\in Q^{r+1}} \left(1 - \overline{x}_i^k - \overline{v}_i^m\right) \geq 0 \; \forall r < \sigma(k)$, and therefore demonstrating the above is equivalent to proving $\overline{x}_i^k - \sum_{m=r+1}^{\sigma(k)} \overline{v}_i^m + \sum_{m\in Q^{r+1}} \left(1 - \overline{x}_i^k - \overline{v}_i^m\right) - \left(\overline{x}_i^k - \sum_{m=r}^{\sigma(k)} \overline{v}_i^m + \sum_{m\in Q^{r}} \left(1 - \overline{x}_i^k - \overline{v}_i^m\right)\right) \geq 0$. But we have

$$
\begin{aligned}
&\overline{x}_i^k - \sum_{m=r+1}^{\sigma(k)} \overline{v}_i^m + \sum_{m\in Q^{r+1}} \left(1 - \overline{x}_i^k - \overline{v}_i^m\right) - \left(\overline{x}_i^k - \sum_{m=r}^{\sigma(k)} \overline{v}_i^m + \sum_{m\in Q^{r}} \left(1 - \overline{x}_i^k - \overline{v}_i^m\right)\right) \\
&\qquad = \overline{v}_i^r + \min\left\{0, 1 - \overline{x}_i^k - \overline{v}_i^r\right\} = \min\left\{\overline{v}_i^r, 1 - \overline{x}_i^k\right\} \geq 0.
\end{aligned}
$$

Hence, $W(r)$ reaches its minimum value for the smallest r such that $W(r) - W(r-1) \leq 0$ and $W(r+1) - W(r) > 0$.

Furthermore, noticing in (2.16) that $\sum_{m\in Q^{r+1}} \left(1 - \overline{x}_i^k - \overline{v}_i^m\right) \leq 0 \; \forall r$ allows us to deduce that $W(r) - W(r-1) \leq 0$ provided that $\overline{x}_i^k - \sum_{m=r}^{\sigma(k)} \overline{v}_i^m \leq 0$, i.e., if r is such that

$\overline{x}_i^k \leq \sum_{m=r}^{\sigma(k)} \overline{v}_i^m$. This fact saves us having to compute the whole sum (2.16) in order to know if $W(r) - W(r-1) \leq 0$ whenever $\overline{x}_i^k \leq \sum_{m=r}^{\sigma(k)} \overline{v}_i^m$.

After finding a separation for valid inequalities (2.12), the next step consists in defining a procedure to incorporate these inequalities into formulation (2IM_1) dynamically in a branch-and-cut framework where the starting subproblem of every child node is the final formulation of the parent node with the corresponding branching x- or v-variable fixed to either zero or one. A scheme of this procedure is depicted in Algorithm 1. Preliminary testing shows that the best strategy amounts to adding these inequalities to the formulation provided that the node depth in the branching tree is less than or equal to 4. The termination criterion is that the optimal value of the linear relaxation of that node does not improve in the last iteration. Both the algorithm and the branch-and-cut procedure used to include dynamically inequalities (2.14) into model (2IM_2) are analogous to these ones.

Algorithm 1 Separation of inequalities (2.12)

Let $(\overline{x}_i^k, \overline{v}_i^m, \overline{z}^k)$ be an optimal fractional solution of the linear relaxation of (2IM_1).
For every customer $k \in K$ do

Step 1. For every product $i \in I^k$ do

Step 1.1. Set $r_i^k = 0$.
Step 1.2. If $r_i^k < \sigma(k)$ and $\sum_{m=r}^{\sigma(k)} \overline{v}_i^m \leq \overline{x}_i^k$, update $r_i^k := r_i^k + 1$ and repeat Step 1.2.

Otherwise, go to Step 1.3.
Step 1.3. If $r_i^k < \sigma(k)$ and $W(r_i^k + 1) - W(r_i^k) \leq 0$, update $r_i^k := r_i^k + 1$ and repeat Step 1.3.

Otherwise, go to Step 2.

Step 2. Set $Q_i^k := \{m \in \{1, \ldots, r_i^k - 1\} : \overline{x}_i^k + \overline{v}_i^m > 1\}$ $\forall i \in I^k$.
Step 3. Incorporate constraint

$$z^k \leq \sum_{i \in I^k} \left(b^{r_i^k} x_i^k + \sum_{m=r_i^k+1}^{\sigma(k)} \left(b^m - b^{r_i^k} \right) v_i^m + \sum_{m \in Q_i^k} \left(b^m - b^{r_i^k} \right) \left(x_i^k + v_i^m - 1 \right) \right)$$

to the formulation if and only if it is violated.

2.4 Polyhedral analysis of the set packing subproblem

In this section, we analyze the subproblem of model (2IM_1) (resp. model (2IM_2)) associated to x- and v-variables and constraints (2.10b)-(2.10e) (resp. constraints (2.11b)-(2.11e)), given that it constitutes a special case of an SP problem. Since this subproblem is the same for both models (2IM_1) and (2IM_2), in the rest of the section we shall refer to the subproblem of model (2IM_1). First, we give an introduction on Set Packing problems and how their properties can be exploited to derive strong valid inequalities, and in particular facets of the polytope given by the convex hull of the integer feasible solutions (see subsection 1.1.3 of Chapter 1).

2.4.1 Set Packing problems

The Set Packing problem is a classical problem in pure Integer Programming and has been extensively studied (see Balas and Padberg (1976)). A *Set Packing Problem* (SP) can be formulated as an integer linear program of the form

$$\max_x \quad cx \tag{2.17a}$$
$$\text{s.t.} \quad Ax \leq e, \tag{2.17b}$$
$$x \in \{0,1\}^n, \tag{2.17c}$$

where $c \in \mathbb{R}^n$, $A \in \{0,1\}^{m\times n}$ and $e^t = (1,\dots,1)$ is a m-vector of ones. The interpretation can be stated as follows. Let $M = \{1,\dots,m\}$ be a set of elements associated with the rows of A, and several subsets $M_j \subset M$ of weight c_j, $j \in \{1,\dots,n\}$ such that M_j contains an element $i \in M$ iff $a_{ij} = 1$. Then find a maximum-weight family of subsets M_j such that each element $i \in M$ belongs to at most one subset M_j.

The SP problem is closely related (in fact, both problems are equivalent) to the *Set Partitioning Problem* (SPP), where the set of constraints (2.17b) has to be satisfied in equality. The name comes from the interpretation, since in this case we seek for subsets M_j that constitute a partition of M. It also has a more distant relative called the *Set Covering Problem* (SC), where (2.17b) is replaced with $Ax \geq e$. The SP, SC and SPP problems find applications in a large variety of fields, like crew scheduling of airlines or railroads, facility location, stock cutting or truck delivery (see Vemuganti (1998)). They can also appear as subproblems associated to a subset of binary variables, like in our case. The three problems are NP-hard (see Garey and Johnson (1979)). Here we focus on the polyhedral structure of the SP problem. We introduce the connection between the SP problem and the Node Packing Problem, and explain how the study of the intersection graph of the latter is useful in the generation of valid inequalities for the former. A more in-depth introduction to these topics can be found in Padberg (1973); Balas and Padberg (1976); Atamtürk et al. (2000).

Let $G = (V,E)$ be a loopless undirected graph with $|V| = n$, $|E| = m$. A *node packing* (vertex packing, stable set, independent set) in G is a non empty subset $V' \subset V$ of pairwise non adjacent nodes. The *Node Packing Problem* (NP) is then the problem of obtaining a node packing of maximal cardinality on G, and can be formulated as

$$\max_x \quad e^t x$$
$$\text{s.t.} \quad A_G x \leq e,$$
$$x \in \{0,1\}^n,$$

where A_G is the (node-edge) incidence matrix of G, i.e. $A_G = (a_{ij})$ is a $n \times m$ (0,1)-matrix such that $a_{ij} = 1$ iff node j is one of the ends of edge i. Clearly, the NP problem is a special case of the SP problem. Furthermore, we can reinterpret the SP as a node packing by noticing that every SP instance can be univocally identified with a graph $G_A = (V_A, E_A)$ called the *intersection graph* (conflict graph). G_A consists of a node v_j representing each binary variable and x_j and $(v_i, v_j) \in E_A$ if and only if $a_{ki}a_{kj} = 1$ in some row k of A, i.e. columns i and j of A have at least one positive entry in common. The *incidence vector* of a subset $V' \subset V$ is a binary vector $(x_1, \ldots, x_{|V|})$ where $x_j = 1$ if and only if $v_j \in V'$, for $j \in \{1, \ldots, |V|\}$. The *packing polytope* $\mathcal{P}_G$ is the convex hull of the incidence vectors of all the packings of the intersection graph G. And since the feasible solutions of the SP problem can be identified with packings in its conflict graph, the polytope $\mathcal{P}_G$ is the same as the polytope given by the convex hull of feasible solutions to the original SP program. This result is useful to identify valid inequalities of an SP instance by means of special structures (subgraphs) of its intersection graph. $G[V'] = (V', E_{V'})$ denotes the *subgraph* of graph $G = (V, E)$ induced by $V' \subseteq V$ when $E_{V'} = \{(v_i, v_j) \in E : v_i, v_j \in S\}$.

One of the first structures of the intersection graph derived in the literature that originates facets of $\mathcal{P}_G$ is called clique.

Definition 2.9 (Clique)**.** A *complete* graph is that in which all the nodes are pairwise adjacent. Given a graph $G = (V, E)$ and a subset of nodes $V' \subseteq V$, $G[V']$ is a *clique* in G if it is a maximal complete subgraph of G, i.e. if $G[V']$ is complete and $G[V' \cup \{v\}]$ is not complete for any node $v \in V \setminus V'$.

Fulkerson (1971) and Padberg (1973) proved that the valid inequalities associated to cliques are facet-defining:

Theorem 2.10. *An inequality $\sum_{v \in V'} x_v \leq 1$, where $V' \subseteq V$, is a facet for $\mathcal{P}_G$ if and only if $G[V']$ is a clique in G.*

Padberg (1973) also proved that some facets can be derived from odd holes (chordless cycles of odd length) in G. Other configurations that help to derive facets of the packing polytope include odd anti-holes (Nemhauser and Trotter (1974)), some types of *webs* and *anti-webs* (Trotter (1975)), subdivided *wheels* (Cheng and Cunningham (1997)), *grilles* (Cánovas et al. (2000)), and many more. Clique facets (valid inequalities induced by a clique) are particularly interesting because they are easier to identify in the graph than other structures, and their addition to a problem generally provides better results (when trying to solve it) than the addition of other types of facets which are more complex. Another reason is that, unlike cliques, other subgraphs $G[V']$ of G (for instance holes or webs) do not produce facets of $\mathcal{P}_G$, but of $\mathcal{P}_{G[V']}$. They are nonetheless useful to derive facet-defining inequalities of the original graph when combined with *lifting* procedures (see Nemhauser and Trotter (1974); Padberg (1975); Landete (2001), among others).

2.4.2 Set packing subproblem of (2IM_1)

In order to apply the SP properties to our problem, we begin by identifying the intersection graph G2IM associated to the previously defined subproblem of formulation (2IM_1). G2IM has a set of nodes associated to variables x_i^k and another set associated to variables v_i^m. As for the edges, the large amount of them makes drawing it impractical, so we will follow a different approach in order to describe the intersection graph based on the following proposition. Before proving the main results of this section, we introduce some useful notation:

Definition 2.11. Let k be a customer, $i \in I^k$ a product and $P \subseteq I^k$ a subset of products in which k is interested. Then we define $B(k, i) = \{j \in I^k : j \prec_k i\}$ as the set of products

k prefers over i, and $\overline{B(k,i)} = \{j \in I^k : j \succ_k i\}$ as the set of products k likes less than i. Similarly, we define $B(k,P) = \{i \in I^k :\ i \prec_k j\ \forall j \in P\}$ as the set of products that are preferred by k to all the products in P. Finally, $\overline{B(k,P)} := \{i \in I^k :\ i \succ_k j\ \forall j \in P\}$. In the special case when $P = \emptyset$ we define $B(k,\emptyset) := I$ and $\overline{B(k,\emptyset)} := I$.

Proposition 2.12. *Given the intersection graph* G2IM *associated to the subgraph of* (2IM$_1$)*:*

(1) *Two nodes x_i^k, x_j^k, $i \neq j$, are adjacent $\forall i, j \in I^k$.*

(2) *Two nodes x_i^k, $x_i^{k'}$, $k \neq k'$, are never adjacent.*

(3) *Two nodes x_i^k, $x_j^{k'}$, $k \neq k'$, $i \neq j$, are adjacent if and only if $\sigma(k) \geq \sigma(k')$ and $j \in B(k,i)$ (or, equivalently, $i \in \overline{B(k,j)}$).*

(4) *Two nodes x_i^k, v_i^m, are adjacent if and only if $m > \sigma(k)$.*

(5) *Two nodes x_i^k, v_j^m, $i \neq j$ are adjacent if and only if $m \leq \sigma(k)$ and $j \in B(k,i)$.*

(6) *Two nodes v_i^m, $v_i^{m'}$, $m \neq m'$, are adjacent $\forall m, m'$.*

(7) *Two nodes v_i^m, $v_j^{m'}$, $i \neq j$, are never adjacent.*

Proof.

(1) A customer k purchases at most one product.

(2) The fact that a customer k purchases a product i does not imply that another customer cannot afford it (that depends on i's price), and therefore does not allow us to determine whether another customer is going to buy it or not.

(3) Let us suppose $x_i^k = 1$, i.e., customer k purchases product i. That implies k is not able to afford any product $j \in B(k,i)$, and therefore no customer k' with $\sigma(k') \leq \sigma(k)$ is able to afford it either, hence $x_j^{k'} = 0$. However, the fact that k purchases product i does not allow us to infer which products will not be purchased by other customers k' richer than k or which customers will not purchase a product $j \in \overline{B(k,i)} \cup \{i\}$.

(4) If $x_i^k = 1$, k can afford product i, so there must exist $m_0 \leq \sigma(k)$ such that $v_i^{m_0} = 1$. Since product i can have one price at most, it follows $v_i^m = 0\ \forall m > \sigma(k)$.

(5) Let us suppose $x_i^k = 1$, i.e. customer k purchases product i. That implies k is not able to afford any product j such that $j \prec_k i$, that is, $v_j^m = 0\ \forall j \in B(k,i)$, $\forall m \leq \sigma(k)$. However, it does not provide any insight into the prices of products $j \in \overline{B(k,i)}$.

(6) A product i can have at most one price.

(7) Knowing the price of a product does not provide any insight into the price of the rest.

□

In what follows, and for the sake of clarity, we show an example of the neighborhoods corresponding with given nodes x_i^k and v_i^m in the intersection graph we have just introduced.

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	Budget
$k=1$	4	2	-	1	3	20
$k=2$	2	5	4	1	3	14
$k=3$	-	$\boxed{4}$	$\boxed{1}$	$\boxed{2}$	3	14
$k=4$	-	-	1	3	2	8
$k=5$	1	2	-	3	-	5

Figure 2.1: Preference matrix of an instance of (2INLM) and nodes x_i^k adjacent to x_5^3 (x_5^3 in grey)

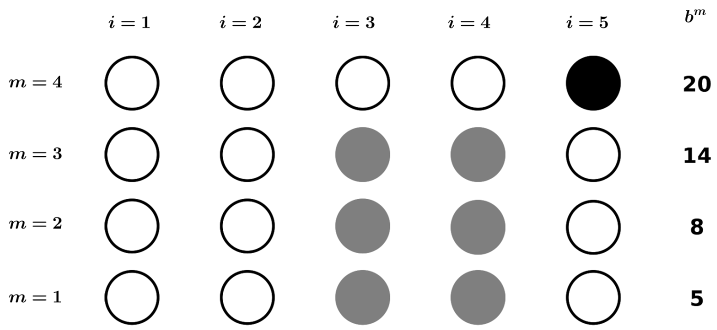

Figure 2.2: Matrix (v_i^m) of an instance of (2INLM) and nodes v_i^m adjacent to x_5^3

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	Budget
$k=1$	4	2	-	1	3	20
$k=2$	2	5	4	1	3	14
$k=3$	-	4	1	2	3	14
$k=4$	-	-	1	$\boxed{3}$	2	8
$k=5$	1	2	-	$\boxed{3}$	-	5

Figure 2.3: Preference matrix of an instance of (2INLM) and nodes x_i^k adjacent to v_4^3

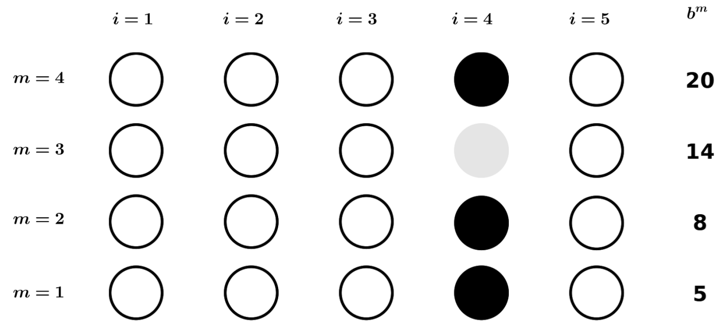

Figure 2.4: Matrix (v_i^m) of an instance of (2INLM) and nodes v_i^m adjacent to v_4^3 (v_4^3 in grey)

Example 2.13. *Figure 2.1 shows a small instance of the RPP that we will use throughout the section combined with Figures 2.2-2.4 to illustrate our results. Thus, the matrix in Figures 2.1 and 2.3 (it is the same matrix) is the preference matrix, and entry (k, i) shows the ranking t_i^k of customer k over product i. If i is the favorite product for customer k, then $t_i^k = 1$; if j is the second most preferred product for k, $t_j^k = 2$,* et cetera. *Similarly, making use of the previously described notation, we have $|M| = 4$, and $b^1 = 5$, $b^2 = 8$, $b^3 = 14$ and $b^4 = 20$.*

Furthermore, we make use of the preference matrix to depict the nodes of G2IM associated to x-variables, so that the node associated to x_i^k is represented in the entry (k, i) of the matrices in Figures 2.1 and 2.3. The nodes associated to v-variables are represented through a different matrix in Figures 2.2 and 2.4, where again entry (m, i) of the matrix corresponds to node (and variable) v_i^m. We use Figure 2.1 to show which nodes related to x-variables are adjacent to a specific x-variable, x_5^3 in our example. Figure 2.2 depicts the set of v-nodes adjacent to this same node. As for Figures 2.3 and 2.4, they show the set of x-nodes and v-nodes (resp.) adjacent to a given v-node, which is v_4^3 in this example. The nodes adjacent to a given one due to the same item of Proposition 2.12 are highlighted using a particular symbol (such as a triangle or a square) or color in each of the figures.

*Let us suppose $x_5^3 = 1$, i.e. customer 3 purchases product 5 ($t_5^3 = 3$ and it is grey in Figure 2.1). Our aim is to determine all nodes of G2IM adjacent to x_5^3 by making use of Proposition 2.12. Due to **(1)**, we know x_5^3 is adjacent to x_2^3, x_3^3 and x_4^3 (marked with a square in Figure 2.1). By **(3)** applied to $k = 1$, $k' = 3$, $j = 5$, we obtain that there exists a link between nodes x_5^3 and x_i^1 $\forall i \in \overline{B(1,5)} = \{1\}$. **(3)** applied to $k = 2$, $k' = 3$ and $j = 5$ indicates x_5^3 is adjacent to x_2^2 and x_3^2 (all of these nodes are marked in Figure 2.1 through a triangle). Since $\sigma(2) = \sigma(3)$, we can also apply **(3)** to $k = 3$, $i = 5$, $k' = 2$, obtaining edges between nodes x_5^3 and x_3^2, x_4^2. Similarly, applying **(3)** to $k = 3$, $i = 5$ and $k' = 4, 5$ leads to edges between x_5^3 and the nodes x_3^4, x_4^4 and x_4^5 (all of which are circled in Figure 2.1).*

*Next, for the same instance of the problem we are going to determine which nodes v_i^m are linked by edges with the same node x_5^3, by making use of Figure 2.2. By Proposition 2.12**(4)** we obtain that x_5^3 and v_5^m are adjacent for $m > \sigma(3) = 3$, i.e. for $m = 4$ (in black in Figure 2.2). Applying **(5)**, there exist edges between x_5^3 and v_j^m for $j \in B(3,5) = \{3, 4\}$ and $m \leq 3$ (colored in dark grey in Figure 2.2).*

*Now let us suppose $v_4^3 = 1$ (i.e. product 4 has a price of $b^3 = 14$), so as to determine which nodes x_i^k are adjacent to it. By Proposition 2.12**(4)**, v_4^3 is adjacent to x_4^k for $k \in \{4,5\}$ (marked with a square in Figure 2.3). Since $\sigma(1) = 4 \geq 3$, for **(5)** v_4^3 and x_i^1 are adjacent for $i \in \overline{B(1,4)} = \{1,2,5\}$. Similarly, we can obtain edges between v_4^3 and x_i^2 for $i \in \overline{B(2,4)} = \{1,2,3,5\}$ and x_i^3 for $i \in \overline{B(3,4)} = \{2,5\}$ (all of them circled in Figure 2.3).*

*Finally, nodes v_i^m adjacent to v_4^3 (in grey) can be seen in Figure 2.4. Thus, by **(6)** v_4^3 is adjacent to v_4^m for $m \neq 3$ (nodes in black), and **(7)** establishes that v_4^3 is not adjacent to any other node.*

Having identified the intersection graph G2IM, the next subsection focuses on characterizing all its cliques.

2.4.3 Characterization of all the cliques in the intersection graph

We first include a lemma that will be useful when characterizing all the cliques.

Lemma 2.14. *Any clique in* G2IM *containing nodes $v_i^{m_1}$, $v_i^{m_2}$ with $m_1 < m_2$, includes v_i^m $\forall m$ such that $m_1 < m < m_2$.*

Proof. Let (V', E') be a clique in G2IM and suppose $v_i^{m_1}, v_i^{m_2} \in V'$, for $m_1 < m_2$.

Let us suppose that there exists $k \in K$ with $x_i^k \in V'$. Then, x_i^k is adjacent to $v_i^{m_1}$, and thus for Prop. 2.12**(4)** it follows $\sigma(k) < m_1$. Therefore, for every $m > m_1 > \sigma(k)$, the same result implies x_i^k is adjacent to v_i^m.

Now let us suppose that $x_j^k \in V'$ for some $k \in K$ and $j \in I^k$, $j \neq i$. By hypothesis we have x_j^k adjacent to $v_i^{m_2}$, which for Proposition 2.12**(5)** implies $i \in B(k,j)$ and $\sigma(k) \geq m_2$. Thus, for every $m < m_2 \leq \sigma(k)$, it follows from the same result that x_j^k is adjacent to v_i^m.

Finally, we know from Proposition 2.12**(6)** and **(7)** that v_j^m adjacent to $v_i^{m_1} \Leftrightarrow j = i$, hence v_i^m is adjacent to $v_i^{m'}$ $\forall m \neq m'$ and $v_j^m \notin V'$ for $j \neq i$.

All in all, we have proven that for m such that $m_1 < m < m_2$, any variable x_j^k or $v_i^{m'} \in V'$ is adjacent to v_i^m. Thus, the statement follows. □

Now we can state the two main results in this section. Note that, in order to keep a consistent notation, a set $\{k_2, \ldots, k_n\}$ is defined in Theorem 2.15 that will be extended to $\{k_1, \ldots, k_n\}$ in Theorem 2.17.

Theorem 2.15. *Given a set of customers $\{k_2, \ldots, k_n\}$, $n \geq 2$, with $\sigma(k_2) \leq \cdots \leq \sigma(k_n)$, and non empty pairwise disjoint sets of products $P^{k_q} \subseteq I^{k_q}$, $q = 2, \ldots, n$, such that*

$$P^{k_q} \subseteq \left(\bigcap_{\substack{r=2:\\ \sigma(k_r)<\sigma(k_q)}}^{q-1} \overline{B(k_q, P^{k_r})} \right) \bigcap \left(\bigcap_{\substack{r=2:\\ \sigma(k_r)=\sigma(k_q)}}^{q-1} \left(\overline{B(k_q, P^{k_r})} \cup B(k_r, P^{k_r}) \right) \right) \quad \forall q \in \{3, \ldots, n\},$$

the following inequalities are valid for $(2\mathrm{IM}_1)$*:*

$$\sum_{q=2}^{n} \sum_{j \in P^{k_q}} x_j^{k_q} \leq 1. \tag{2.19}$$

Valid inequalities (2.19) *are facets for the subproblem of* (2IM$_1$) *if and only if* $\nexists (k_0, i_0) \in K \times I^{k_0}$ *satisfying*

1. $i_0 \in B(k_q, P^{k_q})\ \forall q \in \{2, \dots, n\} : \ \sigma(k_q) \geq \sigma(k_0)$,
2. $i_0 \in \overline{B(k_0, P^{k_q})}\ \forall q \in \{2, \dots, n\} : \ \sigma(k_q) \leq \sigma(k_0)$,

and $|\bigcup_{\substack{q=2: \\ \sigma(k_q)=\sigma(k_2)}}^{n} P^{k_q}| \geq 2$. *Furthermore, all the clique facets for the subproblem of* (2IM$_1$) *containing only* x*-variables are in family* (2.19).

Proof. Let G2IM $= (V_G, E_G)$ be the intersection graph of the subproblem of (2IM$_1$) associated to x- and v-variables and constraints (2.10b)-(2.10e), and let $Q = (V', E')$ be a clique of G2IM containing only x-variables.

Let k_2 be a customer with minimum budget in the clique and a subset of products $P^{k_2} \subseteq I^{k_2}$ such that $x_j^{k_2} \in V'\ \forall j \in P^{k_2}$ (taking into account that, by Proposition 2.12**(1)**, $x_i^{k_2}$ is adjacent to $x_j^{k_2}\ \forall i \neq j$).

Provided that there exist customers k_q, $\forall q \in \{3, \dots, n\}$ such that $\sigma(k_2) \leq \sigma(k_3) \leq \cdots \leq \sigma(k_n)$ and sets of products $P^{k_q} \subseteq I^{k_q}$, $P^{k_q} \neq \emptyset\ \forall q \in \{3, \dots, n\}$, such that $x_j^{k_q} \in V'$ $\forall j \in P^{k_q}$, then by Proposition 2.12**(2)** $P^{k_2}, \dots, P^{k_n}$ are pairwise disjoint, and verify the following conditions:

- $$P^{k_q} \subseteq \bigcap_{\substack{r=2 \\ \sigma(k_r)<\sigma(k_q)}}^{q-1} \overline{B(k_q, P^{k_r})} \quad \forall q \in \{3, \dots, n\}.$$

  Otherwise, there exist k_r with $\sigma(k_r) < \sigma(k_q)$ and products $i \in P^{k_r}$, $j \in P^{k_q}$ such that $x_i^{k_r}$, $x_j^{k_q} \in V'$ but $j \notin \overline{B(k_q, i)}$, and by Proposition 2.12**(3)** this implies $x_i^{k_r}$, $x_j^{k_q}$ are not neighbors in the intersection graph. Therefore, V' does not induce a complete graph.

- $$P^{k_q} \subseteq \bigcap_{\substack{r=2 \\ \sigma(k_r)=\sigma(k_q)}}^{q-1} \left(\overline{B(k_q, P^{k_r})} \cup B(k_r, P^{k_r})\right) \quad \forall q \in \{3, \dots, n\}.$$

  Otherwise, there exist k_r with $\sigma(k_r) = \sigma(k_q)$ and products $i \in P^{k_r}$, $j \in P^{k_q}$ such that $x_i^{k_r}$, $x_j^{k_q} \in V'$ but Proposition 2.12**(3)** does not hold for $k = k_r$, $k' = k_q$ or for $k = k_q$, $k' = k_r$, and hence V' does not induce a complete graph.

Therefore, the above conditions guarantee that the nodes corresponding with the x-variables in an inequality in the form of (2.19) induce a complete graph, so the family of inequalities (2.19) is valid.

In addition, if there exist $(k_0, i_0) \in K \times I^{k_0}$ meeting the conditions of the statement, then $x_{i_0}^{k_0}$ is adjacent in the intersection graph to every other node in V' by Proposition 2.12**(3)** and conditions 1 and/or 2, and therefore the complete subgraph is not maximal.

	x			
	$k=2$	$k=3$	$k=4$	$k=5$
i	12345			
	1235	4		
	234	5		
	235			14
	23	45		
	23		45	
	2	345		
	2	5	34	
	2		345	
	345	2		
	34	25		
	35	24		
	3	245		
	3	2	45	
		2345		
		25	34	
		2	345	
				124

Table 2.2: Facets obtained applying Theorem 2.15 to the instance of the RPP given in Figure 2.1

On the other hand, if $|\bigcup_{\substack{q=2:\\ \sigma(k_q)=\sigma(k_2)}}^{n} P^{k_q}| \geq 2$ holds, no v-variable can be adjacent in the intersection graph to all nodes in V'. Otherwise, $P^{k_2} = \{i\}$ and either $n = 2$ or $\sigma(k_2) < \sigma(k_3)$, and hence variable $v_i^{\sigma(k_2)+1}$ would be adjacent to every node in V' and the complete subgraph would not be maximal. □

Example 2.16. *By making use of the instance shown in Figure 2.1, we are going to give examples of clique facets containing only x-variables and valid inequalities that are not cliques. Consider first the set of x-variables $\{x_1^2, x_2^2, x_3^2, x_5^2\}$. Given that all these variables share the superindex, their corresponding nodes are neighbors in the intersection graph and then $x_1^2 + x_2^2 + x_3^2 + x_5^2 \leq 1$ is a valid inequality for (2INLM). This inequality corresponds with $n = 2$, $k_2 = 2$ ($\sigma(k_2) = 3$) and $P^2 = \{1, 2, 3, 5\}$ in Theorem 2.15. Nevertheless, it does not define a facet for (2INLM) since a pair $(k_0, i_0) = (4, 4)$ exists (with $\sigma(4) = 2 \leq 3$) in the case 1 of the theorem, with $4 \in B(2, \{1, 2, 3, 5\}) = \{4\}$). Then, variable x_4^4 could be added to the LHS to produce the tighter inequality $x_1^2 + x_2^2 + x_3^2 + x_5^2 + x_4^4 \leq 1$. This inequality corresponds now to $n = 3$, $k_2 = 4$, $k_3 = 2$, $P^4 = \{4\}$, $P^2 = \{1, 2, 3, 5\}$ and it is* *not a facet for (2INLM) either, since* $|\bigcup_{\substack{q=2:\\ \sigma(k_q)=\sigma(k_2)}}^{n} P^{k_q}| = |\bigcup_{\substack{q=2:\\ \sigma(k_q)=2}}^{3} P^{k_q}| = |P^{k_2}| = 1$ *and, in this case, variable* v_4^3 *could be added to the inequality to produce the tighter one given by* $x_1^2 + x_2^2 + x_3^2 + x_5^2 + x_4^4 + v_4^3 \leq 1$.

Consider, instead, $(k_0, i_0) = (3, 4)$, *i.e.,* $n = 3$, $k_2 = 3$, $k_3 = 2$, $P^3 = \{4\}$, $P^2 = \{1, 2, 3, 5\}$ *where* $|\bigcup_{\substack{q=2:\\ \sigma(k_q)=\sigma(k_2)}}^{n} P^{k_q}| = |\bigcup_{\substack{q=2:\\ \sigma(k_q)=3}}^{3} P^{k_q}| = |P^3 \cup P^2| = 5$. *No v-variables can be added to the inequality, and conditions 1 and 2 in the theorem are also satisfied, thus producing the facet* $x_1^2 + x_2^2 + x_3^2 + x_5^2 + x_4^3 \leq 1$.

Even though the preprocessing techniques are presented in the following section, in the rest of this section we assume that the decision variables of the richest customer have been preprocessed setting $x_4^1 = 1$, $x_i^1 = 0\ \forall i \neq 4$. Broadly speaking, the reasoning behind this is that since the richest customer can afford any product, he purchases his most preferred one. As we will see in Section 2.5, this can always be applied to the richest customers, and it makes the instance smaller and more manageable.

Table 2.2 shows every clique facet that can be obtained applying Theorem 2.15 to the instance of the RPP given in Figure 2.1. Each row of the table represents a clique, and a variable $x_i^{k'}$ belongs to that clique if and only if the number i appears in column $k = k'$ in that row, for $k' = \{2, \ldots, 5\}$, $i \in I^{k'}$ (variables related to customer 1 do not appear in any clique because they have been previously settled to either zero or one). For instance, the first row of Table 2.2 refers to clique $\sum_{i=1}^{5} x_i^2 \leq 1$, whereas the fourth one makes reference to clique $x_2^2 + x_3^2 + x_5^2 + x_1^5 + x_4^5 \leq 1$.

Theorem 2.17. *Given a nonempty set* $L = \{m_1, \ldots, m_p\} \subseteq M$, *a product* $i \in I$ *and*

- *if* $m_1 > 1$, *a customer* k_1 *such that* $\sigma(k_1) = m_1 - 1$, $i \in I^{k_1}$, *and a set* $P^{k_1} = \{i\}$; *otherwise,* $P^{k_1} = \emptyset$;
- *if* $m_p < |M|$, *customers* $k_2, \ldots, k_n$, $n \geq 2$, *such that* $m_p = \sigma(k_2) \leq \cdots \leq \sigma(k_n)$ ($n = 1$ *otherwise*) *and non empty pairwise disjoint sets of products* $P^{k_q} \subseteq I^{k_q} \setminus \{i\}$, $q = 2, \ldots, n$ *such that* $P^{k_2} \subseteq \overline{B(k_2, i)}$ *and*

$$P^{k_q} \subseteq \left(\bigcap_{\substack{r=1:\\ \sigma(k_r)<\sigma(k_q)}}^{q-1} \overline{B(k_q, P^{k_r})} \right) \bigcap \left(\bigcap_{\substack{r=1:\\ \sigma(k_r)=\sigma(k_q)}}^{q-1} \left(\overline{B(k_q, P^{k_r})} \cup B(k_r, P^{k_r}) \right) \right)$$

$\forall q \in \{3, \ldots, n\}$,

the following inequalities are valid for $(2\mathrm{IM}_1)$:

$$\sum_{m \in L} v_i^m + \sum_{q=1}^{n} \sum_{j \in P^{k_q}} x_j^{k_q} \leq 1. \tag{2.20}$$

Valid inequalities (2.20) *are facets for the previously defined subproblem of* $(2\mathrm{IM}_1)$ *if and only if* $\nexists (k_0, i_0) \in K \times (I^{k_0} \setminus \{i\})$: $\sigma(k_0) \geq m_p$ *satisfying*

1. $i_0 \in B(k_q, P^{k_q})\ \forall q \in \{1, \ldots, n\} : \sigma(k_q) \geq \sigma(k_0)$,

2. $i_0 \in \overline{B(k_0, P^{k_q})}\ \forall q \in \{1, \ldots, n\} : \sigma(k_q) \leq \sigma(k_0)$.

Furthermore, all the clique facets for the subproblem of $(2\mathrm{IM}_1)$ *containing* v*-variables are in family* (2.20).

Proof. Let G2IM $= (V_G, E_G)$ be the intersection graph of the previously defined subproblem of $(2\mathrm{IM}_1)$ and let $Q = (V', E')$ be a clique of G2IM containing v-variables. Taking into account Proposition 2.12(**7**), all v-variables in the same clique must share the subindex, and by Lemma 2.14, all v-variables in the same clique must have consecutive superindices. We represent with $L = \{m_1, \ldots, m_p\}$ this set of consecutive superindices and with i the common subindex. We thus distinguish several cases depending on L:

1. $L = \{1, \ldots, |M|\} = M$.

   Then by Proposition 2.12(**5**) we know that a node x_j^k in the neighborhood of $v_i^1, \ldots, v_i^{|M|}$ must satisfy $\sigma(k) = |M|$ and $j \in \overline{B(k, i)}$. However, as we will see in Section 2.5, the richest customers always purchase their most preferred product. Therefore we have removed all these x-nodes from the intersection graph, i.e., $P^{k_2} = \cdots = P^{k_n} = \emptyset$.

   Since Proposition 2.12(**4**) does not either provide any node adjacent to $v_i^m\ \forall m$, we obtain $P^{k_1} = \emptyset$ and thus the set of nodes $\{v_i^m :\ m \in M\}$ induces a maximal complete subgraph in G2IM.

2. $L = \{m_1, \ldots, |M|\}$ for some $m_1 > 1$.

   As $v_i^m \notin V'\ \forall m \in \{1, \ldots, m_1 - 1\}$, a node adjacent to v_i^m for $m \geq m_1$ but not to $v_i^{m_1-1}$ must belong to the clique. Applying Lemma 2.14 and Proposition 2.12, we know this node corresponds with an x-variable, so there exists a node $x_j^k \in V'$ for some customer k and product j. As in the previous case, Proposition 2.12(**5**) does not provide any node adjacent to $v_i^{|M|}$, thus $P^{k_2} = \cdots = P^{k_n} = \emptyset$. Therefore, node x_j^k must be adjacent to v_i^m for $m \geq m_1$ by Proposition 2.12(**4**), so $j = i$ and $k = k_1$ for a customer k_1: $\sigma(k_1) < m_1$ and $P^{k_1} = \{i\}$. Since $x_i^{k_1}$ is not adjacent to $v_i^{m_1-1}$, also by Proposition 2.12(**4**) $\sigma(k_1) \geq m_1 - 1$, and hence $\sigma(k_1) = m_1 - 1$.

   If we suppose there exists another node $x_j^k \in V'$, then x_j^k must be adjacent to v_i^m $\forall m \geq m_1$ by Proposition 2.12(**4**), and therefore $j = i$. However, x_i^k and $x_i^{k_1}$ are not adjacent for any customer $k \neq k_1$ by (**2**), so the set $\{v_i^m :\ m \geq m_1\} \cup \{x_i^{k_1}\}$ induces a clique in G2IM.

3. $L = \{1, \ldots, m_p\}$ for some $m_p < |M|$.

   Since $v_i^m \notin V'\ \forall m > m_p$, applying Lemma 2.14 and Proposition 2.12 there must exist a node $x_{i_0}^k \in V'$ such that $x_{i_0}^k$ is adjacent to $v_i^{m_p}$ but not to $v_i^{m_p+1}$. Proposition 2.12(**4**) does not provide any node adjacent to v_i^1, hence $P^{k_1} = \emptyset$ and $x_{i_0}^k$ has to be adjacent to v_i^m, $m \leq m_p$, by Proposition 2.12(**5**). Hence, there exists a customer k_2: $\sigma(k_2) \geq m_p$ and a subset of products $P^{k_2} \subseteq \overline{B(k_2, i)}$ such that $i_0 \in P^{k_2}$ and $x_j^{k_2} \in V'\ \forall j \in P^{k_2}$ (taking into account that, by Proposition 2.12(**1**), $x_j^{k_2}$ is adjacent to $x_{j'}^{k_2}\ \forall j \neq j'$). Since $x_{i_0}^{k_2}$ is not adjacent to $v_i^{m_p+1}$, it follows $\sigma(k_2) = m_p$.

Provided that there exist customers k_q, $\forall q \in \{3,\dots,n\}$ such that $\sigma(k_2) \leq \sigma(k_3) \leq \dots \leq \sigma(k_n)$ and sets of products $P^{k_q} \subseteq I^{k_q}$, $P^{k_q} \neq \emptyset$ $\forall q \in \{3,\dots,n\}$, such that $x_j^{k_q} \in V'$ $\forall j \in P^{k_q}$, then by Proposition 2.12**(2)** $P^{k_1},\dots,P^{k_n}$ are pairwise disjoint. Moreover, $P^{k_q} \subseteq I^{k_q} \setminus \{i\}$ $\forall q \in \{3,\dots,n\}$; otherwise, $x_i^{k_q} \in V'$ for some k_q: $\sigma(k_q) \geq m_p$ and is not adjacent to $v_i^{m_p}$ (Proposition 2.12**(4)**), thus V' does not induce a complete graph.

Applying arguments analogous to those of Theorem 2.15, the rest of the conditions stated above must hold.

4. $L = \{m_1,\dots,m_p\}$ for some $m_1 > 1$, $m_p < |M|$.

   Applying arguments analogous to those of the previous items, we can conclude that there exist customers $k_1 \in K$: $\sigma(k_1) = m_1 - 1$, $i \in I^{k_1}$ such that $P^{k_1} = \{i\}$ and $k_2 \in K$: $\sigma(k_2) = m_p$ with $P^{k_2} \subseteq \overline{B(k_2,i)}$, $P^{k_2} \neq \emptyset$. The rest of the conditions also hold applying a reasoning analogous to that of Theorem 2.15.

□

Example 2.18. *By making use of the instance given in Figure 2.1, we are going to give examples of each of the different types of clique facets that can be encountered. We consider $i = 4$ and different possibilities for set L.*

Consider first $L = \{1,2,3,4\}$. Due to condition 1 in Theorem 2.17, $P^{k_1} = \emptyset$, and by condition 2, $n = 1$. Then a facet in the shape of $v_4^1 + v_4^2 + v_4^3 + v_4^4 \leq 1$ is obtained.

If we remove 1 from L, $L = \{2,3,4\}$, now $m_1 = 2$ and condition k_1: $\sigma(k_1) = 1$ and $4 \in I^{k_1}$ applies, giving $k_1 = 5$. Then $P^5 = \{4\}$ and the new facet is $v_4^2 + v_4^3 + v_4^4 + x_4^5 \leq 1$. Removing also 2 from L, $L = \{3,4\}$, facet $v_4^3 + v_4^4 + x_4^4 \leq 1$ is produced.

If, instead, we remove 4 from L, $L = \{1,2,3\}$, by condition 2 a new customer k_2 such that $\sigma(k_2) = m_p = 3$ is required. Customer k_2 can be chosen in the set $\{2,3\}$. If we choose $n = 2$ and $k_2 = 2$, condition 2.a, $P^2 \subseteq \overline{B(2,4)} = \{1,2,3,5\}$, applies. Taking $P^2 = \{1,2,3,5\}$ the new facet is $v_4^1 + v_4^2 + v_4^3 + x_1^2 + x_2^2 + x_3^2 + x_5^2 \leq 1$, since no pair (k_0, i_0) in the conditions of the theorem exists and condition 2.b is also satisfied. If, instead, we choose $n = 3$, $k_2 = 2$ and $k_3 = 3$, a possibility that satisfies all the requirements is to take $P^2 = \{3,5\}$, $P^3 = \{2\}$ to produce the facet $v_4^1 + v_4^2 + v_4^3 + x_3^2 + x_5^2 + x_2^3 \leq 1$. However, if we also remove 3 from L, i.e., $L = \{1,2\}$, 4 is the only customer whose budget meets the requirement $\sigma(k_2) = m_p = 2$, but $\overline{B(4,4)} = \emptyset$, and thus condition 2.a in Theorem 2.17 can not be verified for any customer. Therefore, no clique facet in this instance contains only v-variables v_4^1 and v_4^2.

If we take now $L = \{2,3\}$, again with $k_1 = 5$, $n = 3$, $k_2 = 2$ and $k_3 = 3$, we can obtain facets such as $v_4^2 + v_4^3 + x_4^5 + x_3^2 + x_5^2 + x_2^3 \leq 1$.

Now that we have established the different shapes that clique facets can adopt, we are able to determine whether constraints (2.10b)-(2.10e) always define clique facets in the corresponding subproblem of (2IM$_1$). Thus, we can conclude that constraints (2.10c) and (2.10e) always define clique facets by applying cases 1 and 2 of the proof of Theorem 2.17, respectively. By Theorem 2.15, and given that $B(k, I^k) = \emptyset$ $\forall k$, we know a valid inequality from the family (2.10b) is a clique if and only if $|I^k| \geq 2$ and $\nexists (k_0, i_0) \in K \times I^{k_0}$

	x				v			
	$k=2$	$k=3$	$k=4$	$k=5$	$m=1$	$m=2$	$m=3$	$m=4$
$L=\{m_1,\ldots,m_p\}$	235			1		1	1	
		245	3				3	
	2	45	3				3	
	1235			4		4	4	
	3	25		4		4	4	
	23	5		4		4	4	
	35	2		4		4	4	
	1235		4				4	
	3	25	4				4	
	23	5	4				4	
	35	2	4				4	
	23		5				5	
	3	2	5				5	
$L=\{1,\ldots,m_p\}$	235				1	1	1	
				24	1			
	235			4	1			
				4	2	2	2	
	2	45			3	3	3	
		245			3	3	3	
	2		45		3	3		
		2	45		3	3		
	2	5	4		3	3		
		25	4		3	3		
	3	25			4	4	4	
	35	2			4	4	4	
	23	5			4	4	4	
	1235				4	4	4	
	23				5	5	5	
	3	2			5	5	5	
	23		4		5	5		
	3	2	4		5	5		
$L=\{m_1,\ldots,\lvert M\rvert\}$	1							1
				1		1	1	1
	2							2
		2						2
				2		2	2	2
	3							3
		3						3
			3				3	3
	4							4
		4						4
			4				4	4
				4		4	4	4
	5							5
		5						5
				5			5	5
$L=M$					1	1	1	1
					2	2	2	2
					3	3	3	3
					4	4	4	4
					5	5	5	5

Table 2.3: Facets obtained applying Theorem 2.17 to the instance of the RPP given in Figure 2.1

satisfying $\sigma(k_0) \geq \sigma(k)$ and $i_0 \in \overline{B(k_0, I^k)}$. For example, the valid inequality $\sum_{i\in I^4} x_i^4 \leq 1$ is not a clique facet in the instance of the RPP given by Figure 2.1, since there exists $(3,2) \in K \times I^3$ satisfying $\sigma(3) \geq \sigma(2)$ and $2 \in \overline{B(3, I^4)}$, and therefore x_2^3 is adjacent to x_i^4 $\forall i \in I^4$. As for constraints (2.10d), they do not necessarily define clique facets either: in the same instance, $\sum_{m=1}^{3} v_4^m + \sum_{i\in\overline{B(3,4)}} x_i^3 \leq 1$ is valid but is not a clique facet, because x_3^2 could be added to the LHS and the constraint would still be valid, since the pair $(2,3) \in K \times I^2 \setminus \{4\}$ satisfies $\sigma(2) \geq m_3$ and meets conditions 1 and 2 of Theorem 2.17. But, like in the former case, they define clique facets in most cases. Table 2.3 includes every facet that can be obtained applying Theorem 2.17 to the instance given by Figure 2.1, previously preprocessed setting $x_4^1 = 1$, $x_i^1 = 0\ \forall i \neq 4$.

Even though the valid inequalities given by Theorems 2.15 and 2.17 are facet defining for the subproblem of (2IM$_1$) associated to x- and v-variables and constraints (2.10b)-(2.10e), they might not define facets of the polyhedra obtained once we consider also z-variables and their corresponding constraints of model (2IM$_1$). Nevertheless, they are still strong valid inequalities and, as such, make the extended formulation (2IM$_1$) stronger in turn. As we have previously stated, the same applies to model (2IM$_2$). Additionally, we have incorporated some of these valid inequalities into models (2IM$_1$) and (2IM$_2$), but they do not significantly improve their performance, given that the original models are already tight since they contain mainly inequalities which are facet-defining in the corresponding subproblems, as we have been able to prove through this section. Therefore, in the computational study of Section 2.6 we will test the performance of both models without any additional clique facet of their subproblems.

2.5 Preprocessing techniques

In this section, our aim is to fix x- and v-variables to zero in order to reduce the size of the RPP instances before solving them.

Let us begin by recursively defining a function $u : K \to I$ as follows, for the set of customers according to their budgets in decreasing order:

1. If $\sigma(k) = |M|$, then $u(k) = i$ if and only if $i \in I^k$ and $\nexists j \in I^k : j \prec_k i$.
2. If $\sigma(k) < |M|$ and $\exists i \in I^k$ such that $\forall k' : \sigma(k') > \sigma(k)$, $u(k') \neq i$, then $u(k) = i$ if and only if $i \in I^k$, $\nexists k'$ with $\sigma(k') > \sigma(k)$ such that $u(k') = i$ and $\forall j \in I^k : j \prec_k i$, $\exists k'$, $\sigma(k') > \sigma(k)$, such that $u(k') = j$.
3. If $\sigma(k) < |M|$ and $\forall i \in I^k$, $\exists k'$ with $\sigma(k') > \sigma(k)$ and $u(k') = i$, then $u(k) = i$ if and only if $i \in I^k$ and $\nexists j \in I^k : j \succ_k i$.

Function u assigns, to the richest customers, their most preferred product; and to the rest of the customers, their most preferred product among the ones which have not been previously assigned to any richer customer (or their least preferred one if all of them have already been assigned).

Based on the definition of u, we are going to establish a partition of the set of customers. Thus, let C_r, $r \in \{1,2,3\}$, be such that $k \in C_r$ if and only if $u(k)$ has been defined for k making use of item r of the definition of u. It is clear that $\cup_{r\in\{1,2,3\}} C_r = K$, but given this definition it is possible that both C_2 and C_3 are empty or C_3 is. If $C_2 = C_3 = \emptyset$, then

$\sigma(k) = |M|$ $\forall k \in K$, and the problem becomes trivial: it suffices to establish $v_i^{|M|} = 1$ $\forall i \in I$, every customer purchases his most preferred item and the objective value is the sum of every customer's budget, i.e. $b^{|M|}|K|$. If $C_1 \neq \emptyset \neq C_2$ and $C_3 = \emptyset$, then we will see in Corollary 2.26 that an optimal solution can be found by inspection.

The following result shows the usefulness of this function when fixing x-variables to zero:

Proposition 2.19. *There exist optimal solutions* $(\tilde{v}, \tilde{x})$ *of* (BNLM) *and* (2INLM) *such that* $\tilde{x}_i^k = 0$ $\forall k \in K$, $\forall i \in I^k : i \succ_k u(k)$.

Proof. Suppose we have an optimal solution $(\hat{v}, \hat{x})$ which does not satisfy the statement conditions. By slightly modifying $(\hat{v}, \hat{x})$, we aim at building another solution $(\tilde{v}, \tilde{x})$, with the same objective value, which does satisfy them.

Let us proceed by induction on k. First consider one of the richest customers, $k_0 : \sigma(k_0) = |M|$. Then we know k_0 is able to afford every product he is interested in, and therefore in every optimal solution he purchases his most preferred product. Therefore, $\hat{x}_i^{k_0} = 0$ must hold for all k_0 such that $\sigma(k_0) = |M|$ and $i \in I^{k_0} : i \succ_{k_0} u(k_0)$.

Since $(\hat{v}, \hat{x})$ does not satisfy the statement conditions, there exists $k_0 \in K$ such that $\sigma(k_0) = m_0 < |M|$ and $\forall k$ such that $\sigma(k) > m_0$ $\hat{x}_i^k = 0$ $\forall i \succ_k u(k)$ but $\hat{x}_{i_0}^{k_0} = 1$ for a product $i_0 \succ_{k_0} u(k_0)$. It is clear that $k_0 \in C_2$. The fact that k_0 buys product i_0 implies he cannot afford product $u(k_0)$, i.e., $\sum_{m=1}^{m_0} \hat{v}_{u(k_0)}^m = 0$ and $\hat{x}_{u(k_0)}^{k_0} = 0$. We are going to show that $\hat{x}_{u(k_0)}^k = 0$ $\forall k$, that is to say, that product $u(k_0)$ has not been sold in the considered optimal solution. On the one hand, it is clear that $\hat{x}_{u(k_0)}^k = 0$ for all k such that $\sigma(k) \leq \sigma(k_0)$ because these customers cannot afford it either. On the other hand, let us prove that for all k such that $\sigma(k) > \sigma(k_0)$, it holds $u(k_0) \succ_k u(k)$ or $u(k_0) \notin I^k$. First of all, we know $u(k_0) \neq u(k)$ $\forall k : \sigma(k) > \sigma(k_0)$ because $k_0 \in C_2$. Besides, let us suppose $u(k_0) \prec_{k_1} u(k_1)$ for $k_1 : \sigma(k_1) > \sigma(k_0)$. In this case, $\sigma(k_1) < |M|$, so $k_1 \in C_2 \cup C_3$. But then, by definition of u, $u(k_0) \prec_{k_1} u(k_1) \Rightarrow \exists k_2 : \sigma(k_2) > \sigma(k_1)$ and $u(k_2) = u(k_0)$, which is a contradiction with $k_0 \in C_2$. Therefore, we have proved that customers with budget greater than k_0 do not purchase product $u(k_0)$ because they buy others that prefer more, and customers k such that $\sigma(k) \leq \sigma(k_0)$ cannot afford product $u(k_0)$. Hence, $u(k_0)$ is not sold in this optimal solution.

Let us consider now a price vector $\tilde{v}$ defined by $\tilde{v}_i^m = \hat{v}_i^m$ $\forall m$, $\forall i \neq u(k_0)$ and $\tilde{v}_{u(k_0)}^{m_0} = 1$, $\tilde{v}_{u(k_0)}^m = 0$ $\forall m \neq m_0$. If prices are settled this way, customers k with $\sigma(k) < m_0$ can afford the same products as before, so they purchase the same item. Customers k with $\sigma(k) = m_0$ are now able to afford product $u(k_0)$. However, if they purchase it (because they prefer it over the one they were buying in the previous solution) they spend their whole budget. Therefore, the revenue does not decrease. Further, customers k with $\sigma(k) > m_0$ were already buying a product more preferable than $u(k_0)$ in the previous solution, so they buy the same as previously. Thus, $\tilde{x}_i^k = \hat{x}_i^k$ $\forall k : \sigma(k) \neq m_0$, $\forall i \in I^k$; $\tilde{x}_i^k = \hat{x}_i^k$ $\forall k : \sigma(k) = m_0$ and $u_{k_0} \succ_k j$ for $j : \hat{x}_j^k = 1$, $\forall i \in I^k$; and $\tilde{x}_{u(k_0)}^k = 1$, $\tilde{x}_i^k = 0$ $\forall k : \sigma(k) = m_0$ and $u(k_0) \prec_k j$ for $j : \hat{x}_j^k = 1$, and $\forall i \neq u(k_0)$.

Therefore, through $\tilde{v}$ we have built a feasible solution $(\tilde{v}, \tilde{x})$ with the same objective value as the one given by solution $(\hat{v}, \hat{x})$ and such that $\tilde{x}_i^{k_0} = 0$ $\forall i \succ_{k_0} u(k_0)$. Proceeding by induction on k, we deduce that we can obtain an optimal solution satisfying the statement conditions. □

	Product 1	Product 2	Product 3	Product 4	Product 5	Budgets
Customer 1	-	3	1*	-	2	53
Customer 2	2*	-	1	-	-	40
Customer 3	1*	-	2	4	3	40
Customer 4	4	3	5	2	1*	38
Customer 5	1	-	3	-	2*	32
Customer 6	4	3	1	5	2*	31
Customer 7	1	4*	2	-	3	25
Customer 8	1	-	3	2*	-	25
Customer 9	-	-	2	1*	-	25
Customer 10	2	1*	-	4	3	16

Table 2.4: Preprocessing of the x-variables of Example 2.2

Example 2.20. *To illustrate the above result, we use the instance given in Table 2.1. In Table 2.4, we only show t_i^k from Table 2.1, i.e. the ranking that every customer gives to each product (so the lower the number, the greater the preference over the product). For every customer $k \in K$, t_i^k is circled in the preference matrix provided that $u(k) = i$. If x_i^k is fixed to 0 by Proposition 2.19, then t_i^k appears in grey. If customer k purchases product i in the optimal solution from Table 2.1, t_i^k is marked with an asterisk.*

Now, we present how the preprocessing has been applied for some customers. Since customer 1 is the richest one, by item 1 of the definition of u we obtain that $u(1) = 3$, which is his favorite product. By applying Proposition 2.19, $x_i^1 = 0$ for $i \in \{2, 5\}$. Notice that $u(2) = u(3) = 1$ by item 2 of the definition of u. In the case of customer 2, his most preferred product has been assigned to customer 1. By applying Proposition 2.19, neither customer 2 nor customer 3 purchase any product they like less than product 1. If we turn to customer 5, with budget 32 and $I^5 = \{1, 3, 5\}$, we remark that for each product i in his list of preferences there exists another customer k with budget greater than 32 such that $u(k) = i$ (these are, respectively for products 1, 3 and 5, customers 2, 1 and 4). Therefore, $u(5) = 3$ by item 3 of the definition of u, and no x-variable related to this customer can be set to zero by Proposition 2.19. Furthermore, comparing with the optimal solution displayed in Table 2.1, as expected, in this optimal solution every customer k obtains a product he likes more or the same than product $u(k)$.

Remark 2.21. *Besides being useful to fix variables to zero, the proof of Proposition 2.19 derives an optimal solution $(\tilde{v}, \tilde{x})$ from another solution $(\hat{v}, \hat{x})$ which satisfies $\sum_{i \in I^k} t_i^k \tilde{x}_i^k \leq \sum_{i \in I^k} t_i^k \hat{x}_i^k \ \forall k \in K$, that is, it allows us to obtain an optimal solution in which customers either buy the same product or buy another one they prefer more. It is also remarkable that there may be more than one optimal solution satisfying Proposition 2.19.*

Function u also lets us conclude that some products are not sold in any optimal solution that satisfies Proposition 2.19:

Corollary 2.22. *Let $(\tilde{v}, \tilde{x})$ be an optimal solution of* (BNLM) *or* (2INLM) *satisfying the* *conditions of Proposition 2.19. Then for every product $i \in I$ such that $u^{-1}(i) = \emptyset$, it follows $\tilde{x}_i^k = 0$ for every customer $k \in K$ with $i \in I^k$, i.e., product i is not sold.*

Proof. Let us consider an optimal solution $(\tilde{v}, \tilde{x})$ which meets the requirements given by Proposition 2.19, and a customer k and a product i such that $\hat{x}_i^k = 1$. Then $u(k) = i$ or $i \prec_k u(k)$, and in the last case by definition of u there exists a customer k' with $\sigma(k') > \sigma(k)$ and $u(k') = i$. □

Remark 2.23. *Corollary 2.22 allows us to eliminate $\{x_i^k\}_{k \in K}$ and $\{v_i^m\}_{m \in M}$ for all products i we know will not be sold, thus reducing the size of the problem. Furthermore, after this procedure, and by definition of u, we will always obtain instances of the problem with $|I| \leq |K|$. However, there might still remain products which will not be sold in one or more optimal solutions.*

The following result is useful to fix v-variables to zero, reducing the size of the problem.

Proposition 2.24. *There exist optimal solutions $(\tilde{v}, \tilde{x})$ of* (BNLM) *and* (2INLM) *such that $\forall i$, m : $\nexists k$ with $\sigma(k) = m$ and $i \in I^k$, it follows $v_i^m = 0$.*

Proof. Let us suppose we have an optimal solution $(\hat{v}, \hat{x})$ which does not satisfy the statement requirements, that is, there exist i_0, m_0 such that $\hat{v}_{i_0}^{m_0} = 1$ and $\nexists k$ with $\sigma(k) = m_0$ and $i_0 \in I^k$. We aim at building another solution $(\tilde{v}, \tilde{x})$ in which $\tilde{v}_{i_0}^{m_0} = 0$ and the objective value does not decrease.

First of all, let us suppose product i_0 is sold in solution $(\hat{v}, \hat{x})$, i.e., there exists at least a customer k with $\sigma(k) > m_0$ and $\hat{x}_{i_0}^k = 1$. Let $m_1 = \min\{\sigma(k) : k \in K, \sigma(k) > m_0 \text{ and } \hat{x}_{i_0}^k = 1\}$. Let $\tilde{v}$ be such that $\tilde{v}_i^m = \hat{v}_i^m \; \forall m, \forall i \neq i_0$ and $\tilde{v}_{i_0}^{m_1} = 1$, $\tilde{v}_{i_0}^m = 0 \; \forall m \neq m_1$. If we set prices this way, it is clear that every customer who was previously purchasing product i_0 will continue to purchase it, and will pay b^{m_1} for it instead of b^{m_0}. Moreover, the rest of the customers will also purchase the same products than in solution $(\hat{v}, \hat{x})$. Therefore, we have obtained another solution $(\tilde{v}, \tilde{x})$ with $\tilde{x}_i^k = \hat{x}_i^k \; \forall k \in K, i \in I^k$, and objective value strictly greater than the one given by $(\hat{v}, \hat{x})$, which is a contradiction.

On the other hand, let us assume product i_0 is not sold in solution $(\hat{v}, \hat{x})$. Let $m_1 = \sigma(k_1)$ for a customer $k_1 : i_0 \in I^{k_1}$ and $\sigma(k) \leq \sigma(k_1) \; \forall k : i_0 \in I^k$. Let $\tilde{v}_i^m = \hat{v}_i^m \; \forall m, i \neq i_0$ and $\tilde{v}_{i_0}^{m_1} = 1$, $\tilde{v}_{i_0}^m = 0 \; \forall m \neq m_1$. Given this price vector, it is clear that $\tilde{x}_i^k = \hat{x}_i^k$ $\forall k : \sigma(k) \neq \sigma(k_1)$, $i \in I^k$. As for customers $k : \sigma(k) = \sigma(k_1)$, they will purchase their most preferred product between the one they were purchasing in solution $(\hat{v}, \hat{x})$ and i_0, but if they prefer i_0 they will spend their whole budget. Thus, the objective value of solution $(\tilde{v}, \tilde{x})$ is equal to the one given by solution $(\hat{v}, \hat{x})$ and it satisfies $\tilde{v}_{i_0}^{m_0} = 0$. We obtain the result by induction on i. □

Remark 2.25. *Although optimal solutions satisfying Proposition 2.19 do not necessarily satisfy Proposition 2.24, there exist optimal solutions satisfying both propositions. Furthermore, we can assume that if a variable x_i^k can be fixed to zero in an optimal solution $(\hat{v}, \hat{x})$ according to Proposition 2.19, then i no longer belongs to the list of products of interest of customer k, i.e., $i \notin I^k$, thus fixing more v-variables to zero when applying Proposition 2.24.*

By recursively building function u and using the previous results, x_i^k-variables with $i \succ_k u(k)$ can be removed from all formulations based on v- and x-variables. This implies that $x_{u(k)}^k = 1$ for all richest customers k and their favorite products $u(k)$. Variables in the conditions of Proposition 2.24 can also be removed. In some cases, as shown in the following result, an optimal solution to the problem can be directly obtained from the preprocessing phase:

Corollary 2.26. *If for all customers $k \in K$ with $\sigma(k) < |M|$ an $i \in I^k$ exists such that $\forall k' : \sigma(k') > \sigma(k)$, $u(k') \neq i$, that is, if $C_3 = \emptyset$, an optimal solution can be derived by inspection.*

Proof. Let $(\tilde{v}, \tilde{x})$ be defined as follows: for all $k \in K$, $\tilde{x}_{u(k)}^k = 1$, $\tilde{x}_i^k = 0$ $\forall i \neq u(k)$ and $\tilde{v}_{u(k)}^{\sigma(k)} = 1$, $\tilde{v}_{u(k)}^m = 0$ $\forall m \neq \sigma(k)$; for all $i : u^{-1}(i) = \emptyset$, $\tilde{v}_i^{|M|} = 1$, $\tilde{v}_i^m = 0$ $\forall m < |M|$. We are going to show that solution $(\tilde{v}, \tilde{x})$ is optimal.

First of all, we know by hypothesis that $u(k) = u(k') \Rightarrow \sigma(k) = \sigma(k')$, and therefore $\tilde{v}$ is well defined. Moreover, $\tilde{x}$ is also well defined because for all $k \in K$, $i \prec_k u(k)$ there exists $k' : \sigma(k') > \sigma(k)$ with $u(k') = i$, and thus $\tilde{v}_i^{\sigma(k')} = 1$ and k cannot afford product i. Finally, since in this solution all customers k are purchasing a product for their whole budget $\sigma(k)$, then the objective value is equal to the sum of the budgets of every customer (which is an upper bound), and therefore $(\tilde{v}, \tilde{x})$ is optimal. □

Corollary 2.27. *If $|K| \leq |I|$ and $I^k = I$ $\forall k \in K$, then an optimal solution can be derived by inspection.*

Proof. It suffices to notice that Corollary 2.26 can be applied. □

2.6 Computational results

Computational experiments were carried out in order to compare the different models and check the performance of the valid inequalities proposed in Section 2.3 and the preprocessing techniques described in Section 2.5. The commercial IP solver used through all the testing was Xpress mosel version 4.0.3, on a computer Dell PowerEdge T110 II Server (Intel Xeon E3-1270, 3.40GHz) with 16 GB of RAM.

The reader can find all the results of the computational experiment detailed in several tables, and the most relevant information of the tables is summarized by means of several figures.

To begin with, we performed a first computational study to compare models (BNLM) and (2INLM). Thus, we tested the performance of the linearization of these models by means of z^k- and z_i^k-variables, as well as both linearizations of model (2INLM) including the branch-and-cut algorithm described in Section 2.3.

In this first experiment, the instances include $|K| = 30$ customers whose budgets have been randomly generated independently and uniformly. We consider sets of products of sizes $|I|$ =5, $|I|$ =15 and $|I|$ =25, and lists of products of interest of sizes the 10, 25, 50, 75 and 100% of $|I|$, rounded up. The items included in the lists of products of interest and their order have also been selected independently and uniformly at random, and the

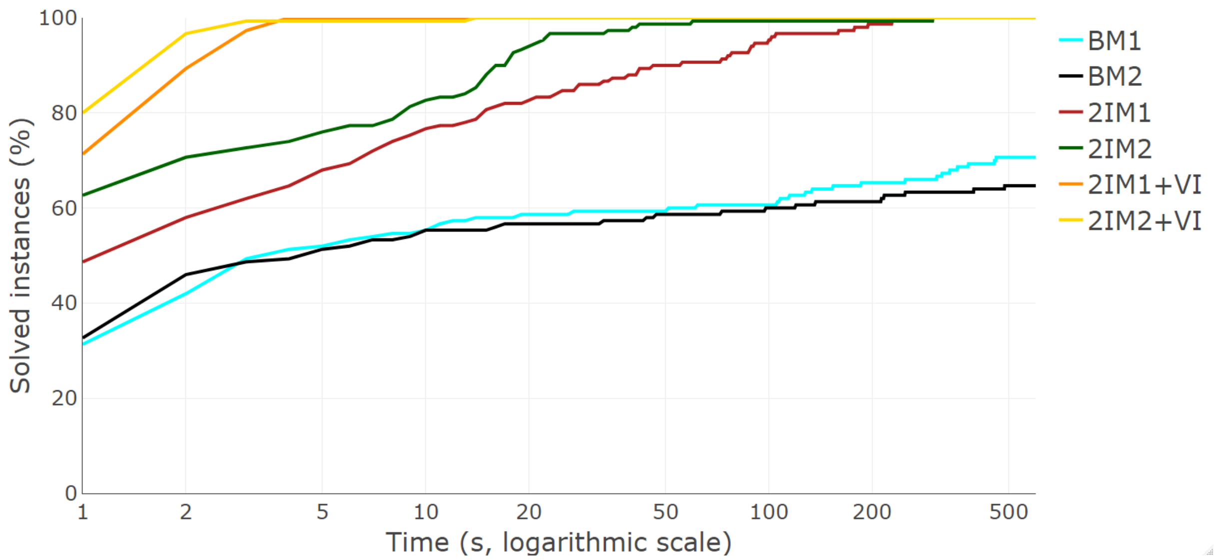

Figure 2.5: Percentage of instances solved with a time limit by models (BM_1), (BM_2), ($2IM_1$), ($2IM_2$) and ($2IM_1$), ($2IM_2$) with the branch-and-cut procedure

number of products of interest is the same for every customer in all the instances. We generated ten instances for each combination of the three mentioned parameters, 150 in total. For the computational study, we have fixed $s_i^k = |I| + 1 - n$ if i is the n-th most preferred product for customer k, $\forall k \in K$, $i \in I^k$.

So as to be able to compare the integrality gaps and resolution times for these models, we disabled automatic cuts and switched Xpress presolve settings off. The time limit for each instance and model was fixed to 600 seconds. The only preprocessing applied to the instances consisted in setting $x_i^k = 1$ for every richest customer k and for every product $i \in I^k$ which is their favorite or first-ranked. In order to check the usefulness of the valid inequalities proposed for formulations ($2IM_1$) and ($2IM_2$) in Section 2.3, we also implemented a branch-and-cut algorithm following the separation procedure explained in Section 2.3. In every node of the branching tree, a fractional solution $(\overline{v}, \overline{x}, \overline{z})$ was obtained after solving the linear relaxation of the corresponding subproblem, and, provided that the depth of this node in the tree was 4 or less, we checked for valid inequalities (2.12) or (2.14), respectively, and re-optimized the subproblem until no more valid inequalities were violated or the linear relaxation bound was no further improved.

Figures 2.5, 2.6 and 2.7 illustrate the results obtained, and refer to Table 2.5. As described in Section 2.3, models (BM_1) and ($2IM_1$) (resp. (BM_2) and ($2IM_2$)) are the linearizations of models (BNLM) and (2INLM) by means of z^k-variables (resp. z_i^k-variables). Models (BM_1), (BM_2), ($2IM_1$), ($2IM_2$), as well as models ($2IM_1$) and ($2IM_2$) with the corresponding branch-and-cut algorithms, appear in the legend of the figures, respectively, as BM1, BM2, 2IM1, 2IM2, 2IM1+VI and 2IM2+VI. Figure 2.5 shows the percentage of instances solved within a given time limit, where the axis of abscissas has been represented using a logarithmic scale. The accumulated percentage of solved instances depending on the number of nodes explored in the branching tree is shown in Figure 2.6, also using a logarithmic scale in the axis of abscissas. And Figure 2.7 shows the percentage of instances which have an integrality gap less than or equal to that of the x-axis. For models (BM_1), (BM_2), ($2IM_1$) and ($2IM_2$), this integrality gap is equal to LRGap $= 100\frac{\text{UB}-\text{OPT}}{\text{OPT}}\%$, where

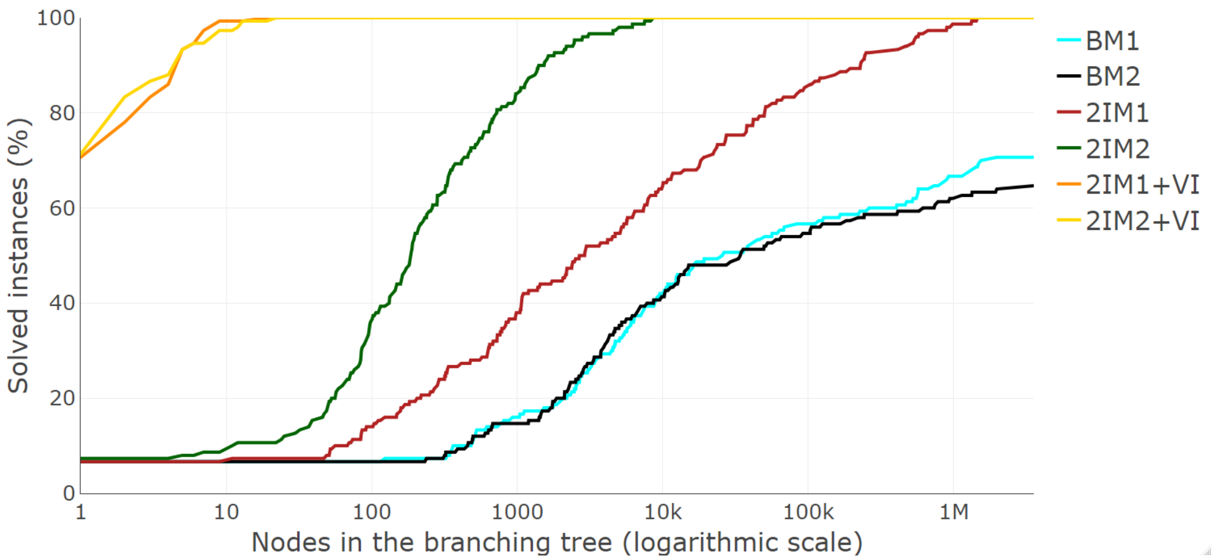

Figure 2.6: Percentage of solved instances depending on the number of nodes explored in the branching tree by models (BM_1), (BM_2), ($2IM_1$), ($2IM_2$) and ($2IM_1$), ($2IM_2$) with the branch-and-cut procedure

UB is the upper bound of the linear relaxation and OPT is the optimal value of the instance. In the case of models ($2IM_1$) and ($2IM_2$) with the corresponding branch-and-cut algorithms, the integrality gap is given by $\text{RGap} = 100\frac{\text{UBC}-\text{OPT}}{\text{OPT}}\%$, where UBC represents the upper bound given by the linear relaxation in which the cuts have been added in the root node.

As we can see in Figure 2.5, models (BM_1) and (BM_2) were only able to solve around the 65% and the 70%, respectively, of the instances proposed within a time limit of 600 seconds. For its part, models ($2IM_1$) and ($2IM_2$) solved all the instances in 300 seconds, and this time is further improved to only a few seconds when adding the branch-and-cut procedure. In fact, we can see how the lines 2IM1+VI and 2IM2+VI of Figure 2.5 are very close to each other and reach 100% almost immediately. In Figure 2.6 we can observe that models ($2IM_1$), (BM_1) and (BM_2) reached the million of nodes explored in the branching tree in some of the instances, and this amount decreases in two orders of magnitude for model ($2IM_2$). Models ($2IM_1$) and ($2IM_2$) with the branch-and-cut algorithm solved the totality of the instances exploring on average less than 10 nodes, highly improving the performance of the other four models. Figure 2.7 shows that models (BM_1) and (BM_2) reached integrality gaps of more than 30% in some instances. The maximum gap reached by model ($2IM_1$) is of around 20%, and this gap was halved when using model ($2IM_2$) and divided by eight when adding the cuts in the root node in models ($2IM_1$) and ($2IM_2$), illustrating how these cuts have a significant importance in the reduction of the integrality gaps.

The results represented in Figures 2.5, 2.6 and 2.7 show that, whilst the linearization of (BNLM) using z^k variables provides slightly better results in terms of time and nodes than the one using z_i^k, the opposite occurs when comparing both linearizations of formulation (2INLM), since model ($2IM_2$) performs clearly better than ($2IM_1$) in terms of both the number of nodes explored in the branching tree and the integrality gaps. The introduction of the branch-and-cut algorithm into the models leads to a considerable improvement in

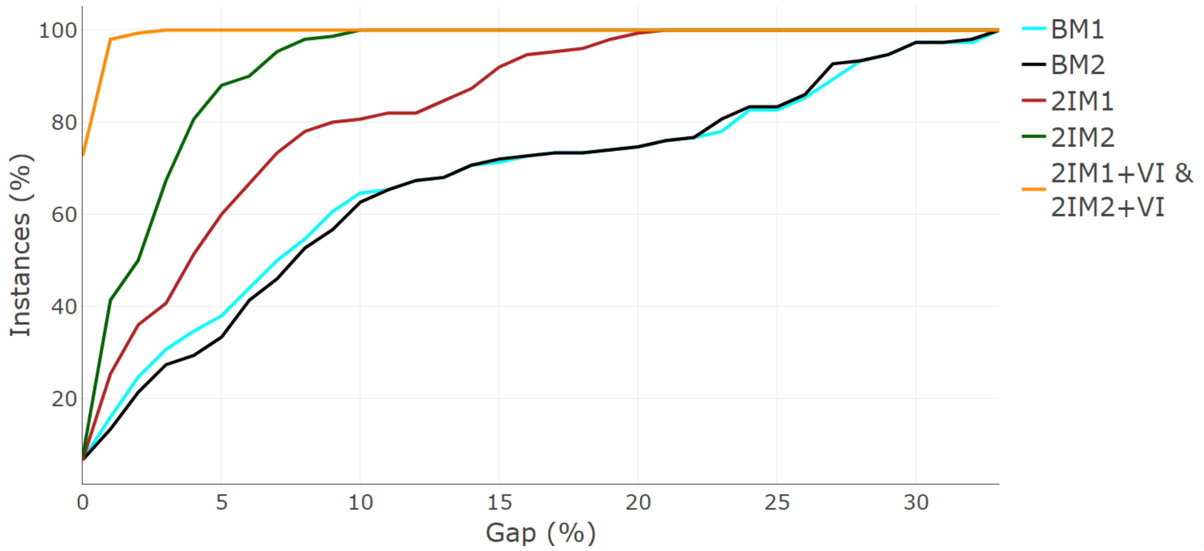

Figure 2.7: In the ordinate axis, the percentage of instances with an integrality gap less than or equal to that of the corresponding abscissas is represented for models (BM_1), (BM_2), ($2\mathrm{IM}_1$), ($2\mathrm{IM}_2$) and ($2\mathrm{IM}_1$) and ($2\mathrm{IM}_2$) with the branch-and-cut procedure

both models.

With the aim of testing the performance of the preprocessing proposed in Section 2.5, we ran the same instances after fixing x- and v-variables to zero by applying Propositions 2.19 and 2.24, respectively, with the six previous models. The results are detailed in Table 2.6, where it can be appreciated the great improvement provided by the preprocessing. The results provided by the two best previous models, ($2\mathrm{IM}_1$) and ($2\mathrm{IM}_2$), both with the branch-and-cut procedure, are represented in Figures 2.8 and 2.9.

Figure 2.8 shows the average time (in seconds, using a logarithmic scale) needed to optimally solve the ten instances previously generated for each number of products ($|I| = 5$, $|I| = 15$ and $|I| = 25$) and each size of the list of products of interest of every customer ($|I^k| = \lceil 0.1|I|\rceil$, $|I^k| = \lceil 0.25|I|\rceil$, $|I^k| = \lceil 0.5|I|\rceil$, $|I^k| = \lceil 0.75|I|\rceil$ and $|I^k| = |I|$). The size of the set of products is included after the letter i in the notation of the instances, and the number of products of interest of every customer appears after the letter s. Regarding the instances, it is noticeable from the results of Figure 2.8 that the difficulty to solve them increases when the number of products in which every customer is interested grows. It is also remarkable that the preprocessing techniques are more efficient in the reduction of the times when the number of products increases: for the instances with 25 products and complete list of products of interest, fixing x- and v-variables to zero according to Propositions 2.19 and 2.24 leads to a reduction in the average resolution times of two and one orders of magnitude for models ($2\mathrm{IM}_1$) and ($2\mathrm{IM}_2$), respectively. This is due to the fact that instances with more products (with respect to the number of customers) lead to the fixing of a greater number of x-variables, which results in the elimination of more v-variables, thus considerably reducing the size of the problem. Finally, we can observe that the average resolution times for the preprocessed instances (green and yellow bars) never exceed five seconds.

The average integrality gaps of the linear relaxation LRGap are represented in Figure 2.9.

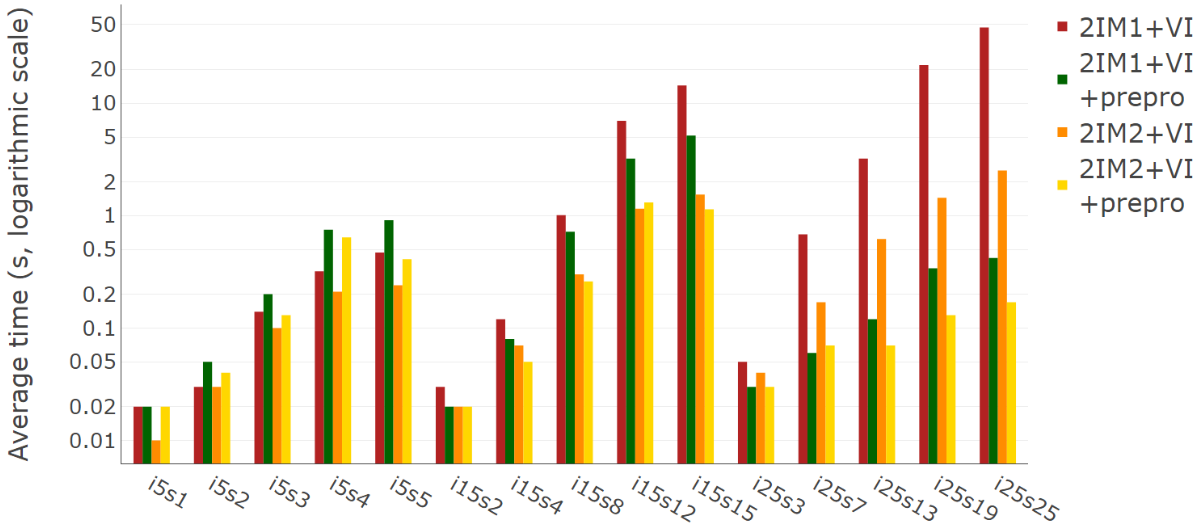

Figure 2.8: Average time needed to solve instances with $|K| = 30$ by models ($2IM_1$) and ($2IM_2$), with and without the preprocessing techniques (ten instances averaged per size)

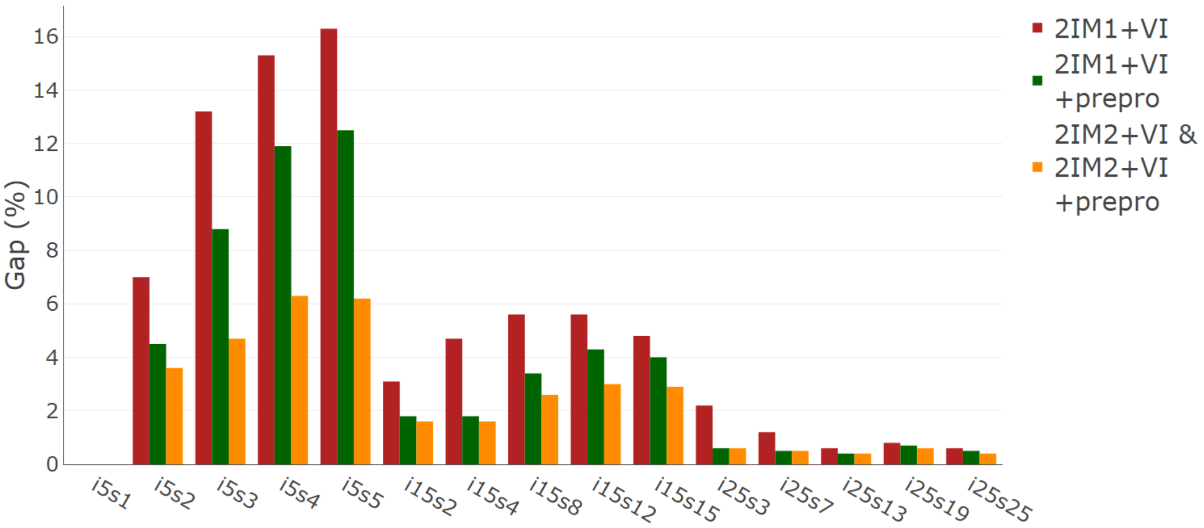

Figure 2.9: Average integrality gaps of the linear relaxation, LRGap, for instances with $|K| = 30$ by models ($2IM_1$) and ($2IM_2$), with and without the preprocessing techniques (ten instances averaged per size)

The greatest integrality gaps are reached when the size of the set of products is small, which may be because these instances have a smaller optimal value. It is noticeable the improvement of the gaps when adding the preprocessing to the model ($2IM_1$), regardless of the number of products and the size of the list of products of interest. Probably due to the small size of the instances, the preprocessing applied to model ($2IM_2$) with the branch-and-cut algorithm does not result in any reduction on the integrality gaps. However, as it will be stated in the second computational study, the preprocessing techniques applied to ($2IM_2$) improve the results when the instances have a bigger size.

Regarding the number of nodes explored in the branching tree, in the majority of the instances only one node is explored, and the average number does not exceed six nodes. Furthermore, the average integrality gaps are reduced to zero in all cases after the cuts in the root node.

$\lvert I\rvert$	$\lvert S^k\rvert$	**(BL$_1$)**				**(BL$_2$)**				**(SLL$_1$)**			**(SLL$_1$) + VI**				**(SLL$_2$)**			**(SLL$_2$) + VI**			
		Nodes	LRGap	t(s)	Sol	Nodes	LRGap	t(s)	Sol	Nodes	LRGap	t(s)	Cuts	Nodes	RGap	t(s)	Nodes	LRGap	t(s)	Cuts	Nodes	RGap	t(s)
5	1	1	0.0	0.0	10	1	0.0	0.0	10	1	0.0	0.0	11.2	1.0	0.0	0.0	1	0.0	0.0	11.1	1.0	0.0	0.0
5	2	486	25.2	0.3	10	566	24.5	0.3	10	198	7.0	0.2	40.8	1.1	0.0	0.0	60	3.6	0.1	27.7	1.3	0.0	0.0
5	3	2543	27.0	0.8	10	2493	26.8	0.7	10	1207	13.2	0.5	85.9	2.7	0.2	0.1	165	4.7	0.2	68.8	2.1	0.2	0.1
5	4	6681	26.2	1.5	10	5824	26.0	1.2	10	7012	15.3	1.2	124.2	3.2	0.5	0.3	282	6.3	0.4	106.4	4.9	0.5	0.2
5	5	9242	25.5	2.3	10	11903	25.3	2.1	10	33559	16.3	4.1	172.7	2.5	0.2	0.5	461	6.2	0.5	157.3	3.3	0.2	0.2
15	2	1413	14.2	0.3	10	1617	14.2	0.3	10	117	3.1	0.1	24.6	1.0	0.0	0.0	81	1.6	0.1	19.0	1.0	0.0	0.0
15	4	35402	8.8	4.0	10	47288	8.8	4.9	10	2091	4.7	0.6	54.9	1.1	0.0	0.1	153	1.6	0.2	34.4	1.0	0.0	0.1
15	8	1316143	7.2	313.2	7	1597499	7.2	331.0	6	133072	5.6	18.7	187.3	2.3	0.1	1.0	737	2.6	0.7	91.3	1.6	0.1	0.3
15	12	1587558	6.6	575.1	1	1803529	6.6	600.0	0	469085	5.6	77.0	454.0	5.0	0.3	7.0	1981	3.0	1.6	207.3	5.8	0.3	1.2
15	15	1232120	5.4	543.0	3	1396684	5.4	600.0	0	474677	4.8	112.2	623.4	4.4	0.2	14.4	3372	2.9	2.4	278.6	3.8	0.2	1.5
25	3	6614	5.1	0.9	10	5825	5.1	0.9	10	229	2.2	0.2	33.3	1.0	0.0	0.1	60	0.6	0.1	20.2	1.0	0.0	0.0
25	7	128768	1.7	16.0	10	1099694	1.7	138.6	9	1224	1.2	0.8	114.7	1.6	0.0	0.7	147	0.5	0.3	45.6	1.3	0.0	0.2
25	13	1684052	0.7	393.0	5	2159784	0.7	525.7	2	3763	0.6	2.3	216.8	1.0	0.0	3.2	352	0.4	0.8	75.2	1.1	0.0	0.6
25	19	1494673	0.9	600.0	0	1342037	0.9	600.0	0	21761	0.8	13.9	606.2	1.5	0.0	21.8	753	0.6	1.7	152.0	1.1	0.0	1.4
25	25	888882	0.6	600.0	0	893893	0.6	600.0	0	27361	0.6	22.1	763.8	1.5	0.0	47.2	641	0.4	1.8	196.5	1.1	0.0	2.5

Table 2.5: Results of the first computational study without the preprocessing techniques. Comparison of models (BL$_1$) and (BL$_2$) with models (SLL$_1$) and (SLL$_2$), as well as models (SLL$_1$) and (SLL$_2$) strengthened with valid inequalities (10 instances averaged per line). All instances have $|K| = 30$ customers, and the table shows the number of products ($|I|$) and the number of products in which every customer is interested ($|S^k|$). Depending on the model, it also includes the number of cuts in the branching tree (Cuts), the number of nodes of the branching tree (Nodes), the integrality gap of the linear relaxation (LRGap), the integrality gap of the linear relaxation after the cuts in the root node (RGap), the execution time in seconds taking into account that the time limit was settled to 600 seconds (t(s)), and the number of instances solved within that time period for the models for which not every instance was solved (Sol)

$\lvert I\rvert$	$\lvert S^k\rvert$	%X	%V	**(BL_1)**			**(BL_2)**			**(SLL_1)**			**(SLL_1) + VI**				**(SLL_2)**			**(SLL_2) + VI**			
				Nodes	LRGap	t(s)	Nodes	LRGap	t(s)	Nodes	LRGap	t(s)	Cuts	Nodes	RGap	t(s)	Nodes	LRGap	t(s)	Cuts	Nodes	RGap	t(s)
5	1	0	76	1	0.0	0.0	1	0.0	0.0	1	0.0	0.0	4.7	1.0	0.0	0.0	1	0.0	0.0	4.7	1.0	0.0	0.0
5	2	6	55	1	17.9	0.2	1	17.5	0.1	1	4.5	0.5	28.3	1.0	0.0	0.1	1	3.6	0.1	27.3	1.0	0.0	0.0
5	3	8	41	11	20.8	1.0	5	20.6	0.5	119	8.8	1.3	69.9	1.0	0.2	0.2	8	4.7	0.7	61.5	1.0	0.2	0.1
5	4	11	25	43	20.3	1.6	18	20.0	1.3	779	11.9	2.2	109.7	2.0	0.5	0.7	46	6.3	1.4	92.4	1.1	0.5	0.6
5	5	13	12	62	19.4	1.8	31	19.2	1.8	3182	12.5	3.4	148.4	2.0	0.2	0.9	72	6.2	1.9	138.8	1.3	0.2	0.4
15	2	17	87	1	3.7	0.0	1	3.7	0.0	1	1.8	0.1	14.7	1.0	0.0	0.0	1	1.6	0.0	15.7	1.0	0.0	0.0
15	4	28	77	1	4.7	0.2	1	4.7	0.1	7	1.8	0.3	31.5	1.0	0.0	0.1	1	1.6	0.2	24.5	1.0	0.0	0.0
15	8	37	62	29	5.7	2.2	2	5.7	1.4	110	3.4	2.2	82.8	1.0	0.1	0.7	2	2.6	1.3	57.9	1.0	0.1	0.3
15	12	42	49	147	5.9	3.6	56	5.9	2.4	2439	4.3	6.5	185.2	1.1	0.3	3.2	40	3.0	2.4	119.9	3.2	0.3	1.3
15	15	45	43	197	5.1	7.5	39	5.1	5.0	2818	4.0	11.0	228.8	1.5	0.2	5.2	88	2.9	4.5	151.2	1.0	0.2	1.1
25	3	39	91	1	1.0	0.0	1	0.9	0.0	0	0.6	0.0	10.8	1.0	0.0	0.0	1	0.6	0.0	9.4	1.0	0.0	0.0
25	7	59	86	0	0.8	0.1	1	0.8	0.1	1	0.5	0.1	18.9	0.9	0.0	0.1	1	0.5	0.1	14	1.0	0.0	0.1
25	13	70	81	0	0.5	0.1	1	0.5	0.2	1	0.4	0.2	25.6	1.0	0.0	0.1	1	0.4	0.2	18.8	1.0	0.0	0.1
25	19	73	77	8	0.8	0.7	1	0.8	0.7	4	0.7	0.9	45.7	1.0	0.0	0.3	1	0.6	0.8	32.2	1.0	0.0	0.1
25	25	77	74	5	0.6	0.5	1	0.6	0.6	9	0.5	0.8	46.3	1.0	0.0	0.4	1	0.4	0.7	32.5	1.0	0.0	0.2

Table 2.6: Results of the first computational study with the preprocessing techniques. Comparison of models (BL_1) and (BL_2) with models (SLL_1) and (SLL_2), as well as models (SLL_1) and (SLL_2) strengthened with valid inequalities, applying the preprocessing described in Section 2.5 to all the instances (10 instances averaged per line). All instances have $|K| = 30$ customers, and the table shows the number of products ($|I|$), the number of products in which every customer is interested ($|S^k|$) and the average percentage of x- and v-variables fixed to zero during the preprocessing ((%X) and (%V), respectively). Depending on the model, it also includes the number of cuts in the branching tree (Cuts), the number of nodes of the branching tree (Nodes), the integrality gap of the linear relaxation (LRGap), the integrality gap of the linear relaxation after the cuts in the root node (RGap) and the average time needed to optimally solve the instances (t(s))

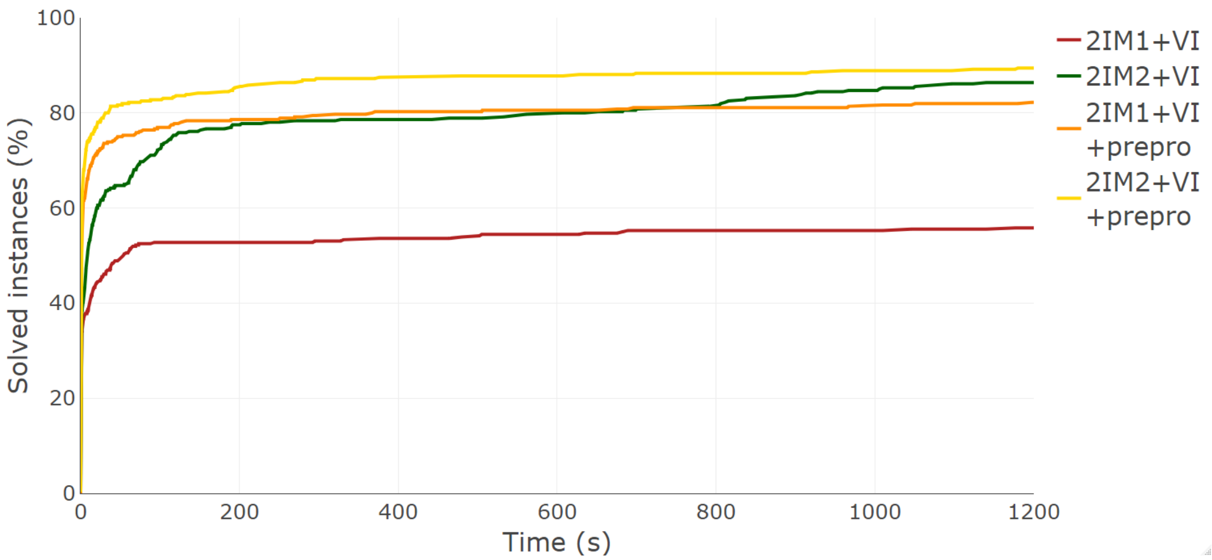

Figure 2.10: Percentage of instances solved with a time limit by models ($2IM_1$) and ($2IM_2$), with and without the preprocessing techniques detailed in Section 2.5

Considering the results of the first computational experiment, we generated instances of bigger and more varied sizes and discarded the models derived from (BNLM). We extended the time limit for each instance and model to 1200 seconds as well. In order to generate the instances of our second computational study, we designed a model based on the Characteristics Model proposed by Fernandes et al. in Fernandes et al. (2016). This model has an economic interpretation, and focuses on the idea that each product has a profile of characteristics, and each customer is interested in several of them. In this way, a product is more preferred by a customer than another provided that more of its characteristics, or the most important ones, are among the ones he desires.

Let C be the set of characteristics, o the number of options for any characteristic and p the number of options in which a customer is interested for any characteristic. The characteristics of every product i are represented by means of a vector of options $E^i = (e^i_c)$, $c \in C$, whose entries are in the set $\{1, 2, \ldots, o\}$, chosen independently and uniformly at random. The set of characteristics in which a customer k is interested is represented by a matrix $A^k_{|C|\times o} = (a^k_{cv})$ where, for every row, p positions are set independently and uniformly at random to 1 (the ones in which k is interested) and $o-p$ positions to 0. The relevance of each characteristic c is determined by its weight $w(c)$, so that $w(c) > w(c')$ if characteristic c is considered (for every customer) more important than c'. In this way, the score each customer k gives to a product i is defined as the sum of the weights of the characteristics of i in which k is interested, i.e., $\text{score}^k_i := \sum_{c\in C} w(c) a^k_{ce^i_c}$. The preferences of a customer are based on the score he has given to each product, since customer k prefers product i over product j if and only if $\text{score}^k_i > \text{score}^k_j$. Each customer is interested in $s \in \{1, \ldots, |I|\}$ products; therefore, if $s < |I|$, the set of products of interest of every customer only includes the s products with the greatest scores for each of them. Note that the lists of products of interest have equal size s for all the customers in all the instances.

The instances for the computational experiment were generated fixing the number of options of each characteristic as $o = 8$; the number of options preferred by each customer

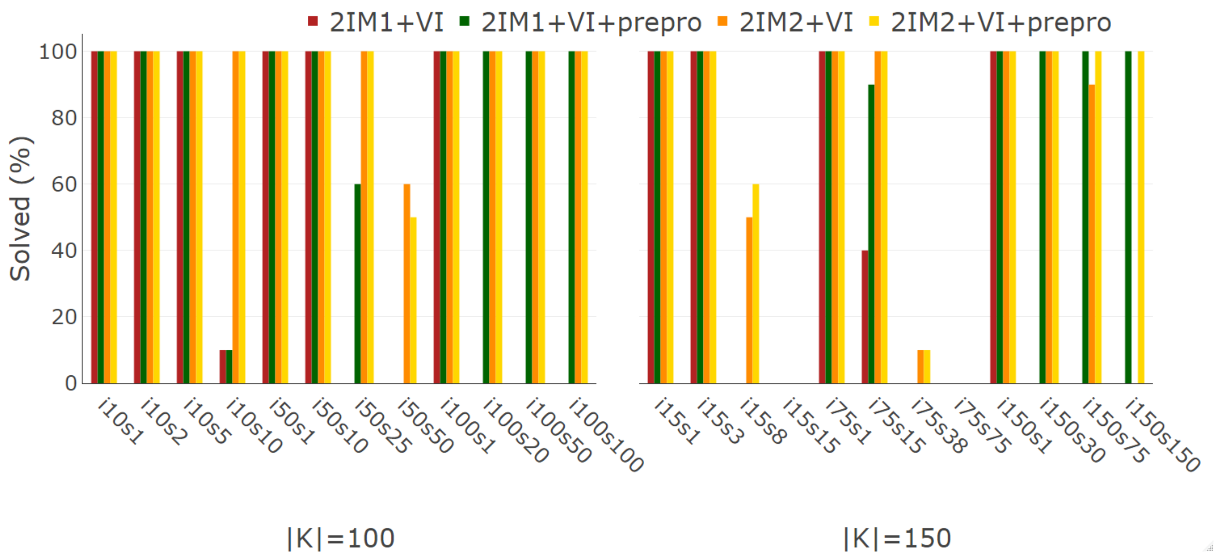

Figure 2.11: Percentage of solved instances depending on their size. Instances with 100 customers are shown in the graphic of the left, and instances with 150 customers, in the graphic of the right. The size of the set of products is included after the letter i in the notation of the instances, and the number of products of interest of every customer appears after the letter s

as $p = 7$; the number of characteristics as $|C| = 50|I|$; and the budgets of the customers are integers randomly selected between 1 and $2|K|$. With the aim of testing the performance of the models (2IM_1) and (2IM_2) with the branch-and-cut procedure, with and without preprocessing, using instances of different sizes and densities, we generated instances of $|K|$ =50, $|K|$ =100 and $|K|$ =150 customers and $0.1|K|$, $0.5|K|$ and $|K|$ products. We generated 10 instances of each size, 360 in total. Once the customers (including their budgets and scores for each product) and products were randomly generated following the previously described procedure, we generated four different instances by modifying s, that is, considering $s = 1$, $s = \lceil 0.2|I| \rceil$, $s = \lceil 0.5|I| \rceil$ and $s = |I|$.

Figures 2.10, 2.11 and 2.12 illustrate the results obtained. We have included a detailed description of the results in which these figures are based in Tables 2.7 and 2.8.

Figure 2.10 shows the percentage of instances solved within a given time limit. In this figure we can observe how model (2IM_1) performed clearly worse than the rest of the models, not reaching the 60% of solved instances. Models (2IM_2) and (2IM_1) and (2IM_2) with the preprocessing techniques had a similar behavior, the three of them solving more than the 80% of instances in less than 1200 seconds. Model (2IM_2) with the preprocessing techniques offered the best results, reaching the 89%. We can also notice that model (2IM_1) with the preprocessing outperformed model (2IM_2) without it at the beginning, but it performed worse after approximately 800 seconds.

Figure 2.11 represents the percentage of instances solved attending to their size. The graphic of the left shows the instances with $|K| = 100$, and the graphic of the right, the instances with $|K| = 150$, given that models (2IM_1) with the preprocessing techniques and (2IM_2) with and without the preprocessing techniques solved all the instances of 50 customers. The size of the set of products ($|I|$) is indicated by the number that

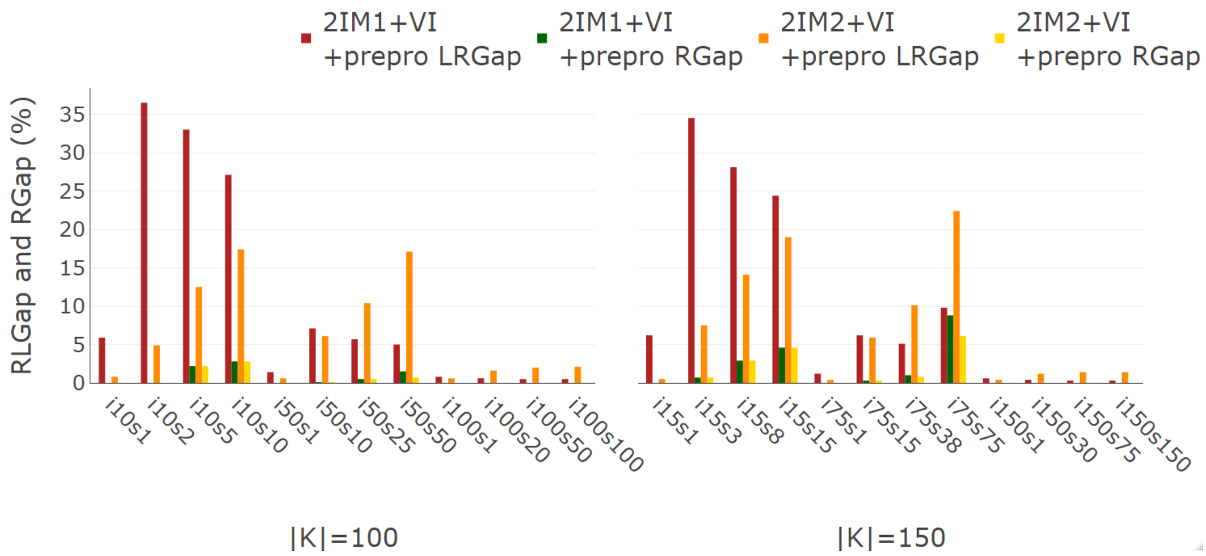

Figure 2.12: Average integrality gaps for models ($2IM_1$) and ($2IM_2$) with the preprocessing techniques. Instances with 100 customers are shown in the graphic of the left, and instances with 150 customers, in the graphic of the right

follows letter i in the notation of the instances; the size of the set of products in which a customer is interested ($|I^k|$), by the number after the letter s. As we had noted in the first computational study, the difficulty of the instances grows when the size of I^k increases. In particular, instances with complete list of products of interest are the most difficult ones, and none of them was solved in less than 1200 seconds for $|K| = 150$ and 15 and 75 products. For the instances with the same number of customers and products, the preprocessing makes a great improvement. This is specially noticeable when $|K| = |I| = |I^k| = 150$, where adding the preprocessing led to the resolution of all the instances, taking into account that none of them had been solved without it.

Figure 2.12 shows the integrality gaps for both models with the preprocessing techniques. The LRGap is calculated using the objective value of the best solution found by any of the models (OV): LRGap $= 100\frac{\text{UB}-\text{OV}}{\text{OV}}\%$, and RGap $= 100\frac{\text{UBC}-\text{OV}}{\text{OV}}\%$, where UBC represents the upper bound given by the linear relaxation in which the cuts have been added in the root node. The notation used to express the size of the instances is the same as the notation of Figure 2.11. It can be observed how, even though there were sizes of instances for which the average LRGap was smaller for model ($2IM_1$) than for ($2IM_2$), the cuts in the root node were more effective in the reduction of the upper bound for model ($2IM_2$) regardless of the case, since the RGap is smaller in this model. In most cases, the reduction of the integrality gap by the branch-and-cut procedure was crucial in the resolution of the instances. It is also remarkable that, for $|K| = |I|$, the LRGap did not reach the 3%, and the inclusion of the cuts reduced it to zero.

$\|K\|$	$\|I\|$	$\|S^k\|$	**(SLL$_1$) + VI**						**(SLL$_1$) + VI + preprocessing**							
			Cuts	Nodes	LRGap	RGap	t(s)	Sol	%X	%V	Cuts	Nodes	LRGap	RGap	t(s)	Sol
50	5	1	68	1	8.4	0.0	0.1	10	0	76	65	1	7.4	0.0	0.0	10
50	5	2	132	2	44.7	0.5	0.4	10	4	56	124	2	38.8	0.5	0.3	10
50	5	3	227	5	39.9	0.8	1.9	10	5	37	216	4	35.8	0.8	1.6	10
50	5	5	516	9	33.5	0.9	8.1	10	7	6	474	13	32.1	0.9	6.9	10
50	25	1	77	1	3.7	0.0	0.1	10	0	95	66	1	2.1	0.0	0.1	10
50	25	5	318	1	10.5	3.1	1.8	10	33	84	186	1	7.6	0.0	0.2	10
50	25	13	1338	35	7.8	7.8	29.9	10	41	66	541	16	7.6	0.3	10.7	10
50	25	25	2331	64839	6.2	6.2	812.3	5	45	45	1286	99	6.2	0.3	91.3	10
50	50	1	85	1	2.3	0.0	0.1	10	0	97	75	1	1.2	0.0	0.1	10
50	50	10	823	5033	1.3	1.3	22.2	10	73	93	137	1	1.2	0.0	0.1	10
50	50	25	358	252303	1.0	1.0	1184.0	1	87	92	140	1	1.0	0.0	0.1	10
50	50	50	282	143773	1.0	1.0	1200.0	0	93	92	144	1	1.0	0.0	0.2	10
100	10	1	134	1	7.3	0.0	0.1	10	0	87	130	1	5.9	0.0	0.1	10
100	10	2	270	1	44.1	0.0	0.6	10	4	76	243	1	36.5	0.0	0.4	10
100	10	5	1130	735	34.9	2.2	40.6	10	7	47	999	815	33.0	2.2	43.7	10
100	10	10	3348	2944	27.1	7.6	1130.7	1	8	8	3099	2969	27.1	2.8	1180.7	1
100	50	1	158	1	3.1	0.0	0.2	10	0	98	133	1	1.4	0.0	0.2	10
100	50	10	1797	8	7.5	7.5	61.2	10	38	85	874	5	7.1	0.1	14.5	10
100	50	25	3821	3790	5.7	5.7	1200.0	0	44	68	2790	532	5.7	0.5	834.8	6
100	50	50	1157	13096	5.0	5.0	1200.0	0	47	45	4841	0	5.0	1.5	1200.0	0
100	100	1	172	1	1.8	0.0	0.3	10	0	99	149	1	0.8	0.0	0.2	10
100	100	20	1507	8133	0.6	0.6	1200.0	0	84	96	297	1	0.6	0.0	0.4	10
100	100	50	969	3365	0.5	0.5	1200.0	0	92	95	314	1	0.5	0.0	0.8	10
100	100	100	1139	2075	0.5	0.5	1200.0	0	96	95	315	1	0.5	0.0	0.8	10
150	15	1	206	1	7.7	0.0	0.2	10	0	92	197	1	6.2	0.0	0.1	10
150	15	3	730	62	39.3	0.7	10.7	10	5	77	651	77	34.5	0.7	9.0	10
150	15	8	3320	2443	28.5	2.9	1200.0	0	8	45	2949	1900	28.1	2.9	1200.0	0
150	15	15	1821	8562	24.4	24.4	1200.0	0	9	8	7746	0	24.4	4.6	1200.0	0
150	75	1	237	1	2.7	0.0	0.4	10	0	98	203	1	1.2	0.0	0.3	10
150	75	15	5831	14	6.2	6.2	1022.6	4	41	86	2146	713	6.2	0.3	493.5	9
150	75	38	2094	807	5.1	5.1	1200.0	0	46	69	5447	0	5.1	1.0	1200.0	0
150	75	75	1316	331	9.8	9.8	1200.0	0	48	47	3542	0	9.8	8.8	1200.0	0
150	150	1	259	1	1.7	0.0	0.5	10	0	99	228	1	0.6	0.0	0.6	10
150	150	30	1380	1325	0.4	0.4	1200.0	0	88	97	445	1	0.4	0.0	1.4	10
150	150	75	1600	139	0.3	0.3	1200.0	0	94	97	462	1	0.3	0.0	2.0	10
150	150	150	357	2	0.3	0.3	1200.0	0	97	96	464	1	0.3	0.0	2.9	10

Table 2.7: Results obtained for model (SLL$_1$) strengthened with valid inequalities, without and with the preprocessing described in Section 2.5, for instances of 50, 100 and 150 customers (10 instances averaged per line). The table includes the number of customers of the instance ($|K|$), the number of products ($|I|$) and the number of products in which every customer is interested ($|S^k|$) and, in the model which includes preprocessing, it also shows the average percentage of x- and v-variables fixed to zero during the preprocessing ((%X) and (%V), respectively). For each model, the table shows the average number of cuts in the branching tree (Cuts), the number of nodes of the branching tree (Nodes), the average integrality gap of the linear relaxation (LRGap), the average integrality gap of the linear relaxation after the cuts in the root node (RGap), the execution time in seconds taking into account that the time limit was settled to 1200 seconds (t(s)) and the number of instances solved within that time period (Sol)

$\lvert K\rvert$	$\lvert I\rvert$	$\lvert S^k\rvert$	**(SLL$_2$) + VI**						**(SLL$_2$) + VI + preprocessing**							
			Cuts	Nodes	LRGap	RGap	t(s)	Sol	%X	%V	Cuts	Nodes	LRGap	RGap	t(s)	Sol
50	5	1	69	1	1.6	0.0	0.0	10	0	76	72	1	1.6	0.0	0.0	10
50	5	2	168	1	7.4	0.5	0.3	10	4	56	170	2	7.3	0.5	0.3	10
50	5	3	293	4	10.6	0.8	1.4	10	5	37	273	3	10.4	0.8	1.5	10
50	5	5	576	6	15.6	0.9	4.1	10	7	6	524	8	15.1	0.9	3.5	10
50	25	1	61	1	1.3	0.0	0.1	10	0	95	67	1	1.3	0.0	0.1	10
50	25	5	519	1	7.7	0.0	0.4	10	33	84	295	1	5.9	0.0	0.2	10
50	25	13	1777	13	17.0	0.3	6.0	10	41	66	767	8	11.2	0.3	2.2	10
50	25	25	4213	19	29.5	0.3	18.0	10	45	45	1651	12	17.6	0.3	7.5	10
50	50	1	57	1	1.1	0.0	0.1	10	0	97	76	1	1.1	0.0	0.1	10
50	50	10	1377	1	9.9	0.0	1.1	10	73	93	226	1	2.7	0.0	0.1	10
50	50	25	4623	1	24.3	0.0	6.7	10	87	92	268	1	3.1	0.0	0.1	10
50	50	50	11186	1	48.5	0.0	29.2	10	93	92	282	1	3.2	0.0	0.1	10
100	10	1	144	1	0.8	0.0	0.1	10	0	87	147	1	0.8	0.0	0.1	10
100	10	2	338	1	4.9	0.0	0.3	10	4	76	333	1	4.9	0.0	0.2	10
100	10	5	1292	173	12.7	2.2	16.6	10	7	47	1191	513	12.5	2.2	22.5	10
100	10	10	3122	2308	17.9	2.8	112.9	10	8	8	2754	4378	17.4	2.8	164.8	10
100	50	1	123	1	0.6	0.0	0.2	10	0	98	135	1	0.6	0.0	0.2	10
100	50	10	2528	4	8.1	0.1	7.8	10	38	85	1232	3	6.1	0.1	3.4	10
100	50	25	8345	247	16.2	0.5	125.9	10	44	68	3442	901	10.4	0.5	113.8	10
100	50	50	20934	719	29.3	0.7	900.5	6	47	45	8237	1956	17.1	0.7	815.3	5
100	100	1	118	1	0.6	0.0	0.2	10	0	99	149	1	0.6	0.0	0.2	10
100	100	20	6849	1	9.9	0.0	13.2	10	84	96	543	1	1.6	0.0	0.3	10
100	100	50	23424	1	24.5	0.0	123.3	10	92	95	638	1	2.0	0.0	0.5	10
100	100	100	58245	1	49.0	0.0	803.2	10	96	95	676	1	2.1	0.0	0.6	10
150	15	1	218	1	0.5	0.0	0.2	10	0	92	220	1	0.5	0.0	0.1	10
150	15	3	942	32	7.6	0.7	7.6	10	5	77	870	28	7.5	0.7	6.2	10
150	15	8	3476	9046	14.3	2.9	810.4	5	8	45	3004	20033	14.1	2.9	938.9	6
150	15	15	7548	5571	19.5	4.6	1200.0	0	9	8	6619	2463	19.0	4.6	1200.0	0
150	75	1	183	1	0.4	0.0	0.3	10	0	98	207	1	0.4	0.0	0.3	10
150	75	15	6569	118	8.0	0.3	69.8	10	41	86	2915	120	5.9	0.3	29.3	10
150	75	38	22443	418	16.1	0.8	1135.0	1	46	69	8638	1602	10.1	0.8	1108.3	1
150	75	75	55813	0	35.3	6.2	1200.0	0	48	47	20458	299	22.4	6.1	1200.0	0
150	150	1	174	1	0.4	0.0	0.4	10	0	99	230	1	0.4	0.0	0.5	10
150	150	30	17432	1	10.1	0.0	66.7	10	88	97	876	1	1.2	0.0	1.0	10
150	150	75	60096	1	25.3	0.0	998.2	9	94	97	987	1	1.4	0.0	1.2	10
150	150	150	122181	0	50.6	16.7	1200.0	0	97	96	1006	1	1.4	0.0	1.3	10

Table 2.8: Results obtained for model (SLL$_2$) strengthened with valid inequalities, without and with the preprocessing described in Section 2.5, for instances of 50, 100 and 150 customers (10 instances averaged per line). The table includes the number of customers of the instance ($\lvert K\rvert$), the number of products ($\lvert I\rvert$) and the number of products in which every customer is interested ($\lvert S^k\rvert$) and, in the model which includes preprocessing, it also shows the average percentage of x- and v-variables fixed to zero during the preprocessing ((%X) and (%V), respectively). For each model, the table shows the average number of cuts in the branching tree (Cuts), the number of nodes of the branching tree (Nodes), the average integrality gap of the linear relaxation (LRGap), the average integrality gap of the linear relaxation after the cuts in the root node (RGap), the execution time in seconds taking into account that the time limit was settled to 1200 seconds (t(s)) and the number of instances solved within that time period (Sol)

Chapter 3

The Rank Pricing Problem with Ties

In this chapter, we consider a generalization of the RPP in which customers are not forced to define a strict preference between all the pairs of candidate products. Instead, we allow for indifference and consider ties in the list of preferences. We name this problem the *Rank Pricing Problem with Ties* (RPPT).

Considering ties in the preference lists of the customers leads to a different bilevel structure of the problem. As detailed in Chapter 2, in the RPP (without ties) the second level problem associated to each customer has a unique optimal solution for a given vector of prices of the products. However, in this extension, the indifference results in second level problems that may have multiple optimal solutions. In the RPPT, we consider the *pessimistic optimal solution*. In case of indifference, the customers' selection of products is the most *natural* for the customers since it is based on the price - they purchase (one of) the cheapest products. Although in the RPPT we consider a positive budget per customer, all our results apply to the extension where customers are endowed with a product-dependent reservation price that reflects their willingness to pay (WTP). The slight changes required are detailed in Section 3.1.

To tackle the problem from the field of discrete optimization, we begin with a formal introduction of the RPPT and propose an integer linear formulation with three-index variables. Next, we derive two exact resolution methods for our three-index formulation.

The first method is based on a second formulation for the RPPT that uses a small set of two-index variables. Since the linear relaxation of this reduced model provides a weaker bound, we project out the variables of the three-index formulation by means of Farkas' Lemma to obtain a set of valid inequalities that strengthen the two-index model. Due to the particular structure of the RPP, the separation problem relative to these valid inequalities can be transformed into a min-cost flow problem. In this way, we avoid solving a linear problem with a commercial solver and instead apply a suitable resolution algorithm, making the separation procedure computationally efficient.

The second resolution approach is motivated by a very recent work by Bertsimas and Mišić (2019). The authors study the Product Line Design (PLD) problem under a ranking-based customer choice model (see Section 1.2.3 from Chapter 2). They introduce a new mixed-integer formulation, theoretically analyze it, and present a solution approach based on Benders decomposition that significantly outperforms the previous results. Thus, our second resolution approach is based on Benders decomposition and takes advantage of the structure of the problem. First we reformulate the three-index model, obtaining a master problem with a set of constraints whose separation can be done by solving linear subproblems. Then we are able to identify a small (polynomial) subset of constraints from the previous set to obtain a reduced master formulation that constitutes a valid formulation for the RPPT. The rest of the constraints (now valid inequalities) are separated in our resolution method and included dynamically, in a branch-and-cut framework. Although the valid inequalities are different from the reduced model ones, the separation procedure is analogous to the first one. To speed up the cut separation in the linear relaxation phase, we include an in-out method, a technique used to stabilize and accelerate the convergence of the cut loop.

We also provide a preprocessing techniques section where we reduce the size of the instances by making use of the properties of the problem. We conclude the chapter with the results of extensive numerical experiments, where we compare the two resolution methods proposed in terms of number of nodes of the branching tree, integrality gap and computational time, and we show the efficiency of the valid inequalities and the preprocessing techniques.

The chapter is organized as follows. Section 3.1 states the notation used throughout the chapter and Section 3.2 is devoted to the presentation of the three-index model. Section 3.3 includes all the results regarding the two-index formulation. In Section 3.4, we provide the results concerning the Benders decomposition resolution approach. Section 3.5 includes the preprocessing techniques, and Section 3.6 contains the computational study.

3.1 Notation

The aim of the RPPT is to establish the prices of the products of the company so as to maximize its revenue, taking into account that we assume unit-demand customers with a positive budget who, once the prices are settled, will purchase their highest-ranked product among the ones they can afford (if any). Besides, if a customer is indifferent between two products and he can afford both, he will purchase the cheapest one (or one of the cheapest if there are more than one). We consider an unlimited supply of products, which is reasonable provided that the company owns at least as many copies of a product as there are customers or when products can be produced quickly at negligible cost.

As for the notation, like in Chapter 2, K denotes the set of customers and I denotes the set of products, with I^k the set of acceptable products for k and $K_i = \{k \in K : i \in I^k\}$. Likewise, the budgets of the customers $B = (b^m)_{m \in M}$ are represented through the previously defined function $\sigma : K \to M$, so $\sigma(k) = m$ if the budget of customer k is the m-th smallest budget b^m.

The acceptable products for k (i.e. the products in I^k) are ranked by k from the *best* to the *worst* in a preference list. However, some customers may not be able to define a clear strict preference over certain products, and they are allowed to express indifference in their preference lists. Therefore, there exists a weak order on the set I^k for each $k \in K$. We denote $i \prec_k j$ when we say that a customer $k \in K$ *prefers* product i to j, and we use $i \sim_k j$ if k is *indifferent between* two products i and j. Naturally,

Table 3.1: Preference matrix, vector of budgets and an optimal solution to an instance of the RPPT

	Prod. 1	Prod. 2	Prod. 3	Prod. 4	Prod. 5	Budgets
Customer 1	1	3	1*	-	2	120
Customer 2	2	1*	1	-	-	95
Customer 3	1	2	4	1*	3	82
Customer 4	-	3	1	3	2*	82
Customer 5	-	1	3	2*	-	79
Customer 6	2	-	1	2	1*	65
Customer 7	3	2	5	1	4*	64
Customer 8	1	4	2	-	3*	53
Optimal prices	-	95	120	79	53	585

$\{j \in I^k : j \preceq_k i\} = \{j \in I^k : j \sim_k i\} \cup \{j \in I^k : j \prec_k i\}$. Furthermore, $\sim_k$ is an equivalence relation (reflexive, symmetric, transitive) which defines a partition $\mathscr{S}^k = \{S_1^k, \ldots, S_{n^k}^k\}$ of the set I^k such that $i, j \in S_n^k$ if $i \sim_k j$ and $i \in S_n^k$, $j \in S_{n'}^k$ with $n < n'$ if $i \prec_k j$. Notice that for a given customer k, $\prec_k$ defines a total order on the set of the equivalence classes $\mathscr{S}^k$.

As explained in Chapter 2, there is always an optimal solution of the RPP in which the prices of the products are equal to a customer budget b^m, $m \in M$. Furthermore, in Section 2.5 we showed that the set of candidate prices of a product could be further reduced. Indeed, there always exists an optimal solution in which the optimal price of a product i is one of the budgets from the set $\{b^m \in B : \exists k \in K_i \text{ such that } \sigma(k) = m\}$. Since this reasoning also applies to the RPPT, we define $M_i := \{m \in M : \exists k \in K_i \text{ with } \sigma(k) = m\}$ as the set of indices of budget values that are candidates to be the optimal price of product i. Moreover, for $k \in K_i$, $M_i^k := \{m \in M_i : m \leq \sigma(k)\}$ represents the set of indices m of candidate prices b^m at which k can purchase i in a feasible solution. Finally, we define $M_{S_n^k} = \cup_{i \in S_n^k} M_i^k$ as the subset of indices $m \in M$ of candidate prices b^m at which k could purchase some product $i \in S_n^k$.

Although for simplicity in the notation we have considered a positive budget per customer, the formulations and algorithms presented in the following sections can be applied to a more general framework where customers are endowed with product-dependent reservation prices. It suffices to redefine M as the set of indices that refer to the different reservation prices of all the customers for all the products, and order all the different reservation prices $\{b^m\}_{m \in M}$ like previously. Next, we define a set of functions $\sigma_i : K_i \to M$ (instead of σ) such that $\sigma_i(k) = m$ if the reservation price of k for i is the m-th smallest reservation price. In this way, the sets M_i and M_i^k would naturally correspond with $M_i := \{m \in M : \exists k \in K_i \text{ with } \sigma_i(k) = m\}$ and $M_i^k := \{m \in M_i : m \leq \sigma_i(k)\}$. Our results are valid assuming that $\sigma_i(k) = \sigma_j(k)$ for any two products $i, j : i \sim_k j$, that is, customers have the same reservation price for the products among which they are indifferent.

Example 3.1. *Table 3.1 shows an instance of the* RPPT *with* $|K| = 8$ *and* $|I| = 5$ *and an optimal solution. The entry* (k, i) *of the preference matrix denotes the index* n *of the equivalence class* S_n^k *to which* i *belongs for* k *(the symbol - indicates that the corresponding product* $i \notin I^k$*). Clearly, the smaller the entry of the preference matrix, the greater the preference of the customer over that product. Customer 1 is thus interested in all products except for product 4, that is,* $I^1 = \{1, 2, 3, 5\}$*, and from the preference matrix we deduce* $1 \sim_1 3 \prec_1 5 \prec_1 2$*, so we have* $|\mathscr{S}^1| = n_1 = 3$ *and* $S_1^1 = \{1, 3\}$*,* $S_2^1 = \{5\}$*,* $S_3^1 = \{2\}$*. Similarly, the acceptable set of customers for product 1 is* $K_1 = \{1, 2, 3, 6, 7, 8\}$*.*

There are 7 different customer budgets: $b^1 = 53, b^2 = 64, \ldots, b^5 = 82, b^6 = 95, b^7 = 120$*. Following the notation,* $\sigma(1) = 7$*, i.e., customer 1 has the* 7^{th} *smallest budget (i.e. the greatest one),* $\sigma(2) = 6$*,* $\sigma(3) = \sigma(4) = 5$*, et cetera. Furthermore, the last row of the table shows a vector of optimal prices along with the optimal value (585). The purchase decision of every customer in this optimal solution is represented by an asterisk next to the entry of the matrix associated to the product he purchases. Notice that, since product 1 is not sold, it does not have an optimal price, but it can implicitly be assumed an optimal price greater than* $b^{|M|} = 120$*.*

The set of indices of budget values that are candidates to be the optimal price of product 4 are $M_4 = \{2, 3, 4, 5\}$*, and in the optimal solution, 4 has price* $b^4 = 79$*. Likewise, the set of indices of candidate prices at which customer 6 may purchase product 4 is* $M_4^6 = \{2, 3\}$*. Indeed, since product 4 is not acceptable for customer 8, its price cannot be set to 53, so customer 6 can only buy product 4 at prices 64 or 65. And finally the set of indices of candidate prices at which customer 6 may purchase a product from* $S_2^6 = \{1, 4\}$ *is* $M_{S_2^6} = \{1, 2, 3\}$*.*

Notice that, even if there are less products than customers and six customers interested in product 1, this product remains unsold in the optimal solution. One could think that, since customer 7 purchases a product with price 53 but he has a budget of 64 and prefers product 1, setting the price 64 for product 1 would lead to a feasible solution with greater objective value. However, the fact that ties are allowed in the RPPT *prevents this solution from being optimal. Indeed, in this case customers 1 and 3 would also purchase product 1 (given that they are indifferent between 1 and the product they are currently purchasing but 1 has a smaller price), and therefore the revenue would be 525 instead of 585.*

3.2 Three-Index Model for the RPPT

In this section, we propose a mixed-integer formulation using two sets of variables. Firstly, we define binary variable v_i^m, $\forall i \in I$, $\forall m \in M_i$, that takes value 1 if the price of product i is equal to the m-th smallest budget b^m. For each $k \in K$, and considering the partition $\mathscr{S}^k$, we define y_n^{km}, $\forall k \in K$, $n \in N^k := \{1, \ldots, n^k\}$, $m \in M_{S_n^k}$, that takes value 1 in a solution provided that customer k purchases a product $i \in S_n^k$ at price b^m.

With these sets of variables, we present a first model called the Three-Index Model (3ITM) for the RPPT:

$$\text{(3ITM)} \quad \max_{\mathbf{v},\mathbf{y}} \quad \sum_{k \in K} \sum_{n \in N^k} \sum_{m \in M_{S_n^k}} b^m y_n^{km} \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{m \in M_i} v_i^m \leq 1 \qquad \forall i \in I, \tag{3.1b}$$

$$\sum_{n \in N^k} \sum_{m \in M_{S_n^k}} y_n^{km} \leq 1 \qquad \forall k \in K, \tag{3.1c}$$

$$y_n^{km} \leq \sum_{i \in S_n^k} v_i^m \qquad \forall k \in K, n \in N^k, m \in M_{S_n^k}, \tag{3.1d}$$

$$\sum_{\substack{m' \in M_i^k: \\ m' \leq m}} v_i^{m'} + \sum_{\substack{m' \in M_{S_n^k}: \\ m' > m}} y_n^{km'} + \sum_{n'=n+1}^{n^k} \sum_{m' \in M_{S_{n'}^k}} y_{n'}^{km'} \leq 1$$
$$\forall k \in K, n \in N^k, i \in S_n^k, m \in M_i^k, \tag{3.1e}$$

$$v_i^m \in \{0, 1\} \quad \forall i \in I, m \in M_i, \tag{3.1f}$$

$$y_n^{km} \in \{0, 1\} \quad \forall k \in K, n \in N^k, m \in M_{S_n^k}. \tag{3.1g}$$

Constraints (3.1b) ensure that each product price is unique. Constraints (3.1c) guarantee that each customer purchases at most one product. Constraints (3.1d) state that if a customer k purchases a product from class S_n^k at price b^m, then there exists $i \in S_n^k$ at price b^m. And constraints (3.1e) are the *preference constraints*, and they ensure that the preferences of the customers are satisfied in any feasible solution. Thus, if the first sum $\sum_{m' \in M_i^k : m' \leq m} v_i^{m'}$ is equal to 1, then k can purchase i at a price smaller than or equal to $b^{m'}$. So the second and third sums of the LHS of (3.1e) are equal to 0, ensuring that k does not purchase either a product from a class $S_{n'}^k$ with $n' > n$, or any product from S_n^k at a higher price $b^{m'}$, $m' > m$. Notice that, even though the inequalities (3.1e) have indices i and n, there is actually one inequality for each $k \in K$, $i \in I^k$, since n is uniquely determined by i because each product i belongs to exactly one class S_n^k (this also applies to the rest of the formulations).

Remark 3.2. *Formulation* (3ITM) *is also valid for the RPP.*

Now we prove that the integrality of the set of y-variables can be relaxed:

Proposition 3.3. *The integrality of variables y_n^{km}, $\forall k \in K$, $\forall n \in N^k$, $\forall m \in M_{S_n^k}$, can be relaxed in formulation* (3ITM). *Indeed, family* (3.1g) *can be replaced with family*

$$y_n^{km} \geq 0 \quad \forall k \in K, n \in N^k, m \in M_{S_n^k}. \tag{3.2}$$

Furthermore, for a given fixed feasible vector $(\bar{v}_i^m) \in \{0, 1\}^{I \times M_i}$ and a fixed customer k, the optimal values of variables y_n^{km} for (3ITM) *with* (3.2) *instead of* (3.1g) *are as follows.*

1. *If $\displaystyle\sum_{i \in I^k} \sum_{m \in M_i^k} \bar{v}_i^m = 0$, then $y_n^{km} = 0 \ \forall n \in N^k, m \in M_{S_n^k}$.*
2. *Otherwise, let $n^* := \min \left\{ n \in N^k : \sum_{i \in S_n^k} \sum_{m \in M_i^k} \bar{v}_i^m \geq 1 \right\}$, $m^* := \min \left\{ m \in M_{S_{n^*}^k} : \sum_{i \in S_{n^*}^k} \bar{v}_i^m \geq 1 \right\}$. Then, $y_{n^*}^{km^*} = 1$, $y_n^{km} = 0$ for $(n, m) \neq (n^*, m^*)$.*

Proof.

1. This is a direct consequence of (3.1d).

2. If $\sum_{i\in I^k}\sum_{m\in M_i^k} \bar{v}_i^m > 0$, let n^*, m^* be as stated, and $i^* \in \{i \in S_{n^*}^k : \bar{v}_i^{m^*} = 1\}$. Then for all $n \in N^k$, $m \in M_{S_n^k}$, $(n,m) \neq (n^*,m^*)$, it holds

   - If $(n,m) < (n^*,m^*)$ (with the lexicographic order), then by the corresponding constraint from (3.1d) we obtain $y_n^{km} \leq \sum_{i\in S_n^k} \bar{v}_i^m = 0$.
   - If $(n,m) > (n^*,m^*)$, then we turn to the constraint from (3.1e) given by $(n,i,m) = (n^*,i^*,m^*)$:

$$\sum_{\substack{m'\in M_{i^*}^k:\\ m'\leq m^*}} \bar{v}_{i^*}^{m'} + \sum_{\substack{m'\in M_{S_{n^*}^k}:\\ m'>m^*}} y_{n^*}^{km'} + \sum_{n'=n^*+1}^{n^k}\sum_{m'\in M_{S_{n'}^k}} y_{n'}^{km'}$$

$$= 1 + \sum_{\substack{m'\in M_{S_{n^*}^k}:\\ m'>m^*}} y_{n^*}^{km'} + \sum_{n'=n^*+1}^{n^k}\sum_{m'\in M_{S_{n'}^k}} y_{n'}^{km'} \leq 1.$$

   We distinguish two cases:

     – If $n = n^*$, then $m > m^*$ and y_n^{km} belongs to the sum $\sum_{m'\in M_{S_{n^*}^k}:m'>m^*} y_{n^*}^{km'}$.

     – If $n > n^*$, then y_n^{km} belongs to $\sum_{n'=n^*+1}^{n^k}\sum_{m'\in M_{S_{n'}^k}} y_{n'}^{km'}$.

   Hence, in both cases the constraint implies $y_n^{km} = 0$.

   We just proved that $y_n^{km} = 0\ \forall (n,m) \neq (n^*,m^*)$. Finally, for $y_{m^*}^{kn^*}$, we have that constraints (3.1b) and (3.1c) reduce to $y_{m^*}^{kn^*} \leq d$ with $d \geq 1$. As for constraints (3.1e), $y_{m^*}^{kn^*}$ may belong to the second or third sum of the LHS for a given k. If $y_{m^*}^{kn^*}$ belongs to the second sum, then $m^* > m$ and hence the sum of v-variables $\sum_{m'\in M_i^k:m'\leq m} v_i^{m'}$ is equal to 0. Otherwise, $y_{m^*}^{kn^*}$ belongs to the third sum, so $n < n^*$ and the way n^* is defined once again implies $\sum_{m'\in M_i^k:m'\leq m} v_i^{m'} = 0$ for such $i \in S_n^k$. Therefore, $y_{m^*}^{kn^*}$ is free, and it will take value 1 in the optimal solution because its coefficient in the objective function is positive.

□

Example 3.4. *Let us describe the variables used to solve the instance given in Table 3.1 with formulation* (3ITM). *First, we define the v-variables associated with each product. For instance, for product 5 we define variables v_5^m for $m \in M_5 = \{1,2,3,5,7\}$. Regarding the y-variables, for customer 2 we have that $n_2 = 2$, and $S_1^2 = \{2,3\}$, $S_2^2 = \{1\}$. For the products in S_1^2, $M_{S_1^2} = \{1,2,3,4,5,6\}$, so we define variables y_1^{2m} for $m \in M_{S_1^2}$. As for S_2^2, we define y_2^{2m} for $m \in M_{S_2^2} = \{1,2,3,5,6\}$ (there are no customers with budget $b^4 = 74$ interested in product 1, so 1 will not have price 74 in an optimal solution). In the optimal solution, customer 2 purchases $2 \in S_1^2$ at price b^4, so $y_1^{24} = 1$.*

Formulation (3ITM) yields very good linear relaxation bounds. As we will address in Section 3.6, the main drawback of this formulation is that it has a large number of variables and constraints, and therefore it is not suitable for instances with a large number of customers or dense matrices of preferences.

3.3 Projecting the customer decision variables on the two-index model

In this section, we discuss how to project out formulation (3ITM) on a formulation of a smaller size, the two-index model (2ITM). The projection results in a set of valid inequalities for (2ITM) for which we develop a separation algorithm.

First, we define the sets of two-index variables of (2ITM). We use variables v_i^m, $\forall i \in I$, $m \in M_i$, that represent, as in (3ITM), the price of a product. Considering once again the partition of I^k into equivalence classes $S_1^k, \dots, S_{n^k}^k$, we define binary variables x_n^k, $\forall k \in K$, $n \in N^k$, as decision variables that take value 1 if customer k purchases some product $i \in S_n^k$, and zero otherwise. And finally, to be able to model the profit of the company, we define continuous variables z_n^k, $\forall k \in K$, $\forall n \in N^k$, that represent the profit associated to a customer k and an equivalence class S_n^k. In a feasible solution, the value of z_n^k is equal to the price of the least expensive product from S_n^k provided that customer k purchases a product from S_n^k, and zero otherwise.

Using these variables, the formulation with two-index variables (2ITM) for the RPPT is:

$$
\begin{aligned}
(2\text{ITM}) \quad \max_{\mathbf{v},\mathbf{x},\mathbf{z}} \quad & \sum_{k \in K} \sum_{n \in N^k} z_n^k && (3.3\text{a}) \\
\text{s.t.} \quad & \sum_{m \in M_i} v_i^m \leq 1 \qquad \forall i \in I, && (3.3\text{b}) \\
& \sum_{n \in N^k} x_n^k \leq 1 \qquad \forall k \in K, && (3.3\text{c}) \\
& x_n^k \leq \sum_{i \in S_n^k} \sum_{m \in M_i^k} v_i^m \qquad \forall k \in K, n \in N^k, && (3.3\text{d}) \\
& \sum_{m \in M_i^k} v_i^m + \sum_{n'=n+1}^{n^k} x_{n'}^k \leq 1 \qquad \forall k \in K, n < n^k, i \in S_n^k, && (3.3\text{e}) \\
& z_n^k \leq b^{\sigma(k)} x_n^k \qquad \forall k \in K, n \in N^k, && (3.3\text{f}) \\
& z_n^k \leq b^{\sigma(k)} + \sum_{m \in M_i^k} \left(b^m - b^{\sigma(k)}\right) v_i^m \qquad \forall k \in K, n \in N^k, i \in S_n^k, && (3.3\text{g}) \\
& v_i^m \in \{0,1\} \qquad \forall i \in I, m \in M_i, && (3.3\text{h}) \\
& x_n^k \in \{0,1\} \qquad \forall k \in K, n \in N^k, && (3.3\text{i}) \\
& z_n^k \geq 0 \qquad \forall k \in K, n \in N^k. && (3.3\text{j})
\end{aligned}
$$

Constraints (3.3b) ensure that each product price is unique. Constraints (3.3c) guarantee that each customer purchases at most one product, i.e., that all customers are unit-demand. Constraints (3.3d) prevent a customer k from purchasing a product $i \in S_n^k$ when he cannot afford it. Constraints (3.3e) are the *preference constraints*, and they guarantee that if a customer k can afford a product i, that is if $\sum_{m\in M_i^k} v_i^m = 1$, then k does not purchase any other product $j \succ_k i$, i.e. $\sum_{n'=n+1}^{n^k} x_n^k = 0$. The sets of constraints (3.3f) and (3.3g) model the profit. Constraints (3.3f) ensure that if customer k does not purchase any product from S_n^k ($x_n^k = 0$), then $z_n^k = 0$ and the corresponding profit is zero. When customer k can afford a product $j \in I^k$, then constraints (3.3g) ensure that the profit associated to k and a class S_n^k is the minimum of the prices of the products in S_n^k. Indeed, when $v_i^{m_0} = 1$ for some $m_0 \leq \sigma(k)$, then $\sum_{m\neq m_0} v_i^m = 0$ and the RHS is equal to $b^{\sigma(k)} - (b^{m_0} - b^{\sigma(k)}) = b^{m_0}$. Since z_n^k is bounded by the price of all the products $i \in S_n^k$, it is actually bounded by the price of the cheapest product from the set. Finally, the objective function (3.3a) represents the profit of the company, that is maximized.

3.3.1 Comparison of formulations (2ITM) and (3ITM)

In this subsection, we compare the bounds given by the linear relaxations of formulations (3ITM) and (2ITM).

Proposition 3.5. *The upper bound given by the linear relaxation of formulation* (3ITM) *is always less than or equal to that of formulation* (2ITM)*.*

Proof. Consider a feasible fractional solution $(\bar{v}, \bar{y})$ of the linear relaxation of (3ITM) that yields an objective value $v(\bar{v}, \bar{y})$. We build a fractional solution $(\hat{v}, \hat{x}, \hat{z})$ of (2ITM) with an objective value $v(\hat{v}, \hat{x}, \hat{z}) \geq v(\bar{v}, \bar{y})$. In this way, if $v(\bar{v}, \bar{y})$ is an optimal solution of the linear relaxation of (3ITM), we obtain $v(\text{2ITM}) \geq v(\hat{v}, \hat{x}, \hat{z}) \geq v(\bar{v}, \bar{y}) = v(\text{3ITM})$, where $v(\text{2ITM})$ (resp. $v(\text{3ITM})$) is the optimal value of the linear relaxation of (2ITM) (resp. (3ITM)).

We define $\hat{v}_i^m := \bar{v}_i^m$, $\hat{x}_n^k := \sum_{m\in M_{S_n^k}} \bar{y}_n^{km}$, $\hat{z}_n^k := \sum_{m\in M_{S_n^k}} b^m \bar{y}_n^{km}$ $\forall k \in K$, $m \in M_i^k$, $n \in N^k$, $i \in S_n^k$.

First, we prove that this solution is feasible for the linear relaxation of (2ITM). Constraints (3.3b) hold because (3.1b) hold. Fixing $\bar{v}$ and $\hat{v}$, the problems are decomposable by customers, so we assume a fixed customer k in the following, and we prove that the associated constraints from sets (3.3c)-(3.3g) hold. As for the corresponding constraint from (3.3c), using the above we have $\sum_{n\in N^k} \hat{x}_n^k = \sum_{n\in N^k} \sum_{m\in M_{S_n^k}} \bar{y}_n^{km}$, and the last sum is less than or equal to 1 because of (3.1c). As for the constraint from (3.3d), it translates to $\sum_{m\in M_{S_n^k}} \bar{y}_n^{km} \leq \sum_{i\in S_n^k} \sum_{m\in M_i^k} \bar{v}_i^m$, which holds because of constraints (3.1d) (summing up on m). Regarding the constraint from (3.3e) $\sum_{m\in M_i^k} \hat{v}_i^k + \sum_{n'=n+1}^{n^k} \hat{x}_{n'}^k \leq 1$, it translates to $\sum_{m\in M_i^k} \bar{v}_i^k + \sum_{n'=n+1}^{n^k} \sum_{m\in M_{S_{n'}^k}} \bar{y}_{n'}^{km} \leq 1$, which is exactly the inequality from set (3.1e) for such k and $m = \sigma(k)$, so it also holds. Constraint $\hat{z}_n^k \leq b^{\sigma(k)} \hat{x}_n^k$ from set (3.3f) holds trivially using the definition of $\hat{x}$ and $\hat{z}$, since $\hat{z}_n^k = \sum_{m\in M_{S_n^k}} b^m \bar{y}_n^{km} \leq b^{\sigma(k)} \sum_{m\in M_{S_n^k}} \bar{y}_n^{km} = b^{\sigma(k)} \hat{x}_n^k$. And finally let us prove the feasibility of the corresponding constraint from (3.3g). To begin with, we know that for a given customer k and product $i \in I^k$, (3.1c) and (3.1e) imply $\sum_{m'\in M_i^k : m' < m} \bar{v}_i^{m'} + \sum_{m'\in M_{S_n^k} : m' \geq m} \bar{y}_n^{km'} \leq 1$ $\forall$ m such that $m - 1 \in M_i^k$. Let us suppose $M_i^k := \{1, 2, \ldots, \sigma(k)\}$. Then, multiplying the previous constraint m such that $m - 1 \in M_i^k$ by $b^m - b^{m-1}$ (where $b^0 = 0$) and adding together all the constraints, we

Table 3.2: Preference matrix of a small instance of the RPPT

	Prod. 1	Prod. 2	Prod. 3	Budgets
Customer 1	1	3	2	2
Customer 2	-	2	1	4
Customer 3	2	1	1	8

obtain:

$$\sum_{m=1}^{\sigma(k)}\sum_{m'=1}^{m-1}\left(b^m-b^{m-1}\right)\bar{v}_i^{m'}+\sum_{m=1}^{\sigma(k)}\sum_{m'=m}^{\sigma(k)}\left(b^m-b^{m-1}\right)\bar{y}_n^{km'}\le\sum_{m=1}^{\sigma(k)}\left(b^m-b^{m-1}\right)=b^{\sigma(k)}. \quad (3.4)$$

The LHS of (3.4) is equal to

$$\begin{aligned}
&\sum_{m=1}^{\sigma(k)}\sum_{m'=1}^{m-1}\left(b^m-b^{m-1}\right)\bar{v}_i^{m'}+\sum_{m=1}^{\sigma(k)}\sum_{m'=m}^{\sigma(k)}\left(b^m-b^{m-1}\right)\bar{y}_n^{km'}\\
&=\sum_{m'=1}^{\sigma(k)-1}\sum_{m=m'}^{\sigma(k)}\left(b^m-b^{m-1}\right)\bar{v}_i^{m'}+\sum_{m'=1}^{\sigma(k)}\sum_{m=1}^{m'-1}\left(b^m-b^{m-1}\right)\bar{y}_n^{km'}\\
&=\sum_{m'=1}^{\sigma(k)-1}\left(b^{\sigma(k)}-b^{m'}\right)\bar{v}_i^{m'}+\sum_{m'=1}^{\sigma(k)}b^{m'}\bar{y}_n^{km'}=\sum_{m'\in M_i^k}\left(b^{\sigma(k)}-b^{m'}\right)\bar{v}_i^{m'}+\hat{z}_n^k.
\end{aligned}$$

All in all, we obtain that constraint $\hat{z}_n^k+\sum_{m'\in M_i^k}\left(b^{\sigma(k)}-b^{m'}\right)\bar{v}_i^{m'}\le b^{\sigma(k)}$ is satisfied, so the corresponding constraint from (3.3g) holds. On the other hand, if $M_i^k\subsetneq\{1,2,\dots,\sigma(k)\}$ it suffices to multiply each constraint associated to $m\in M_i^k$ by $b^m-b^{m'}$, where $m'=\max\left\{m''\in\{0\}\cup M_i^k : m''<m\right\}$ instead, and the same result is obtained applying the previous procedure.

Finally, we need to prove that $v(\hat{v},\hat{x},\hat{z})\ge v(\bar{v},\bar{y})$. But this is straightforward by definition of $\hat{z}$, since $v(\hat{v},\hat{x},\hat{z})=\sum_{k\in K}\sum_{n\in N^k}\hat{z}_n^k=\sum_{k\in K}\sum_{n\in N^k}\sum_{m\in M_{S_n^k}}b^m\bar{y}_n^{km}=v(\bar{v},\bar{y})$. □

Example 3.6. *Let us show through the small example illustrated by Table 3.2 how the linear relaxation bound given by formulation* (3ITM) *can be strictly less than that of* (2ITM)*. An optimal solution of this example is obtained when we assign price $b^1=2$ to product 1 and price $b^2=4$ to product 2 (and product 3 remains unsold). For this price vector, customer 1 purchases 1 and customers 2 and 3 purchase 2, so the optimal value is 10.*

The upper bound given by the linear relaxation of formulation (2ITM) *is 14. The fractional values of v-variables are $v_1^1=1$, $v_2^2=v_3^2=0.5$ (and the rest equal to zero). Likewise, the values of x-variables different from zero are $x_1^1=1$, $x_1^2=x_2^2=0.5$, $x_1^3=0.75$, $x_2^3=0.25$, and the values of z-variables are $z_1^1=2$, $z_1^2=z_2^2=2$, $z_1^3=6$, $z_2^3=2$. However, if we use the same v-values in* (3ITM) *and calculate the y-values by means of constraints* (3.1c)-(3.1e)*, we obtain $y_1^{11}=1$, $y_1^{22}=y_2^{22}=0.5$, $y_1^{32}=1$. This solution yields an objective value* *of 10 in* (3ITM). *In fact, the upper bound given by the linear relaxation of model* (3ITM) *is 12.*

As we will see in the computational experiments of Section 3.6, the upper bounds given by the linear relaxation of formulation (3ITM) *are usually strictly less than those given by formulation* (2ITM). *Regarding the previous instance from Table 3.1, the upper relaxation bounds given by the linear relaxation of formulations* (2ITM) *and* (3ITM) *are, respectively, 640 and 588 (recall that its optimal value is equal to 585).*

3.3.2 Strengthening the two-index model (2ITM)

The linear relaxation of formulation (3ITM) generally yields a smaller upper bound than that of formulation (2ITM). By including variables y_n^{km} in (2ITM) and the related constraints, and then projecting them out, we can derive a set of valid inequalities to strengthen model (2ITM).

We first extend formulation (2ITM) adding y-variables and the corresponding constraints from (3ITM) relating them to the previous variables. By definition, we have $x_n^k = \sum_{m\in M_{S_n^k}} y_n^{km}$ and $z_n^k = \sum_{m\in M_{S_n^k}} b^m y_n^{km}$ for all $k \in K$, $n \in N^k$. Incorporating the constraints from model (3ITM) and using x- and z-variables in place of y-variables when possible leads to:

$$
\begin{aligned}
(2\text{ITM}^+) \quad \max_{\mathbf{v},\mathbf{y},\mathbf{x},\mathbf{z}} \quad & \sum_{k\in K}\sum_{n\in N^k} z_n^k && (3.3\text{a})\\
\text{s.t.} \quad & (3.3\text{b}) - (3.3\text{j}), (3.1\text{d}), (3.1\text{g}),\\
& \sum_{\substack{m'\in M_i^k:\\ m'\le m}} v_i^{m'} + \sum_{\substack{m'\in M_{S_n^k}:\\ m'>m}} y_n^{km'} + \sum_{n'=n+1}^{n^k} x_{n'}^k \le 1\\
& \qquad \forall k\in K, n\in N^k, m\in M_{S_n^k}, i\in S_n^k : m\in M_i^k, && (3.5\text{a})\\
& x_n^k \ge \sum_{m\in M_{S_n^k}} y_n^{km} \quad \forall k\in K, n\in N^k, && (3.5\text{b})\\
& z_n^k \le \sum_{m\in M_{S_n^k}} b^m y_n^{km} \quad \forall k\in K, n\in N^k, && (3.5\text{c})\\
& && (3.5\text{d})
\end{aligned}
$$

Constraints (3.3b)-(3.3j) along with the objective function constitute model (2ITM). Constraints (3.1d) dominate constraints (3.3d). As for (3.5a), they are actually (3.1e) with x replacing y in the third sum. And finally, constraints (3.5b) and (3.5c) relate variables y with x and z. Even though they are inequalities, they are satisfied as equalities by any optimal solution of (2ITM$^+$).

Proposition 3.7. *Consider a fixed customer $k \in K$ and a fixed set of products $S_n^k \in \mathscr{S}^k$. Then the following family of constraints*

$$
z_n^k \le x_n^k \alpha + \sum_{m\in M_{S_n^k}} \sum_{\substack{i\in S_n^k:\\ m\in M_i^k}} \left(1 - \sum_{m'\le m} v_i^{m'} - \sum_{n'=n+1}^{n^k} x_{n'}^k\right)\beta_i^m + \sum_{m\in M_{S_n^k}} \sum_{\substack{i\in S_n^k:\\ m\in M_i^k}} v_i^m \gamma^m \qquad (3.6)
$$

is valid for (2ITM) *if for all* $\alpha \geq 0$, $\beta_i^m \geq 0$ *for* $i \in S_n^k$, $m \in M_i^k$, $\gamma^m \geq 0$ *for* $m \in M_{S_n^k}$, *it holds*

$$\alpha + \sum_{\substack{m' \in M_{S_n^k}: \\ m' < m}} \sum_{\substack{i \in S_n^k: \\ m' \in M_i^k}} \beta_i^{m'} + \gamma^m \geq b^m \qquad \forall m \in M_{S_n^k}. \tag{3.7}$$

Furthermore, the linear relaxation of (2ITM) *plus the set of valid inequalities* (3.6) *is exactly the projection of the linear relaxation of* (3ITM) *on the space of variables* $(\boldsymbol{v}, \boldsymbol{x}, \boldsymbol{z})$.

Proof. For fixed $k \in K$, $n \in N^k$, we are going to project out the y-variables of formulation (2ITM$^+$) to obtain (3.6) and prove the statement. We make use of (3.5b) and (3.5c) from (2ITM$^+$), and we associate dual variables α, β_i^m, γ^m, δ to the corresponding constraints (3.5b), (3.5a), (3.1d), (3.5c), respectively. By Farkas' Lemma (see Chapter 1), we have the following result: given a feasible solution $(\boldsymbol{v}, \boldsymbol{x}, \boldsymbol{z})$ of the linear relaxation of (2ITM), there exists a vector $\mathbf{y}$ satisfying these four sets of inequalities if and only if it holds

$$z_n^k \delta \leq x_n^k \alpha + \sum_{m \in M_{S_n^k}} \sum_{\substack{i \in S_n^k: \\ m \in M_i^k}} \left(1 - \sum_{m' \leq m} v_i^{m'} - \sum_{n'=n+1}^{n^k} x_{n'}^k \right) \beta_i^m + \sum_{m \in M_{S_n^k}} \sum_{\substack{i \in S_n^k: \\ m \in M_i^k}} v_i^m \gamma^m \tag{3.8}$$

$\forall k \in K$, $n \in N^k$, and $\forall (\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}, \delta) \geq \mathbf{0}$ such that

$$\alpha + \sum_{\substack{m' \in M_{S_n^k}: \\ m' < m}} \sum_{\substack{i \in S_n^k: \\ m' \in M_i^k}} \beta_i^{m'} + \gamma^m \geq b^m \delta \qquad \forall m \in M_{S_n^k}. \tag{3.9}$$

If $\delta > 0$, we obtain (3.6) if we normalize by setting $\delta = 1$. Otherwise, the obtained inequality is dominated by (3.3e) and the nonnegativity constraints on variables v_i^m and x_n^k. It is indeed easy to see that for any feasible solution of (2ITM), the RHS of (3.8) is nonnegative. □

Proposition 3.7 provides a family of valid inequalities for (2ITM) of infinite size. Therefore, their inclusion in the model requires the election of a subset of them following a separation procedure. Below, we formally determine the separation problem and show that it is equivalent to a minimum cost flow problem (MCFP).

Let us assume we are given a fractional optimal solution $(\bar{v}_i^m, \bar{x}_n^k, \bar{z}_n^k)$ of the linear relaxation of (2ITM) or a current solution at a given node of the search tree. We solve a separation problem for each customer k and equivalence class $S_n^k \in \mathscr{S}^k$.

First of all, the special structure of conditions (3.7) implies that to minimize of RHS in (3.6), we can set, for each m, at most one β_i^m to a positive value. More precisely, for each m, we define $i_m \in \arg\min_{i \in S_n^k : m \in M_i^k} \left\{ 1 - \sum_{m' \leq m} v_i^{m'} - \sum_{n'=n+1}^{n^k} x_{n'}^k \right\}$ and then set $\beta_i^m = 0 \ \forall i \in S_n^k$, $i \neq i_m$. Hence, the separation problem (SP$_n^k$) can be stated as:

$$(\mathrm{SP}_n^k) \ \min_{\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}} \ \bar{x}_n^k \alpha + \sum_{m \in M_{S_n^k}} \left(1 - \sum_{\substack{m' \in M_{i_m}^k: \\ m' \leq m}} \bar{v}_{i_m}^{m'} - \sum_{n'=n+1}^{n^k} \bar{x}_{n'}^k \right) \beta_{i_m}^m + \sum_{m \in M_{S_n^k}} \sum_{\substack{i \in S_n^k: \\ m \in M_i^k}} \bar{v}_i^m \gamma^m \tag{3.10a}$$

$$\text{s.t.} \quad \alpha + \sum_{m'<m} \beta^{m'}_{i_{m'}} + \gamma^m \geq b^m \qquad \forall m \in M_{S^k_n}, \tag{3.10b}$$

$$\alpha, \beta^m_{i_m}, \gamma^m \geq 0 \qquad \forall m \in M_{S^k_n}.$$

Problem (SP^k_n) is linear and the matrix associated to constraints (3.10b) is binary and possesses the Consecutive Ones Property: the elements equal to 1 in each column appear consecutively. This property permits to solve the problem as a MCFP, see e.g. page 304 in Ahuja et al. (1988). We now describe how to derive this MCFP.

To begin with, we sort the budgets $b^m, m \in M_{S^k_n}$ by increasing order of their values. Then, we transform the constraints in (3.10b) into equalities by introducing slack variables δ^m for each row m in (3.10b). We also add the row $0 \cdot \alpha + 0 \cdot \sum_{m \in M^k_n} \beta^m_{i_m} + 0 \cdot \sum_{m \in M^k_n} \gamma^m + 0 \cdot \sum_{m \in M^k_n} \delta^m = 0$. These modifications lead to an equivalent formulation with the same objective function (3.10a) and the following constraints:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & -1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \boldsymbol{\beta^m_{i_m}} \\ \\ \boldsymbol{\gamma^m} \\ \\ \boldsymbol{\delta^m} \end{bmatrix} = \begin{bmatrix} b^1 \\ b^2 \\ \vdots \\ b^{\sigma(k)} \\ 0 \end{bmatrix},$$

$$\alpha, \beta^m_{i_m}, \gamma^m \geq 0 \qquad \forall m \in M_{S^k_n}.$$

To finish the transformation, we carry out a row operation for each $m = \sigma(k), \sigma(k) - 1, \ldots, 1$ in this order: we subtract the m-th constraint to the $(m+1)$-th one. The equivalent linear formulation (SP-MCFP^k_n) obtained is:

$$\min_{\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}} \quad \alpha \bar{x}^k_n + \sum_{m \in M_{S^k_n}} \beta^m_{i_m} \left(1 - \sum_{\substack{m' \in M^k_{i_m}: \\ m' \leq m}} \bar{v}^{m'}_{i_m} - \sum_{n'=n+1}^{n^k} \bar{x}^k_{n'} \right) + \sum_{m \in M_{S^k_n}} \gamma^m \sum_{\substack{i \in S^k_n: \\ m \in M^k_i}} \bar{v}^m_i \tag{3.11a}$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & -1 & 1 & \cdots & 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & -1 \\ -1 & -1 & \cdots & -1 & 0 & 0 & \cdots & -1 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \boldsymbol{\beta^m_{i_m}} \\ \\ \boldsymbol{\gamma^m} \\ \\ \boldsymbol{\delta^m} \end{bmatrix} = \begin{bmatrix} b^1 \\ b^2 - b^1 \\ \vdots \\ b^{\sigma(k)} - b^{\sigma(k)-1} \\ -b^{\sigma(k)} \end{bmatrix}, \tag{3.11b}$$

$$\alpha, \beta^m_{i_m}, \gamma^m \geq 0 \qquad \forall m \in M_{S^k_n}. \tag{3.11c}$$

The constraint matrix in (3.11b) is the incidence matrix of a graph $G = (N, A)$. Each row corresponds to a node in $N = M_{S^k_n}$ whose supply/demand is given by the corresponding RHS of (3.11b) and each column corresponds to an arc. Hence, the variables represent

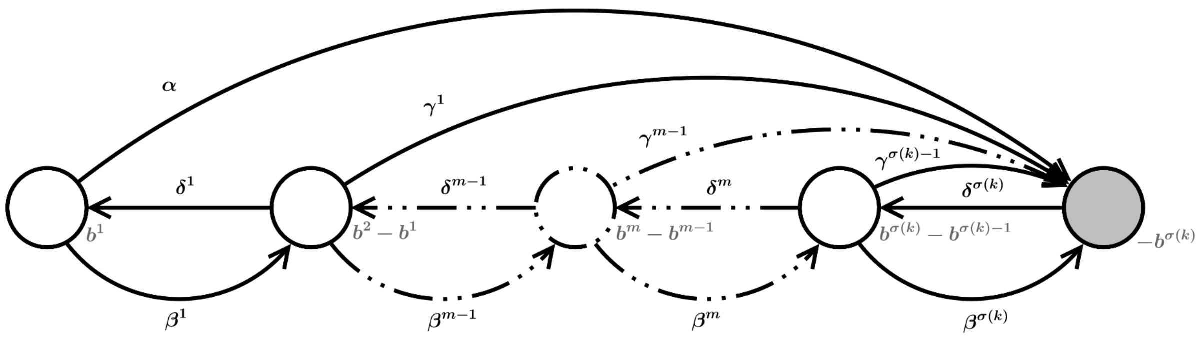

Figure 3.1: MCFP corresponding to (SP-MCFP^k_n). Next to each node we have its supply/demand, and variables $(\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta})$ are associated to the flow of the corresponding arc

uncapacitated flows on the arcs and the objective function consists in minimizing the total cost of the flow. The node corresponding to the last row is the unique sink with demand $b^{\sigma(k)}$ and all other nodes are sources with offer equal to the difference of two consecutive budget values in $M_{S^k_n}$. The MCFP corresponding to problem (SP-MCFP^k_n) is illustrated in Figure 3.1. Given that there is no capacity on the arcs and there is only one sink, the problem can be solved in $M_{S^k_n}$ steps, by computing one shortest path from each source to the sink.

To solve the RPPT with formulation (2ITM) we thus use a branch and cut algorithm that adds violated inequalities from (3.6) at the root node as well as at every node of the branch and bound tree of depth less than 4. Algorithm 2 details the different steps of the separation procedure.

Algorithm 2 Resolution of the separation problems (SP^k_n)

Let $(\bar{v}^m_i, \bar{x}^k_n, \bar{z}^k_n)$ be an optimal fractional solution of the linear relaxation of (2ITM) or a solution found in a node of the search tree of depth less than 4.

For every customer $k \in K$ and integer $n \in N^k$, do

Step 1. Obtain $\bar{i}_m \in \arg\min_{i \in S^k_n} \left\{ 1 - \sum_{m' \leq m} \bar{v}^{m'}_i - \sum_{n'=n+1}^{n^k} \bar{x}^k_{n'} \right\} \; \forall m \in M_{S^k_n}$.

Step 2. Transform the instance of (SP^k_n) into an instance of the MCFP.

Step 3. Compute an optimal flow on the corresponding graph of the instance of MCFP, obtaining $\bar{\alpha}$, $\bar{\beta}^m_{\bar{i}_m}$, $\bar{\gamma}^m$ $\forall m \in M_{S^k_n}$.

Step 4. Incorporate constraint

$$z^k_n \leq \bar{\alpha} x^k_n + \sum_{m \in M_{S^k_n}} \bar{\beta}^m_{\bar{i}_m} \left(1 - \sum_{\substack{m' \in M^k_{\bar{i}_m}: \\ m' \leq m}} v^{m'}_{\bar{i}_m} - \sum_{n'=n+1}^{n^k} x^k_{n'} \right) + \sum_{m \in M_{S^k_n}} \bar{\gamma}^m \sum_{\substack{i \in S^k_n: \\ m \in M^k_i}} v^m_i$$

to (2ITM) provided that it is violated.

3.4 Solution of (3ITM) via Benders Decomposition: the Benders Model

Formulation (3ITM) yields very good linear relaxation bounds but it has a large number of variables and constraints. However, as shown in this section, its structure allows for its resolution by means of a Benders decomposition.

First, we introduce the Benders Model (BDM). To reformulate (3ITM), we need to be able to relax the integrality on the set of y-variables. However, this result is shown in Proposition 3.3. We address the Benders reformulation of (3ITM) and relate it to the Benders Model in the following subsections.

We define continuous variables z^k, $\forall k \in K$, that represent the profit associated to customer k. With this set of variables and the set of v-variables used for (3ITM) and (2ITM), we present the Benders Model (BDM) for the RPPT:

$$
\begin{aligned}
\text{(BDM)} \quad \max_{\mathbf{v},\mathbf{z}} \quad & \sum_{k \in K} z^k && \text{(3.12a)}\\
\text{s.t.} \quad & \sum_{m \in M_i} v_i^m \leq 1 \quad \forall i \in I, && \text{(3.12b)}\\
& z^k \leq \sum_{i \in I^k} \sum_{m \in M_i^k} b^m v_i^m \quad \forall k \in K, && \text{(3.12c)}\\
& z^k \leq b^{\sigma(k)} + \sum_{m \in M_i^k} \left(b^m - b^{\sigma(k)}\right) v_i^m + \sum_{\substack{j \in I^k:\\ j \prec_k i}} \sum_{m \in M_j^k} b^m v_j^m \\
& \qquad \forall k \in K, i \in I^k, && \text{(3.12d)}\\
& v_i^m \in \{0,1\} \quad \forall i \in I, m \in M_i, && \text{(3.12e)}\\
& z^k \geq 0 \quad \forall k \in K. && \text{(3.12f)}
\end{aligned}
$$

Proposition 3.8. *Model* (BDM) *is valid for the* RPPT*.*

Proof. Constraints (3.12c) guarantee that if customer k cannot afford any product, then $z^k = 0$. When k can afford several products, the RHS of (3.12c) is an upper bound on the value of z^k.

Constraints (3.12d) model the preferences and ensure that k purchases his most preferred product (at the cheapest price in case of ties). Indeed, given an integer feasible solution $(\bar{v}, \bar{z})$, let $n^* := \min\{n \in N^k : \sum_{i \in S_n^k} \sum_{m \in M_i^k} \bar{v}_i^k \geq 1\}$, and $i^* \in \arg\min_{i \in S_{n^*}^k} \{\sum_{m \in M_i^k} b^m \bar{v}_i^m\}$. Clearly, $S_{n^*}^k$ is the first class (according to the ranking) from which k is able to afford a product, whereas i^* is one of the cheapest products from $S_{n^*}^k$. So assuming $\bar{v}_{i^*}^{m^*} = 1$, we need to prove that it holds $z^k = \sum_{m \in M_{i^*}^k} b^m \bar{v}_{i^*}^m = b^{m^*}$. Since we are maximizing the objective, it suffices to prove that all the RHSs of (3.12c) and (3.12d) for such k are all greater than or equal to b^{m^*}, and that at least one is equal to b^{m^*}. We have one constraint per product $i \in I^k$, so to begin with we distinguish two cases:

- $i \preceq i^*$. In this case, the last sum of the corresponding constraint from (3.12d) $\sum_{j \in I^k : j \prec_k i} \sum_{m \in M_j^k} b^m v_j^m$ equals 0. We have three subcases to consider:

- $i \prec_k i^*$. Then k cannot afford i or any $j \prec_k i$, so the RHS of (3.12d) is equal to $b^{\sigma(k)}$, an upper bound on the profit from k.
- $i = i^*$. In this case, the RHS of (3.12d) is equal to b^{m^*}:

$$b^{\sigma(k)} + \sum_{m \in M^k_{i^*}} \left(b^m - b^{\sigma(k)}\right) \bar{v}^m_{i^*} = b^{\sigma(k)} + \left(b^{m^*} - b^{\sigma(k)}\right) = b^{m^*}.$$

- $i \sim_k i^*$, $i \neq i^*$. In this case, by definition of i^* we know that $\bar{v}^{\hat{m}}_i = 1$ for some $\hat{m} \geq m^*$. If $\hat{m} > \sigma(k)$, then the RHS of (3.12d) is equal to $b^{\sigma(k)}$. Otherwise, we have $\sum_{m \in M^k_i} b^m \bar{v}^m_i = b^{\hat{m}} \geq b^{m^*}$ and it holds

$$b^{\sigma(k)} + \sum_{m \in M^k_i} \left(b^m - b^{\sigma(k)}\right) \bar{v}^m_i = b^{\sigma(k)} + \left(b^{\hat{m}} - b^{\sigma(k)}\right) = b^{\hat{m}}.$$

- $i \succ_k i^*$. Then it holds

$$b^{\sigma(k)} + \sum_{m \in M^k_i} \left(b^m - b^{\sigma(k)}\right) \bar{v}^m_i + \sum_{\substack{j \in I^k: \\ j \prec_k i}} \sum_{m \in M^k_j} b^m \bar{v}^m_j \geq \sum_{\substack{j \in I^k: \\ j \prec_k i}} \sum_{m \in M^k_j} b^m \bar{v}^m_j \geq b^{m^*},$$

where the last inequality holds because $v^{m^*}_{i^*} = 1$ belongs to the previous sum.

□

3.4.1 Benders Reformulation

We can proceed with a Benders reformulation of (3ITM):

$$(\mathrm{BDM_{MAS}}) \quad \max_{\mathbf{v},\mathbf{z}} \quad \sum_{k \in K} z^k \tag{3.13a}$$

$$\text{s.t.} \quad \sum_{m \in M_i} v^m_i \leq 1 \quad \forall i \in I, \tag{3.13b}$$

$$z^k \leq P^k(v), \quad \forall k \in K \tag{3.13c}$$

$$v^m_i \in \{0,1\} \quad \forall i \in I, m \in M_i, \tag{3.13d}$$

$$z^k \geq 0 \quad \forall k \in K, \tag{3.13e}$$

where $\forall k \in K$, $P^k(v)$ is defined as the optimal value of

$$(\mathrm{BDM_{SUB^k}}) \quad \max_{\mathbf{y}} \quad \sum_{n \in N^k} \sum_{m \in M_{S^k_n}} b^m y^m_n \tag{3.14a}$$

$$\text{s.t.} \quad \sum_{n \in N^k} \sum_{m \in M_{S^k_n}} y^m_n \leq 1, \tag{3.14b}$$

$$y^m_n \leq \sum_{i \in S^k_n} v^m_i \qquad \forall n \in N^k, m \in M_{S^k_n}, \tag{3.14c}$$

$$\sum_{\substack{m' \in M_{S^k_n}: \\ m' \geq m}} y^{m'}_n + \sum_{n'=n+1}^{n^k} \sum_{m' \in M_{S^k_{n'}}} y^{m'}_{n'} \leq 1 - \sum_{\substack{m' \in M^k_i: \\ m' < m}} v^{m'}_i$$

$$\forall n \in N^k, i \in S_n^k, m \in M_i^k, \tag{3.14d}$$

$$y_n^m \geq 0 \quad \forall n \in N^k, m \in M_{S_n^k}. \tag{3.14e}$$

In ($\mathrm{BDM}_{\mathrm{SUB}^k}$), we drop the upper index k of the y-variables for the sake of notation. Constraints (3.13b) guarantee that every product price is unique. This guarantees the feasibility in problem ($\mathrm{BDM}_{\mathrm{SUB}^k}$) for a given integer solution (v_i^m) of (3ITM), since the RHS of constraints (3.14b)-(3.14d) is always nonnegative. Hence, these constraints act as feasibility cuts ensuring that any pricing v in the master problem ($\mathrm{BDM}_{\mathrm{MAS}}$) admits a feasible allocation y in the original model (3ITM). As a consequence, no feasibility cuts are added in our Benders decomposition (see Section 1.1.3 from Chapter 2).

Furthermore, constraint (3.14b) ensures that ($\mathrm{BDM}_{\mathrm{SUB}^k}$) is bounded. Therefore, by linear optimization strong duality (see Section 1.1.1), the optimal value of problem ($\mathrm{BDM}_{\mathrm{SUB}^k}$) is equal to the optimal value of its dual problem, ($\mathrm{BDM}_{\mathrm{SUBD}^k}$). Associating variables $\alpha, \beta_i^m, \gamma_n^m$ to the corresponding constraint from sets (3.14b), (3.14d), (3.14c), respectively, ($\mathrm{BDM}_{\mathrm{SUBD}^k}$) can be stated as

$$(\mathrm{BDM}_{\mathrm{SUBD}^k}) \quad \min_{\alpha,\boldsymbol{\beta},\boldsymbol{\gamma}} \quad \alpha + \sum_{n\in N^k} \sum_{i \in S_n^k} \sum_{m \in M_i^k} \left(1 - \sum_{m' \in M_i^k : m' < m} v_i^{m'}\right) \beta_i^m + \sum_{n\in N^k} \sum_{i \in S_n^k} \sum_{m \in M_i^k} v_i^m \gamma_n^m \tag{3.15a}$$

$$\text{s.t.} \quad \alpha + \sum_{n'=1}^{n-1} \sum_{i \in S_{n'}^k} \sum_{m' \in M_i^k} \beta_i^{m'} + \sum_{i \in S_n^k} \sum_{\substack{m' \in M_i^k : \\ m' \leq m}} \beta_i^{m'} + \gamma_n^m \geq b^m \quad \forall n \in N^k, m \in M_{S_n^k}, \tag{3.15b}$$

$$\alpha, \beta_i^m, \gamma_n^m \geq 0 \qquad \forall n \in N^k, m \in M_{S_n^k}, i \in S_n^k. \tag{3.15c}$$

Now, we can rewrite problem ($\mathrm{BDM}_{\mathrm{MAS}}$) making use of subproblems ($\mathrm{BDM}_{\mathrm{SUBD}^k}$). Thus, defining D^k for each $k \in K$ as the set of feasible solutions ($\alpha^k, \beta_i^{km}, \gamma_n^{km}$) for the dual subproblem ($\mathrm{BDM}_{\mathrm{SUBD}^k}$), we have:

$$(\mathrm{BDM}_{\mathrm{MAS}}) \quad \max_{\mathbf{v},\mathbf{z}} \quad \sum_{k \in K} z^k \tag{3.16a}$$

$$\text{s.t.} \quad \sum_{m \in M_i} v_i^m \leq 1 \quad \forall i \in I, \tag{3.16b}$$

$$z^k \leq \alpha^k + \sum_{n\in N^k} \sum_{i \in S_n^k} \sum_{m \in M_i^k} \left(1 - \sum_{m' \in M_i^k : m' < m} v_i^{m'}\right) \beta_i^{km} + \sum_{n\in N^k} \sum_{i \in S_n^k} \sum_{m \in M_i^k} v_i^m \gamma_n^{km}, \quad \forall k \in K, (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \in D^k, \tag{3.16c}$$

$$v_i^m \in \{0, 1\} \quad \forall i \in I, m \in M_i, \tag{3.16d}$$

$$z^k \geq 0 \quad \forall k \in K. \tag{3.16e}$$

Finally, we state that model ($\mathrm{BDM_{MAS}}$) obtained by means of a Benders reformulation is in fact a reinforcement of the previous Benders Model (BDM):

Proposition 3.9. *The sets of constraints* (3.12c) *and* (3.12d) *are included in* (3.16c).

Proof. We drop the k index from the variables for the sake of notation. Constraints (3.12c) are obtained, for a fixed customer k, when fixing $\alpha := 0$, $\beta_i^m := 0\ \forall i \in I^k$, $m \in M_i^k$, $\gamma_n^m := b^m\ \forall n \in N^k$, $m \in M_{S_n^k}$. The described $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ belongs to D^k because constraints (3.15b) are trivially satisfied, since for each $m \in M_{S_n^k}$ it holds $\alpha + \sum_{n'=1}^{n-1} \sum_{i \in S_{n'}^k} \sum_{m' \in M_i^k} \beta_i^{m'} + \sum_{i \in S_n^k} \sum_{\substack{m' \in M_i^k:\\ m' < m}} \beta_i^{m'} + \gamma_n^m \geq \gamma_n^m = b^m$.

As for constraints (3.12d), consider for fixed $k \in K$, $n^* \in N^k$ and $i^* \in S_{n^*}^k \subset I^k$, and assume $M_{i^*}^k = \{1, \ldots, \sigma(k)\}$. Let us set the values $\alpha := b^1$, $\beta_{i^*}^m := b^{m+1} - b^m$ for $m \in M_{i^*}^k : m < \sigma(k)$, $\beta_{i^*}^{\sigma(k)} := 0$, $\beta_i^m := 0\ \forall i \neq i^*$, $m \in M_i^k$, $\gamma_n^m := b^m\ \forall n < n^*$, $m \in M_{S_n^k}$, $\gamma_n^m := 0$ for $n \geq n^*$, $m \in M_{S_n^k}$. Then it follows $\sum_{m \in M_{i^*}^k} \beta_{i^*}^m = \sum_{m=1}^{\sigma(k)-1} (b^{m+1} - b^m) = b^{\sigma(k)} - b^1$ and $\sum_{m \in M_{i^*}^k : m \geq m'} \beta_{i^*}^m = b^{\sigma(k)} - b^{m'}$ for $m' \in M_{i^*}^k$. Therefore, we have that the RHS of the corresponding constraint from (3.16c) is

$$\alpha + \sum_{n \in N^k} \sum_{i \in S_n^k} \sum_{m \in M_i^k} \left(1 - \sum_{\substack{m' \in M_i^k:\\ m' \leq m}} v_i^{m'} \right) \beta_i^m + \sum_{n \in N^k} \sum_{i \in S_n^k} \sum_{m \in M_i^k} v_i^m \gamma_n^m$$

$$= b^1 + \sum_{m \in M_{i^*}^k} \left(1 - \sum_{\substack{m' \in M_{i^*}^k:\\ m' \leq m}} v_{i^*}^{m'} \right) \beta_{i^*}^m + \sum_{n < n^*} \sum_{i \in S_n^k} \sum_{m \in M_i^k} v_i^m b^m$$

$$= b^1 + \sum_{m \in M_{i^*}^k} \beta_{i^*}^m - \sum_{m' \in M_{i^*}^k} \left(\sum_{\substack{m \in M_{i^*}^k:\\ m \geq m'}} \beta_{i^*}^m \right) v_{i^*}^{m'} + \sum_{\substack{i \in I^k:\\ i \prec i^*}} \sum_{m \in M_i^k} v_i^m b^m$$

$$= b^1 + \left(b^{\sigma(k)} - b^1 \right) - \sum_{m' \in M_{i^*}^k} \left(b^{\sigma(k)} - b^{m'} \right) v_{i^*}^{m'} + \sum_{\substack{i \in I^k:\\ i \prec i^*}} \sum_{m \in M_i^k} v_i^m b^m,$$

which is equal to the RHS of (3.12d) for customer k and product $i^* \in S_{n^*}^k$.

To check whether $(\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma})$ belongs to D^k, and knowing that the vectors are nonnegative by definition, it is left to prove that (3.15b) hold $\forall n \in N^k$, $m \in M_{S_n^k}$, $i \in S_n^k$. To do so, we study three cases depending on $n \in N^k$:

- $n < n^*$. Then for given $m \in M_{S_n^k}$, we have the LHS of (3.15b) equal to

$$\alpha + \sum_{n'=1}^{n-1} \sum_{i \in S_{n'}^k} \sum_{m' \in M_i^k} \beta_i^{m'} + \sum_{i \in S_n^k} \sum_{\substack{m' \in M_i^k:\\ m' < m}} \beta_i^{m'} + \gamma_n^m \geq \gamma_n^m = b^m.$$

- $n = n^*$. Then it holds for $m \in M_{S_n^k}$:

$$\alpha + \sum_{n'=1}^{n^*-1} \sum_{i \in S_{n'}^k} \sum_{m' \in M_i^k} \beta_i^{m'} + \sum_{i \in S_{n^*}^k} \sum_{\substack{m' \in M_i^k: \\ m' < m}} \beta_i^{m'} + \gamma_{n^*}^m = \alpha + \sum_{\substack{m' \in M_{i^*}^k: \\ m' < m}} \beta_{i^*}^{m'} = b^1 + \big(b^m - b^1\big) = b^m.$$

- $n > n^*$. Then for given $m \in M_{S_n^k}$, we have the LHS of (3.15b) equal to

$$\alpha + \sum_{n'=1}^{n^*-1} \sum_{i \in S_{n'}^k} \sum_{m' \in M_i^k} \beta_i^{m'} + \sum_{i \in S_{n^*}^k} \sum_{\substack{m' \in M_i^k: \\ m' < m}} \beta_i^{m'} + \gamma_{n^*}^m = \alpha + \sum_{m' \in M_{i^*}^k} \beta_{i^*}^{m'} = b^{\sigma(k)}.$$

In the three cases, the LHS of (3.15b) is greater than or equal to b^m, so the given $(\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma})$ satisfies (3.15b) and thus it belongs to D^k.

If $M_{i^*}^k \subsetneq \{1, 2, \ldots, \sigma(k)\}$, the proof follows analogously applying the previous procedure to the same α and $\boldsymbol{\gamma}$, but defining $\beta_{i^*}^m := b^{m'} - b^m$, where $m' = \min\{m'' \in M_{i^*}^k : m'' > m\}$, for $m \in M_i^k : m < \sigma(k)$, $\beta_{i^*}^{\sigma(k)} := 0$, $\beta_i^m := 0 \ \forall i \neq i^*$, $m \in M_i^k$. □

3.4.2 Resolution Approach

The classical Benders resolution approach begins by solving to optimality the master problem ($\mathrm{BDM_{MAS}}$) without constraints (3.16c). Then, a subset of constraints from (3.16c) is obtained by solving problems ($\mathrm{BDM_{SUBD^k}}$) for all $k \in K$, and the violated constraints are added to the master problem, which is again solved to optimality. This process is done iteratively until none of the constraints from (3.16c) is violated, and thus the solution is optimal for ($\mathrm{BDM_{MAS}}$). The drawback of this method is that the integer model ($\mathrm{BDM_{MAS}}$) is solved many times, which can take a considerable amount of time.

In the lazy approach, however, the resolution starts by solving the linear relaxation of ($\mathrm{BDM_{MAS}}$) without the set (3.16c), obtaining a fractional solution and an upper bound on the optimal value. In order to decrease this bound, the subproblems ($\mathrm{BDM_{SUBD^k}}$) are solved for each customer using the fractional solution of the master problem, and a set of constraints is added to the problem. Optimality cuts are added at this phase until the bound is no longer improved. The second step of the resolution is to solve the integer problem with the usual branch-and-bound algorithm. In this phase, constraints are added in the so-called lazy fashion, i.e. only checking for them when the resolution of a node in the search tree leads to an integer solution. In such case, if a constraint is violated, the cut is pulled into the active node and the solution is discarded. Otherwise, the solution is feasible for ($\mathrm{BDM_{MAS}}$). At this step, constraints from (3.16c) may also be added at a current fractional node of the branching tree. The interested reader may find the advantages of this method thoroughly explained in Naoum-Sawaya and Elhedhli (2013).

In this work, we solve the Benders Model (BDM) instead of ($\mathrm{BDM_{MAS}}$). The advantage is that, since (BDM) gives feasible solutions for the RPPT, we no longer need to solve ($\mathrm{BDM_{SUBD^k}}$) in order to check the validity of an integer solution of the master. Nevertheless, we can still add valid inequalities from (3.16c) to cut off fractional solutions of (BDM), thus strengthening the model.

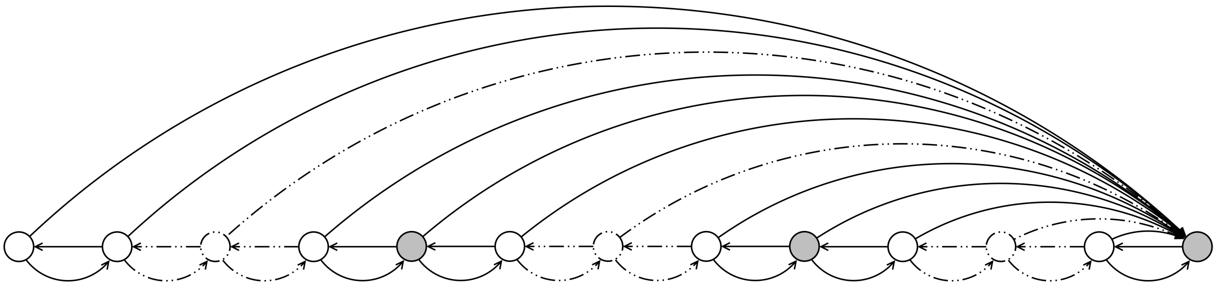

Figure 3.2: Structure of the MCFP graph corresponding to ($\mathrm{BDM}_{\mathrm{SUBD}^k}$). Source nodes appear in white and sink nodes appear in grey

Resolution of the dual subproblem ($\mathrm{BDM}_{\mathrm{SUBD}^k}$)

Solving problem ($\mathrm{BDM}_{\mathrm{SUBD}^k}$) for fractional solutions of (BDM) is interesting because it allows for the incorporation of valid inequalities in the linear relaxation phase, thus helping to decrease the upper bound before solving the integer phase. An analogous procedure to that of the resolution of the separation problem (SP_n^k) in Subsection 3.3.2 can be applied to ($\mathrm{BDM}_{\mathrm{SUBD}^k}$). In this case, it suffices to consider the lexicographical order in the rows of matrix (3.15b) (that is, $(n,m) < (n',m')$ if $n < n'$ or $n = n'$ and $m < m'$) in order to state that it also satisfies the Consecutive Ones Property. Thus, ($\mathrm{BDM}_{\mathrm{SUBD}^k}$) can be transformed into a MCFP and solved by means of an efficient implementation of an existing algorithm.

For our implementation, we have selected the Successive Shortest Path (SSP) Algorithm to obtain the solution of the MCFP. In each iteration, this algorithm selects a shortest path between a supply and a demand node and increases the flow along the path (it also modifies the reduced costs of the arcs used to compute the shortest path and the residual network in each iteration). Since our graph has $\sum_{n\in N^k} |M_{S_n^k}| + 1$ nodes, solving the problem for a given customer k can take at most $\sum_{n\in N^k} |M_{S_n^k}|$ iterations. When the preference matrix is dense, this amounts to $|N^k|\sigma(k)$.

Leveraging the special structure of our MCFP, we have reduced the number of iterations in which a shortest path is computed. The structure of the graph associated to our MCFP is depicted in Figure 3.2. As in the graph from Figure 3.1, the white nodes represent sources, the grey ones represent sinks and sending flow through δ-arcs (the arcs from a node to the previous one) has cost equal to zero. Hence, we need not compute the shortest path between a node with excess supply (n,m) and a node with unfulfilled demand (n',m') whenever $(n,m) > (n',m')$. In the first phase of the algorithm, we select a source (n,m) and a sink (n',m') with $(n,m) > (n',m')$, and then apply the SSP algorithm without computing the shortest path. Then, when for all supply node (n,m) and demand node (n',m') it holds $(n,m) < (n',m')$, we continue with the second phase, where we apply the SSP algorithm in the standard way. This preprocessing of the MCFP reduces the number of iterations in which an algorithm to obtain a shortest path is executed to at most $\sigma(k)$ iterations. Thus, the amount of computational time saved during the first phase is significant.

Finally, note that the transformation of the subproblems into a MCFP can also be used to solve the subproblems of the Benders decomposition proposed by Bertsimas and Mišić (2019) for the resolution of the PLD problem. Indeed, the Consecutive Ones Property holds in this case as well.

In-out stabilization method and overall resolution approach

In this subsection, we present our resolution strategy to solve model (BDM) as well as an in-out stabilization method implemented to speed up the linear relaxation phase of the resolution.

The procedure is divided in two phases:

1. **Linear relaxation phase.** The linear relaxation of (BDM) is solved, obtaining a fractional solution and an upper bound on the optimal value. In order to decrease this bound, the corresponding MCFP of subproblems ($\text{BDM}_{\text{SUBD}^k}$) are solved for each customer and for the fixed fractional solution of the master, and a set of valid inequalities from (3.16c) is derived and added to the formulation. Valid inequalities are added at this phase until the upper bound is no longer improved.
2. **Integer phase.** The integer problem with the subset of constraints derived in the previous phase is solved to optimality by means of a branch-and-cut. Due to the fact that it is very time consuming, no more valid inequalities from (3.16c) are added in this phase.

As we have proved, the SSP algorithm used to solve the transformation of subproblems ($\text{BDM}_{\text{SUBD}^k}$) into a MCFP constitutes an exact algorithm of separation. In this sense, it finds at least one violated constraint for any solution of (BDM) which is infeasible for (BDM_{MAS}). On the other hand, when the problem size is large, computing these inequalities is time consuming, and frequently the upper bound decreases very slowly and many cuts are generated in the process. In order to speed up this cutting phase, we implemented an in-out stabilization method with the aim of generating less cuts of better quality. The steps of the cutting plane in-out algorithm are detailed in Ben-Ameur and Neto (2007) and Bonami et al. (2020).

Let D represent the domain given by all the constraints of problem (BDM_{MAS}), and $P \supseteq D$ the domain given by the constraints from (BDM). Then the in-out stabilization method is based on the election of good separation points. Specifically, at each loop iteration of the linear relaxation phase three points are considered: a point $(v_{out}, z_{out}) \in P \setminus D$ given by the optimal solution of the linear relaxation of the current reduced master problem (BDM), a feasible interior point $(v_{in}, z_{in}) \in D$, and a separation point (v_{sep}, z_{sep}), which is a convex combination of the previous two: $(v_{sep}, z_{sep}) := \lambda(v_{out}, z_{out}) + (1-\lambda)(v_{in}, z_{in})$ with $\lambda \in (0, 1]$. At each iteration, two possibilities can occur. If $(v_{sep}, z_{sep}) \notin D$, then we use it instead of (v_{out}, z_{out}) as a separation point to solve the dual subproblem ($\text{BDM}_{\text{SUBD}^k}$), since the inequalities provided by this point are expected to be more efficient. We finish the iteration by solving the new optimization problem and obtaining a new point (v_{out}, z_{out}). Otherwise, $(v_{sep}, z_{sep}) \in D$, and in this case solving the dual subproblem does not provide new violated cuts. Therefore, in this iteration no constraints are added but (v_{in}, z_{in}) is replaced with (v_{sep}, z_{sep}), which is a feasible point with greater objective value. As we can see, at each iteration either (v_{in}, z_{in}) or (v_{out}, z_{out}) are updated, until convergence is obtained because the relative difference between the two points is lower than a fixed tolerance ϵ. Although λ is a scalar that can change in every iteration, preliminary testing led us to set $\lambda = 0.99$ for all iterations.

As for the interior point (v_{in}, z_{in}), it is frequently obtained using the barrier algorithm with crossover. In our case, an interior point $(v_{in}, z_{in}) \in D$ can be very easily derived by exploiting the particular structure of the problem. To do so, it suffices to build a non-degenerated convex combination of $|I \times M^I| + |K| + 1$ points of the polytope and then compute the centroid. Point (i,m) of the first feasible set of $|I \times M^I|$ points was created taking $v_i^m = 1$, $v_{i'}^{m'} = 0$ for $(i', m') \neq (i,m)$, $\boldsymbol{z} = \boldsymbol{0}$. Point k of the next $|K|$ points is $z^k = b^{\sigma(k)}$, $z^{k'} = 0$ for $k' \neq k$, $v_i^m = 1$ for $i = \min\{i \in S_1^k\}$, $m = \sigma(k)$, $v_{i'}^{m'} = 0$ for $(i', m') \neq (i,m)$. Finally, we used $(\boldsymbol{v}, \boldsymbol{z}) = \boldsymbol{0}$.

3.5 Preprocessing

In this section, we present a preprocessing procedure with the aim of reducing the size of the problem by fixing variables to zero. Note that, even though the results are stated for models (2ITM) and (3ITM), they also apply to subproblems ($\text{BDM}_{\text{SUBD}^k}$) during the resolution of model (BDM). This preprocessing is based on the one described in Chapter 2. We define a recursive function $u' : K \to \mathscr{S}^K$ that assigns the index n of an equivalence class $S_n^k \in \mathscr{S}^k$ to each customer $k \in K$. Function u' is defined as follows, for the set of customers ordered according to their budgets in decreasing order:

1. If $\sigma(k) = |M|$, then $u'(k) := 1$.
2. If $\sigma(k) < |M|$ and it holds $I^k \nsubseteq \left(\cup_{\substack{k' \in K: \\ \sigma(k') > \sigma(k)}} S_{u'(k')}^{k'} \right)$, then
$$u'(k) := \min \left\{ n \in N^k : S_n^k \nsubseteq \left(\cup_{\substack{k' \in K: \\ \sigma(k') > \sigma(k)}} S_{u'(k')}^{k'} \right) \right\}.$$
3. If $\sigma(k) < |M|$ and it holds $I^k \subseteq \left(\cup_{\substack{k' \in K: \\ \sigma(k') > \sigma(k)}} S_{u'(k')}^{k'} \right)$, then $u'(k) := n^k$.

Function u' assigns to the richest customers the equivalence class that contains their favorite products. To the rest of the customers it assigns, among the equivalence classes with products that are not in the equivalence classes associated to richer customers, the equivalence class with the highest ranked ones. And if for every product $i \in I^k$ it holds that i is already in another customer's class $S_{u'(k')}^{k'}$, then u' assigns to k the index of the class with his least favorite products. Function u' allows for the same partition of the set of customers than u did in Chapter 2. Thus, let C_r, $r = \{1,2,3\}$, be such that $k \in C_r$ if and only if $u'(k)$ has been defined for k making use of item r of the definition of u'.

For any instance of the RPP (without ties) it follows $|S_n^k| = 1 \; \forall k, n$, and therefore $u'(k) = n \Rightarrow u(k) = i$ for $i \in S_n^k$, for the function u described in Chapter 2. However, the definition of u does not apply to the RPPT, since there may be more than one product on each class S_n^k. This is the reason for defining a function that assigns equivalence classes to customers instead of products. Now the use of this new function allows us to fix variables to zero:

Proposition 3.10. *For* (2ITM) *(resp.* (3ITM)*), there exists an optimal solution* $(\bar{v}_i^m, \bar{x}_n^k, \bar{z}_n^k)$ *(resp.* $(\bar{v}_i^m, \bar{y}_n^{km})$*) such that* $\bar{x}_n^k = 0$ *(resp.* $\bar{y}_n^{km} = 0$*) for all* $k \in K$*,* $n > u'(k)$*,* $m \in M_{S_n^k}$*.*

Proof. We shall prove the statement for model (2ITM), since the proof for model (3ITM) is analogous. Thus, suppose we have an optimal solution $(\hat{v}_i^m, \hat{x}_n^k, \hat{z}_n^k)$ not satisfying the

Table 3.3: Preprocessing of the instance shown in Table 3.1

	Prod. 1	Prod. 2	Prod. 3	Prod. 4	Prod. 5	Budgets
Customer 1	1	3	1*	-	2	120
Customer 2	2	1*	1	-	-	95
Customer 3	1	2	4	1*	3	82
Customer 4	-	3	1	3	2*	82
Customer 5	-	1	3	2*	-	79
Customer 6	2	-	1	2	1*	65
Customer 7	3	2	5	1	4*	64
Customer 8	1	4	2	-	3*	53
Optimal prices	-	95	120	79	53	585

statement conditions. Our aim is to build another one which does satisfy them. We will proceed by induction on k.

To begin with, it is clear that the statement holds for all customers k with budget $b^{|M|}$. Indeed, since these customers can afford any product, they always get one of their favorite ones, so one in the set S_1^k, and $\hat{x}_n^k = 0$ for $n > 1 = u'(k)$. Now, let $k_0 \in K$ be such that the statement holds $\forall k \in K$ with $\sigma(k) > \sigma(k_0)$ but $\hat{x}_n^{k_0} = 1$ for some $n > u'(k_0)$. Then it is clear that $k_0 \notin C_3$. Besides, from the definition of u' we know there is a product $i_0 \in S_{u'(k_0)}^{k_0} \setminus \cup_{\substack{k \in K: \\ \sigma(k) > \sigma(k_0)}} S_{u'(k)}^k$, and we also know that i_0 remains unsold in this solution.

Hence, consider the vector of prices $\bar{v}_i^m$ obtained by modifying the price of i_0: $\bar{v}_i^m = \hat{v}_i^m \ \forall i \neq i_0, m \in M_i^k$, $\bar{v}_{i_0}^{\sigma(k_0)} = 1$, $\bar{v}_{i_0}^m = 0 \ \forall m \neq \sigma(k_0)$. Given this vector of prices, customers k with $\sigma(k) < \sigma(k_0)$ can afford the same products than in solution $(\hat{v}_i^m, \hat{x}_n^k, \hat{z}_n^k)$, so they make the same purchase. Customers k with $\sigma(k) > \sigma(k_0)$ were already purchasing in the previous solution a product that they liked better than i_0. And customers k with $\sigma(k) = \sigma(k_0)$ might purchase product i_0 in the new solution, but in this case, since they pay their whole budget, the objective value does not decrease with respect to the previous solution. Therefore, $(\hat{v}_i^m, \hat{x}_n^k, \hat{z}_n^k)$ is an optimal solution that meets the statement requirements for customer k_0. Applying the procedure iteratively, we can obtain an optimal solution satisfying the statement. □

Example 3.11. *To illustrate the previous proposition,* Table 3.3 *shows the instance proposed in* Table 3.1 *already preprocessed. For every customer k, the preferences associated to products in $S_{u'(k)}^k$ are circled. Preferences appear in grey for those products in S_n^k such that x_n^k has been fixed to zero. For customer 1, by definition we have $n_1 = 3$, $S_1^1 = \{1, 3\}$, $S_2^1 = \{5\}$ and $S_3^1 = \{2\}$. Since he has the greatest budget, he belongs to C_1 and $u'(1) = 1$. In this case, his favorite products, 1 and 3, are circled, and the rest of the products in his list of preferences appear in grey because $x_2^1 = x_3^1 = 0$ by Proposition 3.10. As for customer 2, since $S_1^2 = \{2, 3\}$ and $2 \notin S_{u'(1)}^1 = \{1, 3\}$, he belongs to C_2, $u'(2) = 1$ and there exists an optimal solution in which customer 2 will not purchase product $1 \in S_2^2$.*

And finally, we turn to customer 5. For all $i \in I^5$, $\exists k \in K : \sigma(k) > \sigma(5)$ and $i \in S^k_{u'(k)}$. Hence, $5 \in C_3$, $u'(5) = n_5 = 3$ and none of the variables associated to the equivalence classes can be set to zero. Note that, in the optimal solution given, all the customers k purchase a product $i \in S^k_n$ such that $n \le u'(k)$.

Finally, we give a condition under which an optimal solution can be found by inspection.

Corollary 3.12. *If $C_3 = \emptyset$, an optimal solution of* (2ITM) *(resp.* (3ITM)*) can be found by inspection.*

Proof. We will prove the statement for formulation (2ITM), and the proof for (3ITM) is analogous. Let us define a solution $(\bar{v}^m_i, \bar{x}^k_n, \bar{z}^k_n)$ of (2ITM) and prove its optimality.

We begin by defining the vector $\bar{v}$ of prices in the following way: $\forall i \in I : i \in S^k_{u'(k)}$ for some $k \in K$, then $\bar{v}^{m_0}_i = 1$ for $m_0 := \max\{m \in M_i : \exists k \in K \text{ with } \sigma(k) = m_0, i \in S^k_{u'(k)}\}$, $\bar{v}^m_i = 0\ \forall m \neq m_0$; and $\forall i \in I$ such that $\{k \in K : i \in S^k_{u'(k)}\} = \emptyset$, then $\bar{v}^{m_0}_i = 1$ for $m_0 := \max\{m \in M_i\}$, $\bar{v}^m_i = 0\ \forall m \neq m_0$.

Now let us see the customers' purchase decision based on vector $\bar{v}$. Thus, given $k \in K$ we have that $\forall i \in S^k_n$ with $n < u'(k)$, it holds by definition of u' that $i \in S^k_{u'(k')}$ for some $k' : \sigma(k') > \sigma(k)$, and therefore $\bar{v}^m_i = 1$ for some $m > \sigma(k)$ and thus k cannot afford i. Hence, we have $\bar{x}^k_n = 0\ \forall n < u'(k)$. Moreover, since $k \notin C_3$, there exists $i_0 \in S^k_{u'(k)}$ such that $\bar{v}^{\sigma(k)}_{i_0} = 1$. This combined with the fact that $\forall i \in S^k_{u'(k)}$ it holds $\bar{v}^m_i = 1$ for some $m \ge \sigma(k)$ by definition of $\bar{v}$, implies that customer k purchases i_0, so $\bar{x}^k_{u'(k)} = 1$ and $\bar{z}^k_{u'(k)} = b^{\sigma(k)}$.

Given that the objective value of this above derived feasible solution is $\sum_{k \in K} b^{\sigma(k)}$, which is an upper bound on the profit the company can obtain, solution $(\bar{v}^m_i, \bar{x}^k_n, \bar{z}^k_n)$ is optimal. □

3.6 Computational results

Extensive computational experiments were carried out to compare the performance of (2ITM) and (BDM) in terms of the number of nodes of the branching tree, computational time and integrality gap, as well as the performance of the valid inequalities derived for both formulations and the preprocessing techniques. We implemented both formulations by means of Mosel version 4.0.3 of Xpress-MP, Optimizer version 29.01.10, running on a Dell PowerEdge T110 II Server (Intel Xeon E3-1270, 3.40 GHz) with 16 GB of RAM.

Regarding the instances, we modified those proposed in Chapter 2, that were designed following a model based on the Characteristics Model proposed by Fernandes et al. (2016). For each previous size ($|K| = 50$, $|K| = 100$ and $|K| = 150$ customers and $0.1|K|$, $0.5|K|$ and $|K|$ products), four types of instances were generated in Chapter 2 modifying $|I^k|$. Out of the four types, we consider the instances with sizes $|I^k| = \lceil 0.2|I| \rceil$, $|I^k| = \lceil 0.5|I| \rceil$ and $|I^k| = |I|$. These instances were proposed for the RPP, so we modified them by randomly adding ties in the ranked lists of preferences of the customers. Thus, for each size we generated three instances varying the number of ties in the list of preferences (denoted *Ties* in Tables 3.4-3.11 and in the following), with 1, 2, 3, 5 or 10 ties depending on the instance. This parameter establishes the relationship between $|I^k|$ and $n^k = |\mathscr{S}^k|$ in the following way: $|I^k| - \text{Ties} = n^k$. We modified 5 instances of each size, 365 in total.

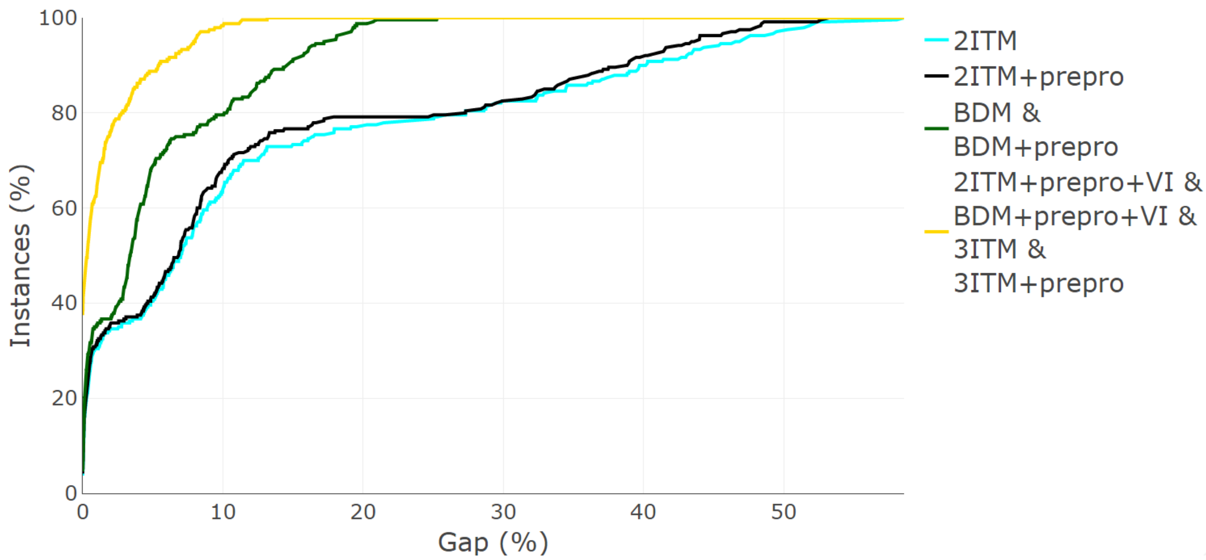

Figure 3.3: In the y-axis, the percentage of instances with an integrality gap less than or equal to that of the x-value is represented for formulations (2ITM), (BDM) and (3ITM), with the branch-and-cut procedures and/or the preprocessing techniques

All the instances are available at https://github.com/cdomsa/RPPT/. The time limit was set to 3600 seconds, and the default setting of Xpress was used.

To begin with, we consider the instances of sizes $|K| = 50$ and $|K| = 100$ to compare formulations (2ITM), (BDM) and (3ITM) with and without the preprocessing techniques and the valid inequalities included in a branch-and-cut procedure. For completeness, we report the results of the computational experiments in six tables. Tables 3.4-3.6 (resp. 3.7-3.9) contain all the results concerning the instances of sizes $|K| = 50$ (resp. $|K| = 100$). In the remaining of the section, the most significant information from those tables is summarized by means of several figures. Formulations (2ITM), (BDM) and (3ITM) are shown in the legends of the figures as 2ITM, BDM and 3ITM. If the instances have been preprocessed using the results from Section 3.5, in the legends it is shown 2ITM+prepro, BDM+prepro and 3ITM+prepro. And for formulations (2ITM) and (BDM) with the branch-and-cut procedure included (as well as the preprocessing techniques), they are shown as 2ITM+prepro+VI and BDM+prepro+VI.

Figure 3.3 is a performance profile that shows the percentage of instances having an integrality gap less than or equal to the value on the x-axis. For the formulations without the branch-and-cut procedure, the integrality gap is $\text{RLGap} = 100\frac{\text{UB}-\text{BV}}{\text{BV}}$, where UB represents the upper bound given by the linear relaxation and BV is the best objective value found by any of the models (the optimal value in most cases). As for formulations (2ITM) and (BDM) with the branch-and-cut procedure and the preprocessing techniques, the integrality gap represented corresponds to: $\text{RGap} = 100\frac{\text{UBC}-\text{BV}}{\text{BV}}$, where UBC is the upper bound obtained after adding the cuts in the root node. Figure 3.3 shows that the linear relaxation bound given by formulation (BDM) is in general much smaller than that of (2ITM), which in some cases goes up to a gap of 50%. Moreover, adding the preprocessing techniques only improves the gaps for formulation (2ITM), and the improvement is very small compared to the gaps provided by the rest of the formulations. The cuts added in the root node are very efficient in formulations (2ITM) and (BDM). Adding these cuts

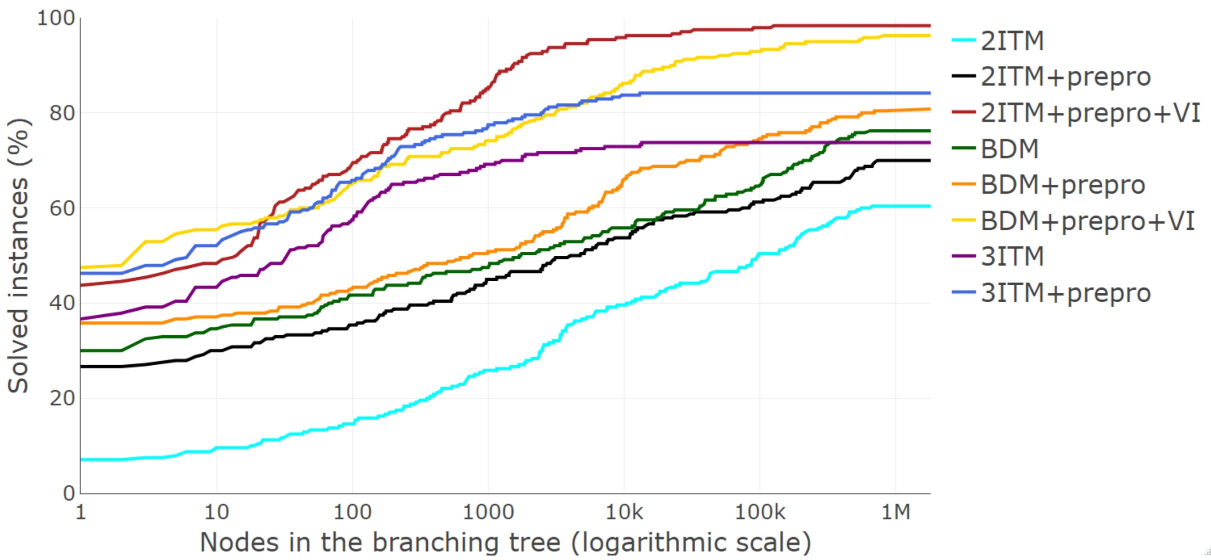

Figure 3.4: Percentage of solved instances depending on the number of nodes explored in the branching tree by formulations (2ITM), (BDM) and (3ITM), with the branch-and-cut procedures and/or the preprocessing techniques

leads to gaps 2-3% in 80% of the instances, and gaps smaller than 14% in all the instances. As we stated in Section 3.3, the upper bound in this case is in fact the bound provided by formulation (3ITM), and this is why the integrality gap is roughly the same for formulation (3ITM) and formulations (2ITM) and (BDM) with the branch-and-cut procedure (since the value BV used is the same in all cases). Hence, Figure 3.3 illustrates the decisive role of the valid inequalities derived in Sections 3.3 and 3.4 when reducing the upper bounds to close the integrality gap and reach optimality.

We also compared the performance of the eight resolution approaches in terms of the number of nodes explored during the branching process. Figure 3.4 shows the percentage of solved instances depending on the number of nodes explored in the branching tree. It is clear that (BDM) outperforms (2ITM) with and without the preprocessing, solving a greater percentage of instances by exploring the same amount of nodes, and that these two formulations with the preprocessing explore far less nodes than without these improvements. As for (3ITM), with the preprocessing it performs better than (2ITM) and (BDM). It is not so straightforward to compare the performance in terms of number of nodes between formulations (2ITM) and (BDM) with the valid inequalities. However, we can see that for greater number of nodes explored, (2ITM) slightly outperforms (BDM), since the former solves around 3% more instances than the latter.

Finally, the percentage of solved instances with respect to the time (up to a time limit of one hour) by the four models is illustrated in Figure 3.5. This figure shows results coherent with the previous ones, in the sense that it shows that formulation (BDM) outperforms (2ITM), but the opposite occurs if we consider the formulations with the valid inequalities and the preprocessing. It is remarkable how formulation (2ITM) solves 44% of the instances in less than 3600 seconds, whereas the same formulation with the improvements solves twice as many. As for formulation (3ITM), the percentage of solved instances shows that, even though it yields the tightest linear relaxation gaps, it performs worse than the two resolution strategies developed in the chapter. Furthermore, some

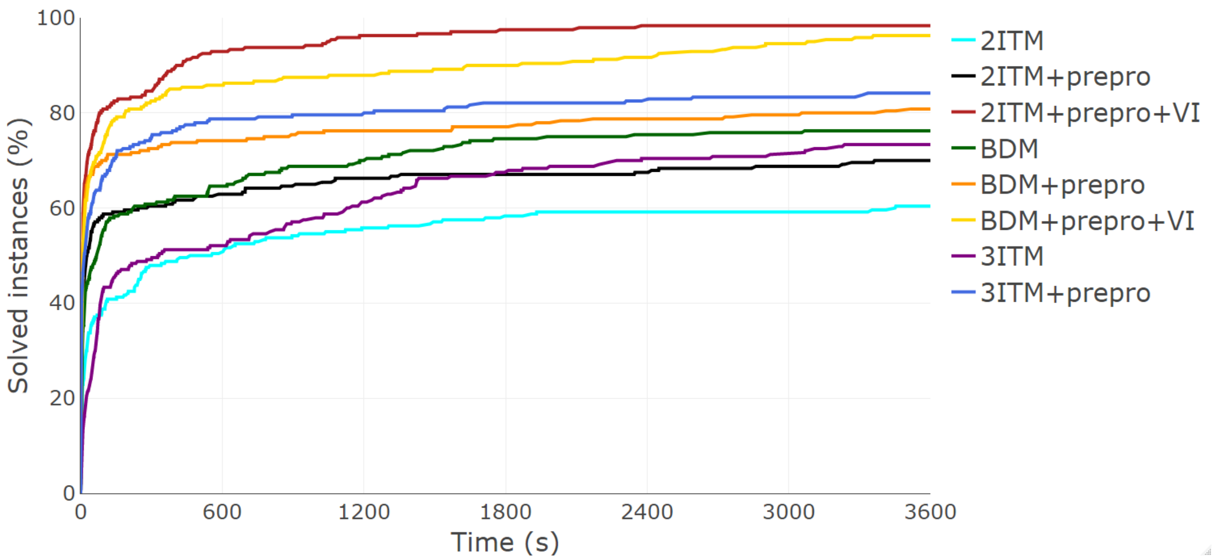

Figure 3.5: Percentage of instances solved (with a time limit of 3600 seconds) by formulations (2ITM), (BDM) and (3ITM), with the branch-and-cut procedures and/or the preprocessing techniques

of the instances with $|K| = 100$ could not be solved with this formulation without the preprocessing due to the extremely large amount of memory required by this formulation at this scale (the unsolved sizes are shown in Table 3.9).

$\|I\|$	$\|I^k\|$	Ties	(2ITM)								(2ITM)+prepro								(2ITM)+ VIs+prepro					
			Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Cuts	RGap	FGap	Nodes	t(s)	Sol.
5	2	1	192	255	2134	14.3	0.0	575	1	5	192	255	2134	14.3	0.0	575	2	5	226	8.2	0.0	73	3	5
5	3	1	331	479	5742	48.3	0.0	722	3	5	320	460	5276	43.0	0.0	1618	6	5	346	3.9	0.0	29	5	5
5	5	1	599	892	13790	36.3	0.0	3090	5	5	571	845	13790	35.1	0.0	10717	32	5	549	2.3	0.0	19	10	5
5	5	2	499	772	14647	43.1	0.0	4430	6	5	481	739	13386	41.0	0.0	191889	205	5	583	6.1	0.0	212	11	5
5	5	3	399	625	13333	49.2	0.0	3170	4	5	392	609	12619	46.3	0.0	360468	437	5	498	10.1	0.0	1131	10	5
25	5	1	638	913	5241	12.3	0.0	156	2	5	470	673	2977	7.6	0.0	151	6	5	168	0.2	0.0	1	1	5
25	5	2	538	792	4790	16.8	0.0	879	5	5	423	616	2975	11.8	0.0	466	8	5	232	0.4	0.0	6	1	5
25	5	3	438	650	4236	23.1	0.0	3758	10	5	386	560	3212	15.4	0.0	2245	13	5	335	1.7	0.0	74	4	5
25	13	1	1784	2521	29953	7.1	0.0	46268	119	5	1103	1558	12663	6.8	0.0	7229	31	5	502	0.3	0.0	13	5	5
25	13	3	1584	2310	28853	8.9	0.0	127533	316	5	1089	1576	14403	8.4	0.0	27434	89	5	591	0.3	0.0	9	5	5
25	13	5	1384	2095	27648	10.8	0.0	370924	977	5	977	1463	14240	10.3	0.0	128565	415	5	653	0.7	0.0	43	10	5
25	25	3	3225	4719	86705	7.0	0.4	477374	2999	2	1856	2710	35790	6.9	0.3	548264	2265	3	1278	0.7	0.0	60	24	5
25	25	5	3025	4511	92283	6.9	0.6	364368	2366	3	1883	2805	39867	6.9	0.5	402477	2022	3	1326	0.7	0.0	28	32	5
25	25	10	2525	3997	90613	9.8	2.4	725923	3600	0	1786	2788	46333	9.8	2.5	310656	3600	0	1397	1.2	0.0	445	43	5
50	10	1	1379	1943	12335	0.5	0.0	22	3	5	403	581	1528	0.4	0.0	1	0	5	28	0.0	0.0	1	0	5
50	10	3	1179	1733	11397	1.8	0.0	15	5	5	499	733	2476	1.3	0.0	1	1	5	77	0.1	0.0	1	0	5
50	10	5	979	1500	10319	4.3	0.0	3569	13	5	550	823	3555	2.9	0.0	120	8	5	169	0.2	0.0	2	1	5
50	25	3	3319	4744	62845	0.4	0.0	1370	17	5	617	895	3797	0.3	0.0	1	1	5	45	0.0	0.0	1	0	5
50	25	5	3119	4537	61104	0.4	0.0	502	16	5	742	1078	5543	0.3	0.0	3	2	5	61	0.0	0.0	1	0	5
50	25	10	2619	4018	56723	1.5	0.0	82853	144	5	837	1253	7770	1.4	0.0	607	15	5	201	0.1	0.0	7	3	5
50	50	3	6700	9747	204387	0.0	0.0	161813	758	4	697	1004	5463	0.0	0.0	1	0	5	12	0.0	0.0	1	0	5
50	50	5	6500	9545	205294	0.1	0.0	245928	751	4	737	1080	6398	0.0	0.0	1	0	5	23	0.0	0.0	1	0	5
50	50	10	6000	9036	209160	0.4	0.3	955965	2891	1	967	1425	10251	0.4	0.0	65	11	5	180	0.0	0.0	2	2	5

Table 3.4: Comparison of models (2ITM), (2ITM) with the preprocessing techniques ((2ITM)+prepro), and (2ITM) including the branch-and-cut method and the preprocessing techniques described in Section 3.5 ((2ITM)+VIs+prepro) (5 instances averaged per line). All instances have $|K| = 50$ customers, and the table shows the number of products ($|I|$), the number of products in which every customer is interested ($|I^k|$) and the number of ties (Ties). Depending on the model, it also includes the number of variables (Var.), constraints (Const.) and non-zero elements (Elem.), the integrality gap of the linear relaxation (LRGap), the average number of valid inequalities added in total (Cuts), the integrality gap of the linear relaxation after the cuts in the root node (RGap), the integrality gap after 3600 seconds (FGap), the number of nodes of the branching tree (Nodes), the average time in seconds needed to optimally solve the instances (t(s)) and the number of instances solved to optimality in less than the time limit of 3600 seconds. Notice that the number of variables, constraints and non-zero elements, as well as the integrality gap (LRGap), are the same for models (2ITM)+prepro and (2ITM)+VIs+prepro

$\|I\|$	$\|I^k\|$	Ties	**(BDM)**								**(BDM)+prepro**								**(BDM)+ VIs+prepro**					
			Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Cuts	RGap	FGap	Nodes	t(s)	Sol.
5	2	1	142	155	2133	14.3	0.0	57	1	5	142	155	2133	14.3	0.0	57	2	5	74	8.2	0.0	66	1	5
5	3	1	181	205	4915	11.1	0.0	640	1	5	177	200	4688	11.1	0.0	234	2	5	92	3.9	0.0	33	5	5
5	5	1	249	305	15122	11.4	0.0	13115	4	5	242	292	13989	11.4	0.0	1896	6	5	119	2.4	0.0	94	24	5
5	5	2	249	305	14285	15.5	0.0	10558	7	5	243	295	13446	15.5	0.0	2405	8	5	134	6.1	0.0	643	24	5
5	5	3	249	305	13001	20.4	0.0	10246	6	5	246	298	12545	20.3	0.0	3951	9	5	155	10.1	0.0	1200	15	5
25	5	1	288	325	3739	2.7	0.0	102	2	5	226	259	2319	2.7	0.0	48	5	5	51	0.2	0.0	1	1	5
25	5	2	288	325	3597	3.4	0.0	169	3	5	238	271	2515	3.4	0.0	107	2	5	82	0.4	0.0	2	2	5
25	5	3	288	325	3384	5.4	0.0	1305	5	5	262	296	2904	5.4	0.0	1179	3	5	122	1.7	0.0	73	2	5
25	13	1	634	725	38731	3.2	0.0	18367	45	5	414	474	15568	3.2	0.0	2426	16	5	93	0.3	0.0	2	9	5
25	13	3	634	725	37947	3.7	0.0	33133	89	5	457	521	19108	3.8	0.0	6804	11	5	128	0.3	0.0	13	14	5
25	13	5	634	725	36832	4.7	0.0	83401	138	5	467	533	19529	4.8	0.0	15765	30	5	160	0.7	0.0	38	12	5
25	25	3	1075	1325	212307	4.1	0.2	215833	1914	4	644	791	71739	4.1	0.0	73039	261	5	146	0.7	0.0	386	122	5
25	25	5	1075	1325	210790	3.9	0.2	446999	1681	4	691	852	84406	4.0	0.1	221484	1374	4	163	0.7	0.0	34	72	5
25	25	10	1075	1325	205431	5.5	1.5	717883	2661	2	789	953	108652	5.6	1.1	761684	2897	1	194	1.2	0.0	7501	88	5
50	10	1	529	600	9775	0.1	0.0	1	0	5	192	233	674	0.1	0.0	0	0	5	11	0.0	0.0	1	0	5
50	10	3	529	600	9542	0.5	0.0	1	0	5	256	308	1773	0.4	0.0	1	0	5	20	0.1	0.0	1	0	5
50	10	5	529	600	9124	1.5	0.0	114	2	5	326	383	3334	1.3	0.0	6	4	5	54	0.2	0.0	1	1	5
50	25	3	1169	1350	113321	0.3	0.0	1	2	5	269	328	2718	0.1	0.0	1	0	5	12	0.0	0.0	1	0	5
50	25	5	1169	1350	112602	0.2	0.0	1	3	5	327	392	5682	0.2	0.0	1	0	5	16	0.0	0.0	1	1	5
50	25	10	1169	1350	109692	0.7	0.0	102	7	5	421	489	11009	0.6	0.0	12	5	5	51	0.1	0.0	1	3	5
50	50	3	2050	2600	757484	0.0	0.0	1	14	5	280	345	4159	0.0	0.0	1	0	5	2	0.0	0.0	1	0	5
50	50	5	2050	2600	756127	0.1	0.0	1	14	5	296	374	6167	0.0	0.0	1	0	5	4	0.0	0.0	1	0	5
50	50	10	2050	2600	752457	0.3	0.0	15	48	5	397	485	14056	0.3	0.0	6	2	5	35	0.0	0.0	1	2	5

Table 3.5: Comparison of models (BDM), (BDM) with the preprocessing techniques ((BDM)+prepro), and (BDM) including the branch-and-cut method and the preprocessing techniques described in Section 3.5 ((BDM)+VIs+prepro) (5 instances averaged per line). All instances have $|K| = 50$ customers, and the table shows the number of products ($|I|$), the number of products in which every customer is interested ($|I^k|$) and the number of ties (Ties). Depending on the model, it also includes the number of variables (Var.), constraints (Const.) and non-zero elements (Elem.), the integrality gap of the linear relaxation (LRGap), the average number of valid inequalities added in total (Cuts), the integrality gap of the linear relaxation after the cuts in the root node (RGap), the integrality gap after 3600 seconds (FGap), the number of nodes of the branching tree (Nodes), the average time in seconds needed to optimally solve the instances (t(s)) and the number of instances solved to optimality in less than the time limit of 3600 seconds. Note that the number of variables, constraints and non-zero elements, as well as the integrality gap (LRGap), are the same for models (BDM)+prepro and (BDM)+VIs+prepro

$\|I\|$	$\|I^k\|$	Ties	(3ITM)								(3ITM)+prepro							
			Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.
5	2	1	873	2877	18286	8.2	0.0	73	9	5	873	2877	18286	8.2	0.0	73	5	5
5	3	1	1757	4742	67652	3.9	0.0	28	19	5	1641	4437	58620	3.9	0.0	39	8	5
5	5	1	4281	9239	354816	2.3	0.0	42	48	5	3871	8349	291712	2.3	0.0	53	43	5
5	5	2	3260	8218	285685	6.1	0.0	96	96	5	3021	7588	245931	6.1	0.0	223	79	5
5	5	3	2240	7198	207648	10.1	0.0	179	114	5	2151	6851	190345	10.1	0.0	178	87	5
25	5	1	1546	6679	32515	0.2	0.0	1	1	5	915	3953	12038	0.2	0.0	1	0	5
25	5	2	1422	6555	29949	0.4	0.0	2	3	5	935	4300	13113	0.4	0.0	2	2	5
25	5	3	1268	6401	26672	1.7	0.0	59	7	5	1023	5150	16925	1.7	0.0	49	9	5
25	13	1	8284	21545	922532	0.3	0.0	4	63	5	3393	8716	181612	0.3	0.0	10	18	5
25	13	3	7453	20714	864773	0.3	0.0	10	57	5	3702	10198	231554	0.3	0.0	9	19	5
25	13	5	6537	19797	801501	0.7	0.0	50	60	5	3422	10267	221575	0.7	0.0	23	56	5
25	25	3	24327	49857	8557096	0.7	0.0	106	1290	5	8491	17225	1408558	0.7	0.0	258	186	5
25	25	5	22209	47739	7846401	0.7	0.0	38	1210	5	8956	19121	1602913	0.7	0.0	65	167	5
25	25	10	16913	42443	6031283	1.2	0.0	3383	1680	5	8590	21365	1787953	1.2	0.0	3315	515	5
50	10	1	3195	13272	117161	0.0	0.0	1	2	5	461	2386	2892	0.0	0.0	1	0	5
50	10	3	2946	13023	109658	0.1	0.0	1	4	5	725	3606	7141	0.1	0.0	1	0	5
50	10	5	2652	12729	101058	0.5	0.0	1	5	5	1034	5189	14554	0.2	0.0	1	1	5
50	25	3	14978	40099	3030930	0.2	0.0	1	74	5	839	3068	10854	0.0	0.0	1	0	5
50	25	5	14191	39313	2951976	0.0	0.0	1	85	5	1202	4001	26591	0.0	0.0	1	0	5
50	25	10	11913	37034	2656153	0.2	0.0	2	194	5	1638	5421	55365	0.1	0.0	1	4	5
50	50	3	51143	101523	34316148	0.0	0.0	1	1208	5	958	3067	19051	0.0	0.0	1	0	5
50	50	5	49052	99432	33017305	0.1	0.0	1	1554	5	1045	3250	25250	0.0	0.0	1	0	5
50	50	10	43824	94204	29349112	0.1	0.0	1	1086	5	1603	4300	55638	0.0	0.0	1	3	5

Table 3.6: Comparison of models (3ITM) and (3ITM) with the preprocessing techniques ((3ITM)+prepro) (5 instances averaged per line). All instances have $|K| = 50$ customers, and the table shows the number of products ($|I|$), the number of products in which every customer is interested ($|I^k|$) and the number of ties (Ties). It also includes the number of variables (Var.), constraints (Const.) and non-zero elements (Elem.), the integrality gap of the linear relaxation (LRGap), the integrality gap after 3600 seconds (FGap), the number of nodes of the branching tree (Nodes), the average time in seconds needed to optimally solve the instances (t(s)) and the number of instances solved to optimality in less than the time limit of 3600 seconds

$\|I\|$	$\|I^k\|$	Ties	**(2ITM)**								**(2ITM)+prepro**								**(2ITM)+ VIs+prepro**					
			Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Cuts	RGap	FGap	Nodes	t(s)	Sol.
10	2	1	393	510	4486	7.8	0.0	3699	3	5	393	510	4486	7.8	0.0	3699	8	5	364	3.3	0.0	153	6	5
10	5	1	1249	1783	29446	37.9	0.0	159008	569	5	1186	1689	30897	36.0	0.3	312317	1569	4	1167	2.8	0.0	171	51	5
10	5	2	1049	1546	32957	42.5	0.0	223376	929	5	1005	1475	29941	39.6	2.3	558139	3305	2	1154	4.4	0.0	1223	71	5
10	5	3	849	1265	30464	48.4	0.0	171011	475	5	827	1225	28605	44.8	8.9	672538	3600	0	1269	7.2	0.0	8582	138	5
10	10	1	2604	3799	102559	28.4	4.9	618809	3600	0	2410	3516	107192	28.4	6.6	192550	3600	0	3002	3.1	0.0	596	309	5
10	10	3	2204	3363	121896	31.3	8.2	262796	3600	0	2057	3137	105999	31.1	10.6	167420	3600	0	2742	4.0	0.0	1754	372	5
10	10	5	1804	2916	118342	36.6	8.4	323288	3600	0	1699	2734	104117	36.3	17.4	167945	3600	0	2165	7.0	0.0	55411	1014	5
50	10	1	2751	3838	38541	7.9	0.5	565948	3130	2	1772	2471	17530	7.6	0.2	335108	2420	4	900	0.1	0.0	21	14	5
50	10	3	2351	3408	36419	10.1	1.7	769607	3600	0	1640	2362	18795	9.3	0.9	302467	2650	2	1003	0.4	0.0	44	17	5
50	10	5	1951	2955	34092	14.5	4.0	563317	3600	0	1515	2260	20812	12.6	2.7	259744	3600	0	1297	1.1	0.0	574	41	5
50	25	3	6621	9436	182474	6.1	2.7	195457	3600	0	4022	5701	84877	6.0	2.0	138000	3600	0	2667	0.6	0.0	437	179	5
50	25	5	6221	9021	196659	6.9	3.4	198695	3600	0	3752	5401	81166	6.9	2.9	143189	3600	0	2911	0.9	0.0	878	300	5
50	25	10	5221	7975	194885	9.4	5.6	193551	3600	0	3517	5336	93183	9.2	4.5	96063	3600	0	3034	1.4	0.1	6028	1310	4
50	50	3	13390	19445	624359	4.6	2.9	63323	3600	0	7450	10711	264937	4.6	2.6	49873	3600	0	6337	0.6	0.0	500	828	5
50	50	5	12990	19037	671452	4.9	3.1	69643	3600	0	7307	10600	266337	4.9	3.0	60680	3600	0	6192	0.7	0.1	2028	1642	4
50	50	10	11990	18026	726233	5.5	3.9	55020	3600	0	6960	10369	273581	5.5	3.5	45780	3600	0	6181	0.9	0.4	2564	2158	3
100	20	1	5717	7895	89067	0.2	0.0	206	31	5	1067	1517	5559	0.2	0.0	0	2	5	92	0.0	0.0	1	1	5
100	20	3	5317	7481	86112	0.3	0.0	17506	198	5	1261	1805	7785	0.3	0.0	25	7	5	166	0.0	0.0	1	2	5
100	20	5	4917	7069	83136	0.8	0.0	203085	1712	4	1442	2078	10633	0.7	0.0	48	20	5	293	0.0	0.0	1	2	5
100	50	3	13878	19493	407392	0.0	0.0	152676	1573	3	1477	2083	12503	0.0	0.0	1	2	5	131	0.0	0.0	1	2	5
100	50	5	13478	19091	424759	0.1	0.0	217922	2263	2	1625	2308	15110	0.1	0.0	7	6	5	214	0.0	0.0	2	3	5
100	50	10	12478	18072	456655	0.4	0.2	345287	3600	0	2305	3294	28349	0.3	0.0	995	30	5	447	0.0	0.0	5	8	5
100	100	3	27480	39497	1714316	0.0	0.0	19712	1764	3	1610	2303	18740	0.0	0.0	0	2	5	114	0.0	0.0	1	2	5
100	100	5	27080	39094	1697498	0.0	0.0	49335	3063	1	1699	2413	20598	0.0	0.0	6	5	5	144	0.0	0.0	1	2	5
100	100	10	26080	38089	1669818	0.1	0.1	33501	3178	1	2577	3638	44435	0.1	0.0	1502	43	5	406	0.0	0.0	1	9	5

Table 3.7: Comparison of models (2ITM), (2ITM) with the preprocessing techniques ((2ITM)+prepro), and (2ITM) including the branch-and-cut method and the preprocessing techniques described in Section 3.5 ((2ITM)+VIs+prepro) (5 instances averaged per line). All instances have $|K| = 100$ customers, and the table shows the number of products ($|I|$), the number of products in which every customer is interested ($|I^k|$) and the number of ties (Ties). Depending on the model, it also includes the number of variables (Var.), constraints (Const.) and non-zero elements (Elem.), the integrality gap of the linear relaxation (LRGap), the average number of valid inequalities added in total (Cuts), the integrality gap of the linear relaxation after the cuts in the root node (RGap), the integrality gap after 3600 seconds (FGap), the number of nodes of the branching tree (Nodes), the average time in seconds needed to optimally solve the instances (t(s)) and the number of instances solved to optimality in less than the time limit of 3600 seconds. Notice that the number of variables, constraints and non-zero elements, as well as the integrality gap (LRGap), are the same for models (2ITM)+prepro and (2ITM)+VIs+prepro

$\lvert I \rvert$	$\lvert I^k \rvert$	Ties	(BDM)								(BDM)+prepro								(BDM)+ VIs+prepro					
			Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Cuts	RGap	FGap	Nodes	t(s)	Sol.
10	2	1	293	310	4486	7.8	0.0	156	2	5	293	310	4486	7.8	0.0	156	3	5	131	3.3	0.0	145	4	5
10	5	1	549	610	34155	9.9	0.0	141037	147	5	527	585	31323	9.9	0.0	6963	26	5	270	2.8	0.0	2024	115	5
10	5	2	549	610	32307	12.6	0.0	176716	232	5	531	589	30087	12.6	0.0	7419	24	5	310	4.4	0.0	8046	228	5
10	5	3	549	610	29296	16.9	0.0	206363	193	5	538	598	28256	16.9	0.0	21906	59	5	426	7.3	0.0	24671	109	5
10	10	1	904	1110	177286	12.6	6.4	1180601	3600	0	844	1036	152404	12.6	1.3	526317	2692	2	354	3.1	0.2	201456	2179	4
10	10	3	904	1110	171147	13.9	7.3	1252562	3600	0	851	1045	149838	13.9	1.0	383198	2331	4	424	4.0	0.0	198660	2385	5
10	10	5	904	1110	161395	17.5	8.7	1082764	3415	1	859	1052	144148	17.5	3.2	589144	2894	2	407	7.0	1.7	550294	2705	3
50	10	1	1051	1150	41842	2.9	0.2	428422	1823	4	718	791	18740	3.0	0.0	494735	1010	5	236	0.1	0.0	1	14	5
50	10	3	1051	1150	40559	3.7	0.8	578994	2530	2	771	848	21389	3.7	0.5	753278	2242	2	284	0.4	0.0	389	20	5
50	10	5	1051	1150	38433	5.4	1.5	494869	3153	1	845	927	24861	5.4	1.2	577492	3336	1	423	1.1	0.0	1807	39	5
50	25	3	2321	2650	470818	3.6	2.5	103666	3600	0	1469	1659	178659	3.5	1.1	260578	3095	1	406	0.6	0.0	3332	269	5
50	25	5	2321	2650	467792	4.0	3.0	169278	3600	0	1457	1644	174387	4.0	1.7	166398	3600	0	460	0.9	0.0	10074	485	5
50	25	10	2321	2650	455880	5.4	4.3	314906	3600	0	1607	1826	210942	5.2	3.3	197917	3600	0	570	1.4	0.0	102590	1295	4
50	50	3	4090	5150	3104143	3.3	2.9	19146	3600	0	2381	2900	970259	3.2	2.3	59410	3600	0	475	0.6	0.0	6938	2276	5
50	50	5	4090	5150	3098997	3.6	3.2	26615	3600	0	2401	2930	987123	3.5	2.5	76590	3600	0	471	0.7	0.3	6523	2762	3
50	50	10	4090	5150	3083366	3.9	3.5	58328	3600	0	2461	3025	1037862	3.9	2.3	54110	3464	1	508	0.9	0.5	8344	3150	2
100	20	1	2017	2200	131396	0.1	0.0	1	4	5	466	557	3241	0.1	0.0	1	0	5	23	0.0	0.0	1	1	5
100	20	3	2017	2200	130255	0.1	0.0	1	3	5	565	665	6282	0.1	0.0	0	1	5	48	0.0	0.0	1	3	5
100	20	5	2017	2200	128915	0.4	0.0	480	18	5	677	781	11306	0.4	0.0	2	2	5	88	0.0	0.0	1	6	5
100	50	3	4578	5200	1699141	0.0	0.0	1	85	5	610	710	12040	0.0	0.0	0	0	5	30	0.0	0.0	1	4	5
100	50	5	4578	5200	1696267	0.1	0.0	3	65	5	669	783	17461	0.1	0.0	2	0	5	56	0.0	0.0	1	7	5
100	50	10	4578	5200	1687876	0.3	0.0	347	136	5	974	1102	50579	0.2	0.0	82	4	5	94	0.0	0.0	1	21	5
100	100	3	8180	10200	11737147	0.0	0.0	1	741	5	623	761	16201	0.0	0.0	1	0	5	24	0.0	0.0	1	5	5
100	100	5	8180	10200	11732410	0.0	0.0	1	1031	5	667	793	22094	0.0	0.0	1	0	5	28	0.0	0.0	1	5	5
100	100	10	8180	10200	11719548	0.1	0.0	318	1188	5	993	1136	91025	0.1	0.0	200	3	5	72	0.0	0.0	1	43	5

Table 3.8: Comparison of models (2ITM) and (BDM) with models (2ITM) and (BDM) including the branch-and-cut method and the preprocessing techniques described in Section 3.5 to all the instances (5 instances averaged per line). All instances have $|K| = 50$ customers, and the table shows the number of products ($|I|$), the number of products in which every customer is interested ($|I^k|$) and the number of ties (Ties). Depending on the model, it also includes the integrality gap of the linear relaxation (LRGap), the average number of valid inequalities added in total (Cuts), the integrality gap of the linear relaxation after the cuts in the root node (RGap), the integrality gap after 3600 seconds (FGap), the number of nodes of the branching tree (Nodes), and the average time needed to optimally solve the instances (t(s)), and the number of instances solved to optimality in less than the time limit of 3600 seconds. Notice that, although the integrality gap (LRGap) only appears in model (2ITM) (resp. (BDM)), it is the same for model (2ITM) (resp. (BDM)) with the valid inequalities and preprocessing

$\|I\|$	$\|I^k\|$	Ties	**(3ITM)** Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	**(3ITM)+prepro** Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.
10	2	1	2024	10028	43716	3.3	0.0	85	15	5	2024	10028	43716	3.3	0.0	85	19	5
10	5	1	10717	30594	1052302	2.8	0.0	1117	392	5	9680	27637	861290	2.8	0.0	1400	392	5
10	5	2	9096	28974	960846	4.4	0.0	2733	875	5	8356	26629	805440	4.4	0.0	2731	611	5
10	5	3	7137	27015	841677	7.3	0.2	5110	1865	4	6777	25710	752836	7.3	0.5	5971	1733	4
10	10	1	37195	76935	10947633	3.1	2.0	2203	3527	1	31931	66027	8310840	3.1	1.7	4915	3489	2
10	10	3	29108	68848	8781025	4.0	2.9	2728	3600	0	25401	60137	6856567	4.0	2.7	2726	3600	0
10	10	5	21021	60761	6649939	7.0	5.6	3194	3600	0	18657	53992	5299679	7.0	5.5	3090	3600	0
50	10	1	11023	51306	812215	0.1	0.0	6	50	5	4979	23135	181666	0.1	0.0	2	15	5
50	10	3	10243	50526	819447	0.6	0.0	149	121	5	5387	26550	226671	0.4	0.0	63	106	5
50	10	5	9217	49500	831763	1.1	0.0	727	559	5	5840	31409	309317	1.1	0.0	893	253	5
50	25	3	55827	156466	23026505	0.7	0.3	317	2990	3	21717	60907	4489254	0.6	0.1	1662	1704	4
50	25	5	52864	153502	22630712	1.0	0.8	545	3600	0	20285	58675	4234851	0.9	0.3	3659	2993	2
50	25	10	44266	144905	20960097	1.6	1.4	172	3600	0	21141	68811	5509201	1.4	1.1	3511	3600	0
50	50	3	-	-	-	-	-	-	-	-	63795	128454	42201681	0.6	0.5	6	3600	0
50	50	5	-	-	-	-	-	-	-	-	62418	128434	41485972	0.7	0.6	71	3600	0
50	50	10	-	-	-	-	-	-	-	-	59306	129637	41221665	0.9	0.8	15	3600	0
100	20	1	22538	104477	3143550	0.0	0.0	1	53	5	1369	9948	14558	0.0	0.0	1	0	5
100	20	3	21833	103773	3219012	0.0	0.0	1	91	5	1894	12225	27826	0.0	0.0	1	1	5
100	20	5	21018	102957	3271394	0.0	0.0	1	39	5	2631	15522	62318	0.0	0.0	1	1	5
100	50	3	-	-	-	-	-	-	-	-	2258	11274	57809	0.0	0.0	1	1	5
100	50	5	114394	319256	94102376	0.0	0.0	1	1953	5	2679	12368	89959	0.0	0.0	1	2	5
100	50	10	106726	311588	94417488	0.1	0.0	0	2488	4	4937	18053	319214	0.0	0.0	2	8	5
100	100	3	-	-	-	-	-	-	-	-	2328	10834	61538	0.0	0.0	1	2	5
100	100	5	-	-	-	-	-	-	-	-	2655	11419	113196	0.0	0.0	1	2	5
100	100	10	-	-	-	-	-	-	-	-	5306	16188	541860	0.0	0.0	1	16	5

Table 3.9: Comparison of models (3ITM) and (3ITM) with the preprocessing techniques ((3ITM)+prepro) (5 instances averaged per line). All instances have $|K| = 100$ customers, and the table shows the number of products ($|I|$), the number of products in which every customer is interested ($|I^k|$) and the number of ties (Ties). It also includes the number of variables (Var.), constraints (Const.) and non-zero elements (Elem.), the integrality gap of the linear relaxation (LRGap), the integrality gap after 3600 seconds (FGap), the number of nodes of the branching tree (Nodes), the average time in seconds needed to optimally solve the instances (t(s)) and the number of instances solved to optimality in less than the time limit of 3600 seconds

Formulations (2ITM) and (BDM) solved to optimality the majority of the instances with 50 customers, and the same formulations including the branch-and-cut and the preprocessing solved all of them and nearly all the instances with $|K| = 100$. Hence, we decided to compare our two best resolution methods with the instances of greater size, those with 150 customers. Table 3.10 and 3.11 report the results of the comparison of formulations (2ITM) and (BDM) with formulations (2ITM) and (BDM) including the branch-and-cut method and the preprocessing techniques. Many of the instances could not be solved with formulation (3ITM) due to the requirement of an extensive amount of memory, so (3ITM) is not included in the comparison. Tables 3.10 and 3.11 show the number of products ($|I|$) of the instances, the number of products in which every customer is interested ($|I^k|$) and the number of ties (Ties). Depending on the model, they also feature the integrality gap of the linear relaxation (LRGap), the average time (in seconds) needed to optimally solve the instances (t(s)), and the number of instances solved to optimality in less than the time limit of 3600 seconds.

As we can see, the relationship between the number of customers $|K|$ and products $|I|$ determines the difficulty of the instance: the instances with $|I| = 15$ and $|I| = 75$ take generally much more computational time. Only (2ITM) and (BDM) with the branch-and-cut and preprocessing are able to solve some of the instances with $|K| = 150$ and $|I| = 75$. The fact that they are more difficult than those with 15 customers is explained because the preference matrices of the latter ones are less dense and they have a much smaller number of variables and constraints, so the branch-and-cut and the branching procedures are faster. As for the instances with $|K| = |I|$ (the ones at the bottom of each table), they take less time to solve due to the preprocessing techniques, which eliminate a great number of customer decision variables when the number of products is large compared to the number of customers. Within the instances with the same amount of customers and products, the increase in the numbers of products in the list of preferences of each customer ($|I^k|$) also increases the difficulty of the instance, as well as the growth in the number of Ties.

Overall, it is clear that the preprocessing techniques, and specially the branch-and-cut procedures applied, constitute a major improvement in the performance of both (2ITM) and (BDM). The cuts added in the root node are essential to reduce the integrality gap and allow for the resolution of up to twice as many instances. Comparing the two best formulations without the upgradings, it can be seen that the linear relaxation gap is always smaller for formulation (BDM) than for (2ITM). However, from the number of nodes explored in the branching tree, the average time, the number of instances solved and the final integrality gaps, it is clear that formulation (2ITM) slightly outperforms formulation (BDM) when the preprocessing and the valid inequalities are included. The reason is that computing the valid inequalities for formulation (BDM) is harder and time consuming. Indeed, we compute one inequality for each customer for (BDM), but we obtain one inequality per customer and product in the case of (2ITM). The fact that valid inequalities added to (2ITM) can be separated by products makes the processes of computing the inequalities and branching a lot more efficient.

$\lvert I\rvert$	$\lvert I^k\rvert$	Ties	(2ITM)								(2ITM)+VIs+prepro								
			Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Var.	Const.	Elem.	Cuts	RGap	FGap	Nodes	t(s)	Sol.
15	3	1	1028	1439	18119	46.2	0.3	558241	1609	3	990	1382	17242	1029	2.3	0.0	771	37	5
15	8	1	3155	4489	108783	29.5	7.6	449124	3600	0	2934	4165	113693	3415	3.1	0.0	4607	770	5
15	8	2	2855	4169	130052	31.5	9.6	203008	3600	0	2686	3910	114439	3275	3.8	0.0	15601	1983	5
15	8	3	2555	3825	127240	34.3	11.2	238364	3600	0	2407	3591	112013	3098	5.0	0.8	22502	2570	2
15	15	1	5967	8706	326637	24.1	12.1	155967	3600	0	5446	7916	346657	7822	3.9	2.4	3952	3593	1
15	15	3	5367	8076	413086	26.0	14.5	51748	3600	0	4936	7401	347075	7258	4.8	3.3	7594	3403	1
15	15	5	4767	7439	407277	27.6	17.2	42113	3600	0	4425	6873	348163	6563	5.2	4.4	8586	3600	0
75	15	1	6332	8764	114317	6.2	2.0	194129	3600	0	3838	5331	50537	2004	0.4	0.0	174	67	5
75	15	3	5732	8141	117091	7.4	2.6	264952	3600	0	3673	5222	53224	2236	0.4	0.0	250	87	5
75	15	5	5132	7506	116070	9.1	3.9	262027	3600	0	3593	5241	59716	2470	0.7	0.0	599	184	5
75	38	3	15484	21963	588959	4.7	2.7	70717	3600	0	8613	12234	240080	5992	0.7	0.3	2409	2243	3
75	38	5	14884	21352	633155	5.3	3.2	70146	3600	0	8607	12364	252728	6522	0.9	0.5	3516	2841	2
75	38	10	13384	19823	666236	6.0	4.0	78166	3600	0	8170	12100	267623	6526	0.9	0.5	2948	2811	2
75	75	3	30285	44170	1931936	4.7	3.6	17219	3600	0	16147	23509	776235	14382	1.6	1.4	474	3600	0
75	75	5	29685	43566	2090607	4.1	3.0	13893	3600	0	15994	23418	783569	13787	1.0	0.8	506	3600	0
75	75	10	28185	42049	2316487	5.0	3.8	15908	3600	0	15719	23390	813468	14346	1.5	1.4	504	3600	0
150	30	3	12399	17232	263960	0.1	0.0	9414	172	5	2131	3033	15167	165	0.0	0.0	1	4	5
150	30	5	11799	16622	266082	0.2	0.0	90355	1248	4	2334	3346	18508	249	0.0	0.0	1	5	5
150	30	10	10299	15074	259868	0.7	0.4	260290	3600	0	3318	4848	36911	638	0.0	0.0	4	12	5
150	75	3	31641	44244	1357318	0.0	0.0	63340	2480	2	2338	3315	22617	167	0.0	0.0	1	5	5
150	75	5	31041	43637	1374458	0.0	0.0	89260	3136	1	3116	4411	39343	190	0.0	0.0	1	6	5
150	75	10	29541	42127	1479226	0.1	0.1	100169	3600	0	3478	4968	48703	410	0.0	0.0	1	11	5
150	150	3	61980	89247	5753533	0.0	0.0	2438	2997	3	2246	3176	27484	113	0.0	0.0	1	5	5
150	150	5	61380	88645	5712502	0.0	0.0	8389	3600	0	2723	3859	41706	164	0.0	0.0	1	8	5
150	150	10	59880	87141	5610723	0.0	0.0	7753	3600	0	3624	5217	75929	401	0.0	0.0	1	18	5

Table 3.10: Comparison of models (2ITM) and (2ITM) including the branch-and-cut method and the preprocessing techniques described in Section 3.5 ((2ITM)+VIs+prepro) (5 instances averaged per line). All instances have $|K| = 150$ customers, and the table shows the number of products ($|I|$), the number of products in which every customer is interested ($|I^k|$) and the number of ties (Ties). Depending on the model, it also includes the number of variables (Var.), constraints (Const.) and non-zero elements (Elem.), the integrality gap of the linear relaxation (LRGap), the average number of valid inequalities added in total (Cuts), the integrality gap of the linear relaxation after the cuts in the root node (RGap), the integrality gap after 3600 seconds (FGap), the number of nodes of the branching tree (Nodes), the average time in seconds needed to optimally solve the instances (t(s)) and the number of instances solved to optimality in less than the time limit of 3600 seconds. Notice that the integrality gap (LRGap) is the same for models (2ITM) and (2ITM)+VIs+prepro

$\lvert I\rvert$	$\lvert I^k\rvert$	Ties	(BDM)								(BDM)+VIs+prepro								
			Var.	Const.	Elem.	LRGap	FGap	Nodes	t(s)	Sol.	Var.	Const.	Elem.	Cuts	RGap	FGap	Nodes	t(s)	Sol.
15	3	1	578	615	16188	8.5	0.0	46215	72	5	563	598	15318	335	2.3	0.0	3020	54	5
15	8	1	1205	1365	163047	11.5	7.2	900227	3600	0	1135	1279	142751	583	3.1	0.2	206063	1647	4
15	8	2	1205	1365	159294	12.6	7.7	738058	3600	0	1147	1293	142874	643	3.8	0.7	202375	2321	3
15	8	3	1205	1365	154624	14.4	9.2	615985	3600	0	1147	1293	138598	725	5.0	1.5	197529	2949	2
15	15	1	1917	2415	763326	13.2	11.7	235058	3600	0	1777	2211	642871	414	4.9	4.2	31246	3600	0
15	15	3	1917	2415	751428	14.5	12.9	170589	3600	0	1785	2228	639680	484	5.3	4.7	38046	3600	0
15	15	5	1917	2415	736042	15.3	13.5	193678	3600	0	1801	2245	638659	550	5.4	5.1	30173	3600	0
75	15	1	2282	2475	183747	3.0	1.9	159600	3600	0	1444	1588	70375	395	0.4	0.0	1098	206	5
75	15	3	2282	2475	181003	3.4	2.1	158060	3600	0	1516	1666	77219	474	0.4	0.0	1473	113	5
75	15	5	2282	2475	177148	4.4	2.9	160915	3600	0	1646	1802	90544	578	0.7	0.0	5956	185	5
75	38	3	5134	5925	2266873	3.1	2.8	48331	3600	0	2927	3395	711300	638	0.7	0.2	43386	2940	3
75	38	5	5134	5925	2260209	3.5	3.2	52101	3600	0	3033	3519	763355	697	0.9	0.4	26709	3078	2
75	38	10	5134	5925	2239892	3.9	3.6	55390	3600	0	3193	3704	842504	776	0.9	0.5	18824	3519	1
75	75	3	8835	11475	14386886	3.8	3.7	1113	3600	0	4816	6205	4171180	276	2.4	2.3	5	3600	0
75	75	5	8835	11475	14376325	3.2	3.1	1351	3600	0	4869	6265	4267463	301	1.8	1.6	34	3600	0
75	75	10	8835	11475	14347859	4.0	3.8	1352	3600	0	5029	6479	4565250	294	2.4	2.3	4	3600	0
150	30	3	4449	4800	618676	0.1	0.0	1	25	5	900	1063	12929	45	0.0	0.0	1	4	5
150	30	5	4449	4800	615724	0.1	0.0	6	32	5	1013	1183	19082	60	0.0	0.0	1	7	5
150	30	10	4449	4800	606094	0.4	0.0	10209	343	5	1550	1741	59653	167	0.0	0.0	2	20	5
150	75	3	10191	11550	8282888	0.0	0.0	0	495	5	929	1105	20580	37	0.0	0.0	1	7	5
150	75	5	10191	11550	8276411	0.0	0.0	2	536	5	1202	1406	57576	47	0.0	0.0	1	7	5
150	75	10	10191	11550	8260023	0.1	0.0	163	859	5	1377	1601	86963	86	0.0	0.0	1	17	5
150	150	3	18030	22800	56975718	0.0	0.0	0	3600	0	894	1062	24022	24	0.0	0.0	1	6	5
150	150	5	18030	22800	56965475	0.0	0.0	0	3600	0	1046	1245	57155	28	0.0	0.0	0	8	5
150	150	10	18030	22650	103220061	0.0	0.0	0	3600	0	1324	1617	158907	71	0.0	0.0	0	29	5

Table 3.11: Comparison of models (BDM) and (BDM) including the branch-and-cut method and the preprocessing techniques described in Section 3.5 ((BDM)+VIs+prepro) (5 instances averaged per line). All instances have $|K| = 150$ customers, and the table shows the number of products ($|I|$), the number of products in which every customer is interested ($|I^k|$) and the number of ties (Ties). Depending on the model, it also includes the number of variables (Var.), constraints (Const.) and non-zero elements (Elem.), the integrality gap of the linear relaxation (LRGap), the average number of valid inequalities added in total (Cuts), the integrality gap of the linear relaxation after the cuts in the root node (RGap), the integrality gap after 3600 seconds (FGap), the number of nodes of the branching tree (Nodes), the average time in seconds needed to optimally solve the instances (t(s)) and the number of instances solved to optimality in less than the time limit of 3600 seconds. Notice that the integrality gap (LRGap) is the same for models (BDM) and (BDM)+VIs+prepro

Ins	(2ITM)+VIs+prepro LR phase Best Bound	Time (s)	Cuts phase Best Bound	Time (s)	MIO phase Obj	Best Bound	Time (s)	Nodes	Total time (s)
1	217071	0.5	155889	3534.5	148414	148414	8575.4	21113	12109.9
2	216549	0.6	154642	5482.7	143469	143483	72255.2	276914	77737.9
Ins	**(BDM)+ VIs+prepro** **LR phase** Best Bound	Time (s)	**Cuts phase** Best Bound	Time (s)	**MIO phase** Obj	Best Bound	Time (s)	**Nodes**	**Total time (s)**
1	172006	1.1	155889	21540.3	148414	148414	2103.7	23279	23644
2	170810	0.9	154642	21805.8	143469	143481	3487.1	120415	25292.9

Table 3.12: Results of two large-scale instances ($|K| = 350$, $|I| = 10$) given by formulations (2ITM) and (BDM) including the branch-and-cut method and the preprocessing techniques. The *LR phase* of the table shows the bound and time of the linear relaxation phase. The *Cuts phase* includes the bound after the cuts in the root node and the time to generate them. And the *MIO phase* shows the best solution (Obj), the best bound and the time. We set a final integrality gap of 0.01% or lower for this integer phase. Finally, the table shows the number of nodes explored in the branching tree and the total time in seconds

Motivated by the results obtained by Bertsimas and Mišić (2019) with a Benders decomposition procedure to tackle the PLD problem, we decided to test the performance of our resolution schemes using some large-scale instances. We randomly generated two instances of the RPPT with 350 customers, all with different budgets, and 10 products. We tested both instances with formulations (2ITM) and (2ITM) including the corresponding branch-and-cut procedures and the preprocessing techniques, and the results are shown in Table 3.12.

Table 3.12 shows that the time needed to solve the Cuts phase is much smaller for formulation (2ITM), with times of around an hour for the first instance and an hour and a half for the second. Formulation (BDM), on the contrary, takes nearly six hours to add the cuts in the Cuts phase. These results are consistent with the ones obtained in the previous experiment.

Nonetheless, we can see a different performance in the MIO phase. Formulation (BDM) takes less than an hour to close the gap and reach optimality for both instances. Regarding instance 1, the MIO phase for formulation (2ITM) takes two hours and a half. But for instance 2, this phase takes 72255 seconds, i.e. more than 20 hours. Comparing the number of nodes explored during the MIO phase with the time taken to solve instance 1, we see that both formulations explore a similar amount of nodes, but formulation (2ITM) takes four times longer. We observe a similar pattern for instance 2. Therefore, it is clear that exploring a node is much faster for formulation (BDM) than for (2ITM), and this is decisive in the reduction of the MIO phase time.

Chapter 4

The Capacitated Rank Pricing Problem with envy

In this chapter, we tackle an extension of the RPP that we have named the Capacitated Rank Pricing Problem with envy. In the first place, we assume that each customer has a different reservation price that reflects his willingness to pay for each product of his interest, and therefore he purchases the highest-ranked product among the ones priced under their corresponding reservation prices. Furthermore, we assume that the company can only offer a limited amount of copies of each product for sale and might not have enough supply to satisfy its clients. Limiting the number of copies of each product implies that in some solutions, some customers are unable to purchase their favorite product because it has sold out, even if they can afford it. This results in two versions of the problem that differ in the type of solution sought by the company.

On the one hand, the company may opt for a solution in which its profit is maximized restricting the search to the solutions that avoid (possible) conflicts among customers. In this case, it shall choose what is known as an envy-free solution, that is, a solution in which it can provide to each customer the product he prefers the most among all those that have a price lower than his reservation price. An envy-free solution always exists, it suffices to consider the solution where the products are very expensive and no customer can afford them, so there is no possible envy among customers. On the other hand, the company may choose to maximize its profit regardless of possible conflicts among customers. Such conflicts arise when a customer prefers a product more than the one he is given and he can afford it, but the product is not available because all the copies have been sold to other customers. In some settings, envy solutions can lead to dissatisfied customers, what can result in the loss of clients in the long term. The concept of envy-freeness and its introduction in the pricing literature is further discussed in subsection 1.2.2 of Chapter 1.

Although an envy-free pricing (i.e. a pricing that admits a feasible, envy-free allocation of products) prevents the customers' possible displeasure with their purchase, it generally provides a smaller profit for the company than a solution that allows for envy. Furthermore, like in the RPP, if the pricing is envy-free the allocation of the products to the customers is uniquely determined. Thus, any envy-free pricing leads to a unique feasible allocation of the products. However, if the envy-freeness is not required, a pricing can lead to multiple feasible allocations of the products to the customers that yield different profit. As a consequence, a feasible solution in the envy-free setting is also a solution for the envy setting, and moreover the version of the CRPP that allows for envy is inherently more difficult than the envy-free one. We address the resolution of the envy version of the CRPP by means of discrete optimization methods, basing our results on the ones developed in Chapters 2 and 3.

We introduce two integer linear formulations for the CRPP, one with three-index customer decision variables and a second one that makes use of a much smaller set of variables but generally provides worse linear relaxation bounds. The three-index formulation is strengthened by means of alternative sets of constraints derived leveraging the capacity constraints and the three-index variables. As for the reduced formulation, three families of valid inequalities of exponential size are presented along with their respective separation procedures to include them dynamically in branch-and-cut frameworks. We then compare the performance of both models and the different resolution algorithms described by means of extensive computational experiments.

The rest of the chapter is organized as follows. In Section 4.1, we introduce the notation, explain the difference between the envy-free and the envy case by means of an example, and prove that the assignment of the CRPP with envy is NP-complete. In Section 4.2, we introduce the three-index integer linear formulation and several families of valid inequalities derived to strengthen it. Section 4.3 is devoted to the presentation of the reduced formulation. In Section 4.4 we include the families of valid inequalities for the reduced model and the separation procedures developed to incorporate constraints in a branch-and-cut fashion. Finally, Section 4.5 includes the computational study.

4.1 Notation and problem description

Recall that the CRPP aims at maximizing the profit of a company selling different products with a limited supply, taking into account the preferences and reservation prices of the unit-demand customers. Since we tackle the version of the problem that allows for envy, once the prices are settled, clients are assigned their highest-ranked product among the ones that are not sold out and have a price below the corresponding reservation price (if any).

Like in previous chapters, let us define $K = \{1, \dots, |K|\}$ as the set of customers and $I = \{1, \dots, |I|\}$ as the set of products. Each customer $k \in K$ has a subset of products $I^k \subseteq I$ he could potentially purchase, and he ranks the products in I^k from the *best* to the *worst* (ties in the ranking are not allowed). If k ranks product i higher than product j, we say that k *prefers* i to j, and we denote $i \prec_k j$. Furthermore, $\{j \in I^k : j \preceq_k i\} = \{i\} \cup \{j \in I^k : j \prec_k i\}$. For a given product $i \in I$, we define $K_i := \{k \in K : i \in I^k\}$ as the set of customers that could purchase product $i \in I$. Without loss of generality, we assume $I^k \neq \emptyset \ \forall k \in K$, $K_i \neq \emptyset \ \forall i \in I$.

In this extension, each customer k has a reservation price for each product $i \in I^k$, instead of a fixed budget. Therefore, the notation concerning the reservation prices varies slightly with respect to that of Chapters 2 and 3. Since different customers may have the same reservation price for the same product i, we define set $M_i = \{1, \dots, |M_i|\}$ as the set of indices that refer to the different reservation prices of the customers, and $(b_i^m)_{m \in M_i}$ as the set of different reservation prices for i, so that $b_i^{m_1} < b_i^{m_2}$ if $m_1 < m_2$. To represent the reservation prices of a customer for a product, we define a function $\sigma_i : K_i \to M_i\ \forall i \in I$ such that $\sigma_i(k) = m$ iff the reservation price of k is the m-th smallest reservation price b_i^m. Following the notation and taking into account that, like previously, there always exists an optimal solution of the CRPP in which the price of any product i is equal to the reservation price of a customer for such product, $M_i^k := \{m \in M_i : m \leq \sigma_i(k)\}$ represents the set of indices m of candidate prices b_i^m at which k could purchase i in a feasible solution. Finally, let c_i represent the number of copies for sale of product i, $\forall i \in I$.

As explained in the introduction, the feasible solutions of the CRPP differ depending on the setting that we choose. If we solve the envy-free version of the CRPP, then for a fixed pricing there is a unique feasible allocation of the products to the customers because each of them purchases the highest-ranked product below his reservation price (if any). On the other hand, if we allow for envy in the solutions, different allocations with different revenues may exist. In the following, we illustrate the difference between an envy-free solution of the CRPP and a solution that allows for envy by means of the example of Figure 4.1.

	Prod. 1	Prod. 2
	$c_1 = 2$	$c_2 = 2$
Customer 1	2 | 5	1* | 3
Customer 2	1* | 4	2 | 4
Customer 3	2* | 3	1 | 2
Opt. price	3	3

(a) Envy-free solution

	Prod. 1	Prod. 2
	$c_1 = 2$	$c_2 = 2$
Customer 1	2* | 5	1 | 3
Customer 2	1 | 4	2* | 4
Customer 3	2* | 3	1 | 2
Opt. price	3	4

(b) Solution that allows for envy

Figure 4.1: Optimal solutions of an instance of the CRPP

Example 4.1. *Figure 4.1a shows an instance of the CRPP problem with $|K| = 3$ customers and $|I| = 2$ products. The number of copies c_i of each product i is represented right below the product. In our instance, there are two copies of each product for sale. The left number in an entry represents the preference of the customer for that product. If product i is the highest ranked product for customer k, then the number represented is 1; and for his second product, the number is 2. The right number in each entry corresponds to the reservation price of that customer for that product. In this example, customers 1, 2 and 3 have reservation prices of 3, 4 and 2 for product 2, respectively. Therefore, the number of different reservation prices for product 2 is $|M_2| = 3$. As for the ordered reservation prices, they are $b_2^1 = 2$, $b_2^2 = 3$ and $b_3^2 = 4$. Lastly, the reservation prices for product 2 are related to the customers by means of function σ_2. For instance for customer 1 it holds $\sigma_2(1) = 2$ because he owns the second lowest reservation price for product 2. Similarly, $\sigma_2(2) = 3$ and $\sigma_2(3) = 1$. The optimal envy-free solution is obtained setting the prices depicted at the bottom of the table. Customers purchase the product whose preference is* *marked with an asterisk in the preference matrix, and the total revenue of the company is equal to 9.*

However, the revenue of the company increases if we allow for envy among customers. Figure 4.1b contains the same instance with the optimal solution in the envy setting. In this solution, customer 2 is allocated product 2 even though product 1 is under his reservation price and he prefers 1 to 2, because product 1 is sold out. Thus, this allocation of products is not feasible for the envy-free setting. The optimal value in this case is equal to 10. A different allocation of products (with the same pricing and with envy) that yields a smaller optimal value consists in assigning product 1 to customers 1 and 2. Product 2 remains unsold because customer 2 is purchasing product 1 and customers 1 and 3 have a reservation price below 4. In this case, the revenue is equal to 6.

We have just seen that, given a pricing, different allocations (in the envy setting) may have different revenue. This leads to the question of whether there exists an efficient algorithm for the problem of finding an allocation of maximum revenue in the restricted version of the CRPP where the prices are fixed. In the following subsection, we show that this problem is in fact NP-complete.

4.1.1 Complexity of the non envy-free assignment problem of the CRPP

In Rusmevichientong et al. (2006) it is proved that the RPP is NP-complete in the strong sense, and hence so is the CRPP. We now consider a fixed pricing of the products in the CRPP, and study the assignment of the products to the customers. It is clear that in the envy-free case, the assignment of the products is straightforward: each customer purchases the highest-ranked product in his list below his reservation price (if any). In this subsection, however, we prove that the non envy-free assignment of the CRPP is NP-complete. To do so, we consider a restricted case of the CRPP where all the prices of the products are fixed to one single price p and the number of copies of each product is equal to one. We begin the subsection introducing a matching problem called the Stable Marriage problem with Ties and Incomplete Lists (SMTI). We then reduce an NP-hard particular case of the SMTI to our problem to prove the result. We refer the reader to subsection 1.2.3 of Chapter 1 for a brief introduction to bipartite matching problems with preferences.

An instance of the SMTI involves a set $U = \{u_1, \ldots, u_{n_2}\}$ of men, a set $W = \{w_1, \ldots, w_{n_1}\}$ of women, and a set $E \subseteq U \times W$ of acceptable man-woman pairs. Thus, each man $u_i \in U$ has an acceptable set of women $A(u_i) = \{w_j \in W : (u_i, w_j) \in E\}$ and likewise, each woman w_j has an acceptable set of men $A(w_j) = \{u_i \in W : (u_i, w_j) \in E\}$. The agents are the men and women in $U \cup W$. Each agent $a_k \in U \cup W$ has a preference list in which he/she ranks $A(a_k)$. However, agents are allowed to express indifference in their preference lists. We denote $w_j \prec_{u_i} w_{j'}$ when man u_i prefers woman w_j to $w_{j'}$, and we use $w_j \sim_{u_i} w_{j'}$ if u_i is indifferent between two women w_j and $w_{j'}$. Specifically, $\sim_k$ is an equivalence relation (reflexive, symmetric, transitive) and there exists a linear order over the equivalence classes of $A(a_k)$ for each $k \in K$. The special case of the SMTI in which every man-woman pair is acceptable and all the preferences are strict is called the Stable Marriage problem (SM). The SM was first studied by Gale and Shapley (1962).

An assignment M is a subset of E. If $(u_i, w_j) \in E$, then u_i is assigned to w_j and w_j is assigned to u_i. For each $a_k \in U \cup W$, the set of assignees of a_k in M is denoted by $M(a_k)$. If $a_k \in U \cup W$ and $M(a_k) = \emptyset$, a_i is unassigned; otherwise a_k is assigned. A matching M is an assignment such that $|M(a_k)| \leq 1 \; \forall a_k$.

Given a matching M in an instance I of the SMTI, a pair (u_i, w_j) is said to block M, or to be a blocking pair of M, if the following conditions are satisfied:

1. u_i is unassigned or prefers w_j to his assigned woman in M, and
2. w_j is unassigned or prefers u_i to her assigned man in M.

M is said to be weakly stable (or simply stable) if it has no blocking pair. Every instance of the SMTI has a stable matching, and finding one can be done in linear time. Furthermore, all the stable matchings in an instance of the SM have the same size. However, instances of the SMTI can have stable matchings of different sizes. In fact, the problem of finding a maximum cardinality weak stable matching in an instance of the SMTI, called MAX-SMTI, is NP-hard. The NP-hardness holds even in the restricted case where the ties occur in the women's preference lists only, any tie forms the whole list in which it appears, and each tie is of length 2 (Manlove et al. (2002, Theorem 2)). In the following, we reduce the restricted case of the MAX-SMTI aforementioned to our problem to prove that the non envy-free assignment of the CRPP is NP-hard. Define the following decision problem:

Name: Non envy-free assignment of the CRPP
Instance: K customers, I products with the same fixed price, a number of copies of each product, a subset of acceptable products I^k for each customer k, a preference list (with no ties) of products for each customer, integer reservation prices of the customers, an integer $T \in \mathbb{Z}^+$.
Question: Does the given instance admit a feasible non envy-free assignment of products of revenue equal to T?

Proposition 4.2. *The non envy-free assignment of the* CRPP *is NP-complete, even if there is only one copy of each product and the price of every product is fixed to a common price 1.*

Proof. It is easy to see that the non envy-free assignment of the CRPP is in NP. Given a pricing and an assignment of the products, checking for feasibility translates to checking that, for a given customer, all the products he prefers to the one he is assigned either are sold out or have a price above his reservation price. Clearly, this can be done in polynomial time. To show NP-hardness, we reduce the particular version of the MAX-SMTI where ties occur in the women's preference lists only, any tie forms the whole list in which it appears, and each tie is of length 2.

Let I be an instance of the MAX-SMTI, $U = \{u_1, \ldots, u_{n_2}\}$ the set of men, $W = \{w_1, \ldots, w_{n_1}\}$ the set of women, and $E \subseteq U \times W$. Assume that the preference list of every man has no ties, and that every woman w_j has an acceptable set $A(w_j)$ with two men and is indifferent between them. We construct an instance I' of the CRPP as follows: let U be the set of customers, and W be the set of products. The acceptable products for a customer coincide with the acceptable women for the corresponding man and his preference list is also the same. The set of customers interested in a product w_j is defined as the set $A(w_j)$. Assume that there is only one copy of each product w_j. Set the price of every product to 1. Finally, for each customer u_i, set the reservation price of w_j equal to 0 if $w_j \notin A(u_i)$, and equal to 1 otherwise. In this way, every customer can afford every product in his list of preference. We claim that the SMTI instance I has a (weak) stable matching of size T if and only if I' admits a feasible solution of revenue T.

For, suppose that I has a stable matching M of size T. We construct an assignment of products M' to customers assigning product w_j to customer u_i for each $(u_i, w_j) \in M$. Clearly, the revenue of this assignment is equal to the number of customers that purchase, i.e. T. Now suppose, with the aim of contradiction, that the assignment is not feasible. Then there exists a customer u_i that has been assigned product w_j but he prefers product $w_{j'}$. Furthermore, $u_i \in A(w_{j'})$ and $w_{j'}$ is not sold out. But then $(u_i, w_{j'})$ is a blocking pair in M, since u_i prefers $w_{j'}$ to w_j and $w_{j'}$ is unassigned.

Conversely, suppose that M' is a feasible assignment of the products to the customers with revenue T. Then, $M = \{(u_i, w_j) \in E : u_i \text{ purchases product } w_j \text{ in } M'\}$ is a stable matching of size T of I of the SMTI. Otherwise, let $(u_i, w_{j'})$ block M. Then $w_{j'}$ is unassigned (because every men is tied in the list of preferences of $w_{j'}$), and there exists w_j such that $(u_i, w_j) \in E$ and u_i prefers $w_{j'}$ to w_j. But then the assignment M' is not feasible in the CRPP, because u_i is purchasing w_j but he prefers $w_{j'}$, and $w_{j'}$ is not sold out. □

Having seen that the assignment of products is intrinsically different in the envy and envy-free versions of the CRPP, we now provide the formulations and results for the envy version of the problem.

4.2 Three-index mixed integer formulation for the CRPP

We first define the variables required to introduce our three-index formulation for the CRPP named (3ICM). Like in previous chapters, we define binary variable v_i^m, $\forall i \in I$, $\forall m \in M_i$, that takes value 1 if the price of product i is equal to the m-th smallest reservation price b_i^m, and binary variable y_i^{km}, $\forall k \in K$, $i \in I^k$, $m \in M_i^k$, that takes value 1 if customer k purchases product i at price b_i^m. Lastly, we define binary variable u_i^m, $\forall i \in I$, $m \in M_i$, which takes value 1 if that product i sells out at price b_i^m. Formulation (3ICM) is as follows:

$$\text{(3ICM)} \quad \max_{\mathbf{v},\mathbf{y},\mathbf{u}} \sum_{k \in K} \sum_{i \in I^k} \sum_{m \in M_i^k} b_i^m y_i^{km} \tag{4.1a}$$

$$\text{s.t.} \quad \sum_{i \in I^k} \sum_{m \in M_i^k} y_i^{km} \leq 1 \quad \forall k \in K, \tag{4.1b}$$

$$\sum_{m \in M_i} v_i^m \leq 1 \quad \forall i \in I, \tag{4.1c}$$

$$\sum_{k \in K_i} \sum_{m \in M_i^k} y_i^{km} \leq c_i \quad \forall i \in I, \tag{4.1d}$$

$$y_i^{km} \leq v_i^m \quad \forall k \in K, i \in I^k, m \in M_i^k, \tag{4.1e}$$

$$u_i^m \leq v_i^m \quad \forall i \in I, m \in M_i, \tag{4.1f}$$

$$c_i \sum_{m \in M_i} u_i^m \leq \sum_{k \in K_i} \sum_{m \in M_i^k} y_i^{km} \quad \forall i \in I, \tag{4.1g}$$

$$\sum_{m \in M_i} u_i^m + (c_i - 1) \sum_{m \in M_i} v_i^m \geq \sum_{k \in K_i} \sum_{m \in M_i^k} y_i^{km} \quad \forall i \in I, \tag{4.1h}$$

$$\sum_{m \in M_i^k} v_i^m \leq \sum_{m \in M_i^k} u_i^m + \sum_{\substack{j \in I: \\ j \preceq_k i}} \sum_{m \in M_j^k} y_j^{km} \quad \forall k \in K, i \in I^k, \tag{4.1i}$$

$$v_i^m, u_i^m \in \{0,1\} \quad \forall i \in I, m \in M_i, \tag{4.1j}$$

$$y_i^{km} \in \{0,1\} \quad \forall k \in K, i \in I^k, m \in M_i^k. \tag{4.1k}$$

Constraints (4.1b) ensure that customers are unit-demand. Constraints (4.1c) guarantee that each product has at most one price. If $\sum_{m \in M_i} v_i^m = 0$, then the product is not proposed to the customers (or equivalently its price is set arbitrarily high). Constraints (4.1d) are the *capacity constraints*, and they assure the capacity is not exceeded. Constraints (4.1e) guarantee that customer k purchases product i at its right price. Constraints (4.1f)-(4.1h) force the u-variables to take their right value. Constraints (4.1f) establish that a product i can only sell out at price b_i^m (i.e. $u_i^m = 1$) if it has such price. Constraints (4.1g) force u_i^m to be 0 when $\sum_{k \in K_i} \sum_{m \in M_i^k} y_i^{km} < c_i$, and (4.1h) imply $u_i^m = 1$ when $\sum_{k \in K_i} \sum_{m \in M_i^k} y_i^{km} = c_i$. Constraints (4.1i) are the *preference constraints* and guarantee that the preferences are satisfied when the products are not sold out. When $\sum_{m \in M_i^k} v_i^m = 1$, customer k can afford product i. Therefore, either $\sum_{m \in M_i^k} u_i^m = 1$, that is, the product is sold out, or customer k must purchase i or a product he prefers over i, so $\sum_{\substack{j \in I: \\ j \preceq_k i}} \sum_{m \in M_j^k} y_j^{km} = 1$. Finally, the objective function (4.1a) is the revenue of the company.

The set of u-variables used to derive formulation (3ICM) is not essential, in the sense that a formulation can be derived using only the sets of v- and y-variables. However, in the following we introduce several sets of valid inequalities that can strengthen formulation (3ICM), some of which incorporate u-variables, so this is the motivation to include them. First, we give a necessary definition:

Definition 4.3. Let i be a product and c_i the number of copies available. Then we define c_i^m as the minimum between the capacity c_i and the number of customers that can purchase i in a feasible solution at price b_i^m, i.e. $c_i^m := \min\{c_i, |\{k \in K_i : \sigma_i(k) \geq m\}|\}$.

This definition allows for the introduction of a strengthened set of capacity constraints:

Proposition 4.4. *The set of constraints*

$$\sum_{k \in K_i} \sum_{m \in M_i^k} y_i^{km} \leq \sum_{m \in M_i} c_i^m v_i^m \qquad \forall i \in I \tag{4.2}$$

is valid for CRPP *and dominates set* (4.1d).

Proof. If $\sum_{m \in M_i} v_i^m = 0$, then i is not sold and therefore $\sum_{k \in I^k} y_i^{km} = 0$. Otherwise, there exists $v_i^m = 1$ for some $m \in M_i$, and $\sum_{k \in I^k} \sum_{m \in M_i^k} y_i^{km} \leq \min\{c_i, |\{k \in K_i : \sigma_i(k) \geq m\}|\} =: c_i^m$. Therefore, (4.2) are valid.

Let us prove that they dominate (4.1d). The fact that $c_i^m \leq c_i \; \forall m \in M_i$ implies:

$$\sum_{k \in K_i} \sum_{m \in M_i^k} y_i^{km} \leq \sum_{m \in M_i} c_i^m v_i^m \leq \sum_{m \in M_i} c_i v_i^m \leq c_i.$$

□

Since sets (4.1d) and (4.2) have the same number of constraints, from now on we will consider formulation (3ICM) with (4.2) instead of (4.1d). Set (4.2) can be further strengthened:

Proposition 4.5. *The set of constraints*

$$\sum_{\substack{k \in K_i: \\ \sigma_i(k) \geq m}} y_i^{km} \leq c_i^m v_i^m \qquad \forall i \in I, m \in M_i \tag{4.3}$$

is valid for CRPP *and dominates set* (4.2).

Proof. The proof of the validity of set (4.3) follows an analogous reasoning than that of set (4.2). The fact that (4.3) dominate (4.2) is clear, since we can obtain (4.2) from (4.3) summing up each size of the constraints for $m \in M_i$. □

We can also strengthen sets (4.1g) and (4.1i):

Proposition 4.6. *The family of constraints*

$$c_i u_i^m \leq \sum_{\substack{k \in K_i: \\ \sigma_i(k) \geq m}} y_i^{km} \qquad \forall i \in I, m \in M_i \tag{4.4}$$

is valid for CRPP *and dominates family* (4.1g).

Proof. If $u_i^m = 0$ then the inequality holds trivially. Otherwise, $u_i^m = 1$ means that product i is sold out and that it has price b_i^m, so $\sum_{\substack{k \in K_i: \\ \sigma_i(k) \geq m}} y_i^{km} = c_i$. The fact that (4.4) dominate (4.1g) is clear. □

Proposition 4.7. *The set of constraints*

$$v_i^m \leq u_i^m + y_i^{km} + \sum_{\substack{j \in I: \\ j \prec_k i}} \sum_{m \in M_j^k} y_j^{km} \qquad \forall k \in K, i \in I^k, m \in M_i^k \tag{4.5}$$

is valid for CRPP *and dominates constraints* (4.1i).

Proof. If $v_i^m = 0$, then the inequality holds trivially. If $v_i^m = 1$, then product i is sold out (i.e. $u_i^m = 1$) or k purchases i at price b_i^m and $y_i^{km} = 1$, or k purchases a product he prefers to i (at any price) and thus $\sum_{\substack{j \in I: \\ j \prec_k i}} \sum_{m \in M_j^k} y_j^{km} = 1$. The fact that (4.5) dominate (4.1i) is straightforward considering (4.1f). □

Proposition 4.8. *The set of constraints*

$$(c_i - |S|)\, u_i^m \leq \sum_{\substack{k \in K_i \setminus S: \\ \sigma_i(k) \geq m}} y_i^{km} \qquad \forall i \in I, m \in M_i, S \subset K_i : |S| < c_i \tag{4.6}$$

is valid for CRPP. *If we set* $S = \emptyset$*, we obtain constraints* (4.4).

Proof. If $u_i^m = 0$, then the inequality holds trivially. If $u_i^m = 1$, then product i is sold out, so exactly c_i copies are sold among customers from S and from $K_i \setminus S$. In the worst case, every customer $k \in S$ has bought one copy of i, so at least $c_i - |S|$ copies of i must be sold to customers from $K_i \setminus S$. $\square$

MET1 is the first method considered in the computational study presented in Section 4.5, and consists in formulation (3ICM). To assess the performance of the valid inequalities presented, we compare MET1 with MET2, where we solve formulation (3ICM) and include valid inequalities from (4.3), (4.4), (4.5) and (4.6) in a branch-and-cut fashion. Since sets (4.3)-(4.5) include a polynomial number of inequalities, we separate them by complete enumeration.

As for set (4.6), it includes an exponential number of inequalities. To separate them, it is useful to notice that (4.6) come from the linearization of constraints

$$c_i u_i^m \leq \sum_{k \in K_i^m} y_i^{km} u_i^m \qquad \forall i \in I, m \in M_i, k \in K_i^m, \tag{4.7}$$

where $K_i^m := \{k \in K_i : \sigma_i(k) \geq m\}$.

In order to linearize (4.7), we must replace each nonlinear term of the RHS with an upper bound. Since all the variables are binary, it suffices to select one of the variables involved in each term. In this way, for any set $S \subset K_i$, we obtain the linear constraint $c_i u_i^m \leq \sum_{k \in S} u_i^m + \sum_{k \in K_i \setminus S} y_i^{km}$, which is precisely (4.6). From here, the separation procedure is simple. Given a fractional solution $(\bar{v}_i^m, \bar{y}_i^{km}, \bar{u}_i^m)$ of (3ICM), for each i and m we consider set $S := \{k \in K_i : \bar{u}_i^m \leq \bar{y}_i^{km}\}$. The resultant constraint is added if and only if it is violated. Preliminary testing amounts for adding the violated cuts (from the four families) both at the root node and in the nodes of the branching tree of depth less than or equal to 3.

4.3 Two-index mixed integer formulation for the CRPP

In this section, we introduce formulation (2ICM), that involves a much smaller number of variables than (3ICM). As for variables, we use the sets v_i^m and u_i^m, $\forall i \in I$, $m \in M_i$, previously introduced, and we define two more sets. In order to model the customers' purchasing decision, we define binary variable x_i^k, $\forall i \in I$, $k \in K_i$, that takes value 1 if customer k purchases product i. Finally, in order to model a linear objective function, we define z_i^k, $\forall k \in K$, $i \in I^k$, as a continuous variable that takes as value the price of product i if customer k purchases it, and 0 if k does not purchase it. Using these sets of variables, we introduce formulation (2ICM):

$$
\begin{aligned}
\text{(2ICM)} \quad \max_{\mathbf{v},\mathbf{x},\mathbf{z},\mathbf{u}} \quad & \sum_{k\in K}\sum_{i\in I^k} z_i^k && (4.8a)\\
\text{s.t.} \quad & \sum_{i\in I^k} x_i^k \leq 1 \quad \forall k \in K, && (4.8b)\\
& \sum_{m\in M_i} v_i^m \leq 1 \quad \forall i \in I, && (4.8c)\\
& \sum_{k\in K_i} x_i^k \leq \sum_{m\in M_i} c_i^m v_i^m \quad \forall i \in I, && (4.8d)\\
& x_i^k \leq \sum_{m\in M_i^k} v_i^m \quad \forall k \in K, i \in I^k, && (4.8e)\\
& u_i^m \leq v_i^m \quad \forall i \in I, m \in M_i, && (4.8f)\\
& c_i \sum_{m\in M_i} u_i^m \leq \sum_{k\in K_i} x_i^k \quad \forall i \in I, && (4.8g)\\
& \sum_{m\in M_i} u_i^m + (c_i - 1)\sum_{m\in M_i} v_i^m \geq \sum_{k\in K_i} x_i^k \quad \forall i \in I, && (4.8h)\\
& \sum_{m\in M_i^k} v_i^m \leq \sum_{m\in M_i^k} u_i^m + \sum_{\substack{j\in I:\\ j\preceq_k i}} x_j^k \quad \forall k \in K, i \in I^k, && (4.8i)\\
& z_i^k \leq b_i^{\sigma_i(k)} x_i^k \quad \forall k \in K, i \in I^k, && (4.8j)\\
& z_i^k \leq \sum_{m\in M_i^k} b_i^m v_i^m \quad \forall k \in K, i \in I^k, && (4.8k)\\
& v_i^m, u_i^m \in \{0,1\} \quad \forall i \in I, m \in M_i, && (4.8l)\\
& x_i^k \in \{0,1\} \quad \forall k \in K, i \in I^k, && (4.8m)\\
& z_i^k \geq 0 \quad \forall k \in K, i \in I^k. && (4.8n)
\end{aligned}
$$

Constraints (4.8b) guarantee that customers purchase at most one product from the company. Constraints (4.8c) ensure that each product has at most one price. Constraints (4.8d) are the *capacity constraints*, and they assure that no more than c_i copies of product i are sold. Constraints (4.8e) prevent customer k from purchasing product i if he cannot afford it. Constraints (4.8f)-(4.8h) force the u-variables to take their right value: (4.8f) establish that a product i can only sell out at price b_i^m (i.e. $u_i^m = 1$) if it has such price; (4.8g) force u_i^m to be 0 when $\sum_{k\in K_i} x_i^k < c_i$; and (4.8h) imply $u_i^m = 1$ when $\sum_{k\in K_i} x_i^k = c_i$. Constraints (4.8i) are the *preference constraints* and guarantee that the preferences are satisfied when the products are not sold out. Note that when $\sum_{m\in M_i^k} v_i^m = 1$, customer k can afford product i. Therefore, either $\sum_{m\in M_i^k} u_i^m = 1$, that is, the product is sold out, or customer k must purchase i or a product he prefers over i, so $\sum_{\substack{j\in I:\\ j\preceq_k i}} x_j^k = 1$. The last sets of constraints (4.8j)-(4.8k) bound the z-variables to their value. If customer k does not purchase product i, then by (4.8j) we deduce $z_i^k = 0$. On the other hand, if $x_i^k = 1$, then the RHS of (4.8j) is an upper bound on z_i^k, and k is able to afford i, so $v_i^{\bar m} = 1$ for some $\bar m \leq \sigma_i(k)$. But then by (4.8k) we obtain that $z_i^k \leq b_i^{\bar m}$, which is exactly the price of i.

Finally, the objective function (4.8a) is the sum of the revenue of the company obtained for each customer k and product i.

Much like (3ICM), formulation (2ICM) can be strengthened adding valid inequalities. In the following section, we include three sets of valid inequalities developed for (2ICM).

4.4 Valid inequalities for formulation (2ICM) and resolution schemes

In this section, we introduce three families of valid inequalities for formulation (2ICM). In the last subsection, we also explain the separation procedures developed to include them and the branch-and-cut algorithms. We have also included an in-out stabilization procedure to avoid the tailing off effect. In Section 4.5, we test and compare the performance of these resolution schemes.

4.4.1 Projecting out the customer decision variables

Formulation (3ICM) has a larger number of variables and constraints than (2ICM), but the linear relaxation of (3ICM) with the valid inequalities (4.3)-(4.6) gives a stronger upper bound. In this subsection, we discuss how to project out formulation (3ICM) on formulation (2ICM), obtaining a subproblem that can be solved for a given fractional feasible solution of (2ICM) to derive valid inequalities.

First of all, let us extend formulation (2ICM) adding the y-variables used in (3ICM), as well as the necessary constraints to relate them to the rest of the variables. By definition, $x_i^k = \sum_{m\in M_i} y_i^{km}$ and $z_i^k = \sum_{m\in M_i} b_i^m y_i^{km}$ for all $k \in K$, $i \in I^k$. Adding the valid inequalities developed in Section 4.2, we obtain:

$$\text{(2ICM+)} \qquad \max_{\mathbf{v},\mathbf{y},\mathbf{x},\mathbf{z},\mathbf{u}} \quad \sum_{k\in K}\sum_{i\in I^k} z_i^k \tag{4.8a}$$

$$\text{s.t.} \quad (4.8b)-(4.8n), (4.1e), (4.1k), (4.3), (4.4),$$

$$v_i^m \le u_i^m + y_i^{km} + \sum_{\substack{j\in I:\\ j\prec_k i}} x_j^k \quad \forall k \in K, i \in I^k, m \in M_i^k, \tag{4.9a}$$

$$z_i^k \le \sum_{m\in M_i^k} b_i^m y_i^{km} \quad \forall k \in K, i \in I^k, \tag{4.9b}$$

$$x_i^k \ge \sum_{m\in M_i^k} y_i^{km} \quad \forall k \in K, i \in I^k. \tag{4.9c}$$

Constraints (4.8b)-(4.8n) along with the objective function constitute formulation (2ICM). Constraints (4.1e), (4.3), (4.4) and (4.9a) dominate, respectively, constraints (4.8e), (4.8d), (4.8g) and (4.8i). In fact, constraints (4.9a) are (4.5) with x_j^k replacing $\sum_{m\in M\in M_j^k} y_j^{km}$ in the third sum. Finally, constraints (4.9b) and (4.9c) relate the y-variables with the x- and z-variables. Although (4.9b) appear as inequalities, they are satisfied as equalities by any optimal solution of (2ICM+) due to the objective function. As for (4.9c), let us assume that there exists a feasible solution with $x_i^k > \sum_{m \in M_i^k} y_i^{km}$. Then since the variables are binary, it follows $x_i^k = 1$ and $\sum_{m \in M_i^k} y_i^{km} = 0$. But then $\sum_{m \in M_i^k} y_i^{km} = 0 \Rightarrow \sum_{m \in M_i^k} b_i^m y_i^{km} = 0$, so by (4.9b) we obtain $z_i^k = 0$. Therefore k is purchasing product i but the revenue of the company associated to this customer is 0, so the solution is not optimal.

Proposition 4.9. *Consider a fixed product $i \in I$. Then the following family of constraints*

$$\sum_{k \in K_i} z_i^k \alpha^k + \sum_{m \in M_i} c_i u_i^m \lambda^m + \sum_{k \in K_i} \sum_{m \in M_i^k} \left(v_i^m - u_i^m + \sum_{\substack{j \in I: \\ j \prec_k i}} x_i^k \right) \epsilon^{km} \leq \sum_{k \in K_i} x_i^k \beta^k + \sum_{k \in K_i} \sum_{m \in M_i^k} v_i^m \gamma^{km} + \sum_{m \in M_i} c_i^m v_i^m \delta^m \quad (4.10)$$

is valid for (2ICM) *if for α^k, $\beta^k \geq 0$ $\forall k \in K_i$, $\gamma^{km}, \epsilon \geq 0$ $\forall k \in K_i, m \in M_i^k$ and $\delta^m, \lambda^m \geq 0$, $\forall m \in M_i^k$, it holds*

$$b_i^m \alpha^k + \lambda^m + \epsilon^{km} \leq \beta^k + \gamma^{km} + \delta^m \qquad \forall k \in K_i, m \in M_i^k. \quad (4.11)$$

Proof. For a fixed $i \in I$, we associate dual variables α^k, β^k, γ^{km}, δ^k, λ^m and ϵ^{km} to constraints (4.9b), (4.9c), (4.1e), (4.3), (4.4) and (4.9a), respectively.

Then by Farkas' Lemma (see Lemma 1.2 from Chapter 1), we obtain the following result: given a solution $(\boldsymbol{v}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u})$, there exist variables y satisfying the above sets of constraints if and only if

$$\sum_{k \in K_i} z_i^k \alpha^k + \sum_{m \in M_i} c_i u_i^m \lambda^m + \sum_{k \in K_i} \sum_{m \in M_i^k} \left(v_i^m - u_i^m + \sum_{\substack{j \in I: \\ j \prec_k i}} x_j^k \right) \epsilon^{km} \leq \sum_{k \in K_i} x_i^k \beta^k + \sum_{k \in K_i} \sum_{m \in M_i^k} v_i^m \gamma^{km} + \sum_{m \in M_i} c_i v_i^m \delta^m$$

for all $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\lambda} \geq 0$, $\forall k \in K_i$, $m \in M_i^k$ satisfying

$$b_i^m \alpha^k + \lambda^m + \epsilon^{km} \leq \beta^k + \gamma^{km} + \delta^m.$$

□

As for constraints (4.6), they cannot be projected out because the set includes an exponential number. However, an alternative consists in developing an equivalent set of valid inequalities for (2ICM). Thus, we can linearize constraints

$$c_i u_i^m \leq \sum_{k \in K_i^m} x_i^k v_i^m u_i^m \qquad \forall i \in I, m \in M_i, k \in K_i^m, \quad (4.12)$$

with $K_i^m := \{k \in K_i : \sigma_i(k) \geq m\}$, in a similar manner to family (4.7), obtaining

$$c_i u_i^m \leq \sum_{k \in S_1} u_i^m + \sum_{k \in S_2} v_i^m + \sum_{\substack{k \in K_i \setminus \\ (S_1 \cup S_2)}} x_i^k, \qquad \forall i \in I, m \in M_i, S_1, S_2 \subset K_i : S_1 \cap S_2 = \emptyset. \quad (4.13)$$

Family (4.13) can be reduced noticing that the subfamily where $S_2 = \emptyset$ dominates (4.13) due to (4.8f). Nonetheless, preliminary testing showed that the inclusion of valid inequalities from this family does not improve the linear relaxation bound from (2ICM), so (4.13) is not included in the computational experiments of Section 4.5.

4.4.2 Separation of valid inequalities from family (4.10)

In what follows, we are going to analyze the problem that results from fixing $\lambda = \delta = \epsilon = 0$ in (4.10). The reason is that the inequalities associated to variables α, β and γ can be separated by customer (as well as by product). Next, we study the particular case $\alpha = 1$. Thus, in order to obtain the strongest bound we look for values of α and β which provide the smaller value of the RHS of (4.10). Considering a fixed product i and customer k, our problem (SP1) can be stated as:

$$\text{(SP1)} \qquad \min_{\beta,\gamma} \quad x_i^k \beta + \sum_{m \in M_i^k} v_i^m \gamma^m \tag{4.14a}$$

$$\text{s.t.} \quad \beta + \gamma^m \geq b_i^m \quad \forall m \in M_i^k, \tag{4.14b}$$

$$\beta, \gamma \geq 0. \tag{4.14c}$$

For a given β, constraints (4.14b) can be expressed as $\gamma^m \geq b_i^m - \beta$ for each m. Hence, $\gamma^m = \max\{0, b_i^m - \beta\}$. Let us see that the best value for β belongs to the set of candidate reservation prices $\{b_i^1, \dots, b_i^{\sigma_i(k)}\}$. Suppose that $b_i^r < \beta < b_i^{r+1}$. Then $\gamma^m = 0$ for $m \leq r$, and $\gamma^m = b_i^r - \beta$ for $m > r$. Substituting γ in the objective function of (SP1), we obtain:

$$x_i^k \beta + \sum_{\substack{m \in M_i^k: \\ m > r}} \left(b_i^r - \beta\right) v_i^m = \left(x_i^k - \sum_{\substack{m \in M_i^k: \\ m > r}} v_i^m \right) \beta + \sum_{\substack{m \in M_i^k: \\ m > r}} b_i^r v_i^m. \tag{4.15}$$

The above linear function in β attains its minimum value in one of its extreme values, that is, $\beta = b_i^r$ or $\beta = b_i^{r+1}$, depending on the sign of the slope $x_i^k - \sum_{\substack{m \in M_i^k: \\ m > r}} v_i^m$. This leads to the following proposition:

Proposition 4.10. *The family of inequalities*

$$z_i^k \leq b_i^{r_i^k} x_i^k + \sum_{m = r_i^k + 1}^{\sigma_i(k)} \left(b_i^m - b_i^{r_i^k}\right) v_i^m, \qquad \forall k \in K, i \in I^k, r_i^k \in \{0, \dots, \sigma_i(k)\} \tag{4.16}$$

is valid for (2ICM).

Family (4.16) is included in a family of valid inequalities of exponential size originally developed in Chapter 2 for the RPP (we include it here because we have replaced the budgets with reservation prices):

$$z_i^k \leq b_i^{r_i^k} x_i^k + \sum_{m = r_i^k + 1}^{\sigma_i(k)} \left(b_i^m - b_i^{r_i^k}\right) v_i^m + \sum_{m \in Q_i^k} \left(b_i^m - b_i^{r_i^k}\right) \left(x_i^k + v_i^m - 1\right), \tag{4.17}$$

$\forall k \in K, i \in I^k$, any integer $r_i^k \in \{0, \dots, \sigma_i(k)\}$ and any subset $Q_i^k \subseteq \{1, \dots, r_i^k - 1\}$. In the following, we prove that family (4.16) dominates (4.17).

Proposition 4.11. *Given $k \in K$, $i \in I^k$, the strongest valid inequality from family* (4.17) *is obtained when $Q_i^k = \emptyset$.*

Proof. Let us consider $k \in K$, $i \in I^k$ fixed and, for the sake of notation, $r := r_i^k$ and $Q := Q_i^k$. First of all, let $q := \min_Q\{m\}$ (and therefore, $q < r$ and $b_i^q < b_i^r$). Then the following inequality

$$z_i^k \leq b_i^r x_i^k + \sum_{m=q}^{\sigma_i(k)} \left(b_i^m - b_i^r\right) v_i^m + \left(b_i^q - b_i^r\right)\left(x_i^k - 1\right) \tag{4.18}$$

is valid and stronger than the corresponding inequality of (4.17), for all r and Q.

Let us prove its validity first. If $x_i^k = 1$, then $v_i^m = 1$ for some $\bar{m} \leq \sigma_i(k)$, and the inequality becomes $z_i^k \leq b_i^r + (b_i^{\bar{m}} - b_i^r) = b_i^{\bar{m}}$, valid because the price of product i is exactly $b_i^{\bar{m}}$. On the other hand, if $x_i^k = 0$, then the inequality becomes $z_i^k \leq \sum_{m=q}^{\sigma_i(k)} \left(b_i^m - b_i^r\right) v_i^m + b^r - b^q$. If $v_i^{m_0} = 1$ for $m_0 \geq r$, then the first sum of the RHS is non negative. If $v_i^{m_0} = 1$ for $m_0 < r$, then the inequality is $z_i^k \leq b_i^{m_0} - b_i^r + b_i^r - b_i^q = b_i^{m_0} - b_i^q$, non negative because of the definition of q. In both cases, since $z_i^k = 0$ and the RHS of the inequality is greater than or equal to zero, the inequality holds.

Now, to prove that it is stronger, let us subtract the RHS of (4.18) from the RHS of (4.17) to see that the difference is non negative:

$$\begin{aligned}
&\left(b_i^r x_i^k + \sum_{m=r+1}^{\sigma_i(k)} \left(b_i^m - b_i^r\right) v_i^m + \sum_{m \in Q} \left(b_i^m - b_i^r\right)\left(x_i^k + v_i^m - 1\right)\right) - \\
&\left(b_i^r x_i^k + \sum_{m=q}^{\sigma_i(k)} \left(b_i^m - b_i^r\right) v_i^m + \left(b_i^q - b_i^r\right)\left(x_i^k - 1\right)\right) = \\
&= \sum_{m \in \{q,\ldots,r-1\} \setminus Q} \left(b_i^m - b_i^r\right) v_i^m - \max_{m \in Q \setminus \{q\}} \left\{b_i^m - b_i^r\right\}\left(x_i^k - 1\right) \geq 0.
\end{aligned}$$

In the following, let us consider $W(r)$ as the RHS of (4.18) with q fixed to study its variation when r increases:

$$\begin{aligned}
W(r+1) - W(r) &= \left(b_i^{r+1} - b_i^r\right) x_i^k + \sum_{m=q}^{\sigma_i(k)} \left(b_i^{r+1} - b_i^r\right) v_i^m + \left(b^- r_i - b_i^{r+1}\right)\left(x_i^k - 1\right) = \\
&= \left(b_i^{r+1} - b_i^r\right)\left(1 - \sum_{m=q}^{\sigma_i(k)} v_i^m\right) \geq 0.
\end{aligned} \tag{4.19}$$

Supposing q is fixed yields a non negative difference whenever we increment r, so the optimal is to select the smallest possible r. Given that $q \leq r$, we will hence choose $r = q$. This implies that $Q = \emptyset$, and we have proved the statement. □

4.4.3 An additional set of valid inequalities

In this subsection, we present another set of valid inequalities developed for model (2ICM). Unlike set (4.16), this set includes one inequality per product, and in this case the capacity is used to strengthen the inequalities.

Proposition 4.12. *Consider the set of inequalities*

$$\sum_{k\in K_i} z_i^k + \sum_{m\in M_i}\sum_{m'=m+1}^{|M_i|}\left(b_i^{m'} - b_i^{m'-1}\right) U_i^{mm'} u_i^m \leq$$
$$\sum_{k\in K_i} b_i^{r^k} x_i^k + \sum_{m\in M_i}\sum_{m'=1}^{m}\left(b_i^{m'} - b_i^{m'-1}\right) V_i^{mm'} v_i^m, \tag{4.20}$$

$\forall i \in I, r^k \in \{0,\dots,\sigma_i(k)\}$ $\forall k \in K_i$, *where* $U_i^{mm'} := \max\{0, c_i - |\{k \in K_i : \sigma_i(k) \geq m, r^k < m'\}|\}$ *and* $V_i^{mm'} := \min\left\{c_i^m, |\{k \in K_i : \sigma_i(k) \geq m, r^k < m'|\}\right\}$, $\forall i \in I$, $m \in M_i$, $m' \in M_i$. *Set* (4.20) *is valid for* (2ICM).

Proof. Let $i \in I$ be a fixed product and let $r^k \in \{0,\dots,\sigma_i(k)\}$ be fixed reservation prices indices $\forall k \in K_i$. Let $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{u}})$ be a feasible (integer) solution of formulation (2ICM). If $\sum_{m\in M_i} \bar{v}_i^m = 0$, then the corresponding inequality holds trivially because all the variables have value zero. Therefore, we assume $\bar{v}_i^{\bar{m}} = 1$ for some $\bar{m} \in M_i$. This yields $\bar{u}_i^m = 0$ for $m \neq \bar{m}$, and $\bar{u}_i^{\bar{m}} \in \{0,1\}$.

First of all, let $\bar{K}_i := \{k \in K_i : \bar{x}_i^k = 1\}$ be the set of customers purchasing i. Then the following inequality holds:

$$\sum_{k\in K_i} \bar{z}_i^k + \sum_{m\in M_i}\sum_{\substack{k\in\bar{K}_i:\\ r^k>m}}\left(b_i^{r^k} - b_i^m\right)\bar{u}_i^m \leq \sum_{k\in K_i} b_i^{r^k}\bar{x}_i^k + \sum_{m\in M_i}\sum_{\substack{k\in\bar{K}_i:\\ r^k<m}}\left(b_i^m - b_i^{r^k}\right)\bar{v}_i^m. \tag{4.21}$$

Indeed, if $\bar{u}_i^{\bar{m}} = 1$ the LHS of (4.21) is equal to $\sum_{k\in\bar{K}_i} b_i^{\bar{m}} + \sum_{k\in\bar{K}_i:\ r^k>\bar{m}}\left(b_i^{r^k} - b_i^{\bar{m}}\right)$, which is equal to its RHS, $\sum_{k\in\bar{K}_i} b_i^{r^k} + \sum_{k\in\bar{K}_i:\ r^k<\bar{m}}\left(b_i^{\bar{m}} - b_i^{r^k}\right)$. Otherwise, the LHS is $\sum_{k\in\bar{K}_i} b_i^{\bar{m}}$ and thus it also holds. Now, in order to see that the statement holds, we shall prove that the LHS (resp. RHS) of (4.20) is smaller than or equal to (resp. greater than or equal to) the LHS (resp. the RHS) of (4.21). Given that $\bar{u}_i^m = \bar{v}_i^m = 0$ $\forall m \neq \bar{m}$ and $\bar{v}_i^{\bar{m}} = 1$, this translates to proving:

(1) $$\sum_{\substack{k\in\bar{K}_i:\\ r^k<\bar{m}}}\left(b_i^{\bar{m}} - b_i^{r^k}\right) \leq \sum_{m'=1}^{\bar{m}}\left(b_i^{m'} - b_i^{m'-1}\right) V_i^{\bar{m}m'},$$

(2) $$\sum_{\substack{k\in\bar{K}_i:\\ r^k>\bar{m}}}\left(b_i^{r^k} - b_i^{\bar{m}}\right) \geq \sum_{m'=\bar{m}+1}^{|M_i|}\left(b_i^{m'} - b_i^{m'-1}\right) U_i^{\bar{m}m'} \quad \text{if} \quad \bar{u}_i^{\bar{m}} = 1.$$

To prove (1), we have

$$\sum_{\substack{k\in\bar{K}_i:\\ r^k<\bar{m}}}\left(b_i^{\bar{m}} - b_i^{r^k}\right) = \sum_{\substack{k\in\bar{K}_i:\\ r^k<\bar{m}}}\sum_{m'=r^k+1}^{\bar{m}}\left(b_i^{m'} - b_i^{m'-1}\right) = \sum_{m'=1}^{\bar{m}}\left(b_i^{m'} - b_i^{m'-1}\right)\sum_{\substack{k\in\bar{K}_i:\\ r^k<m'}} 1$$
$$= \sum_{m'=1}^{\bar{m}}\left(b_i^{m'} - b_i^{m'-1}\right)|\{k \in \bar{K}_i : r^k < m'\}|,$$

where the second equality is obtained switching the sums in m' and m. Thus, the fact that $|\bar{K}_i| \leq c_i^{\bar{m}}$ and that for all $k \in \bar{K}_i$ it holds $\sigma_i(k) \geq \bar{m}$ implies $|\{k \in \bar{K}_i : r^k < m'\}| \leq \min\left\{c_i^{\bar{m}}, |k \in K_i : \sigma_i(k) \leq \bar{m}, r^k < m'|\right\} =: V_i^{\bar{m}m'}$, and (1) holds.

To prove (2), we follow an analogous procedure:

$$\sum_{\substack{k\in\bar{K}_i:\\ r^k>\bar{m}}} \left(b_i^{r^k} - b_i^{\bar{m}}\right) = \sum_{\substack{k\in\bar{K}_i:\\ r^k>\bar{m}}} \sum_{m'=\bar{m}+1}^{r^k} \left(b_i^{m'} - b_i^{m'-1}\right) = \sum_{m'=\bar{m}+1}^{|M_i|} \left(b_i^{m'} - b_i^{m'-1}\right) \sum_{\substack{k\in\bar{K}_i:\\ r^k\geq m'}} 1$$

$$= \sum_{m'=\bar{m}+1}^{|M_i|} \left(b_i^{m'} - b_i^{m'-1}\right) |\{k \in \bar{K}_i : r^k \geq m'\}|.$$

The result follows because $|\{k \in \bar{K}_i : r^k \geq m'\}| = |\bar{K}_i| - |\{k \in \bar{K}_i : r^k < m'\}| = c_i - |\{k \in \bar{K}_i : \sigma_i(k) \geq \bar{m}, r^k < m'\}| \geq c_i - |\{k \in K_i : \sigma_i(k) \geq \bar{m}, r^k < m'\}|$, and therefore $|\{k \in \bar{K}_i : r^k \geq m'\}| \geq U_i^{\bar{m}m'}$. □

Method MET3 consists in solving an instance of the CRPP using formulation (2ICM). We have developed four additional methods based on (2ICM), that are fully detailed in the following subsection.

4.4.4 Separation algorithms and resolution schemes using formulation (2ICM) and an in-out stabilization method

We begin this subsection with the introduction of the three separation procedures designed to include valid inequalities (4.10), (4.16) and (4.20) dynamically into formulation (2ICM). Then, we show the four resolution methods developed and the in-out stabilization method proposed. The first three methods (MET4-MET6) correspond to a branch-and-cut involving valid inequalities from each of the three families presented, namely (4.10), (4.16) and (4.20). The last one (MET7) is a branch-and-cut including a combination of valid inequalities from families (4.16) and (4.20). These approaches are compared in Section 4.5 using extensive computational experiments.

The first method MET4 incorporates violated inequalities from (4.10) to formulation (2ICM). In order to obtain the dual variables for the inequalities in (4.10), we solve the following separation problem (SP_i) (by means of a commercial solver) for each product $i \in I$:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\boldsymbol{\lambda}} \quad \sum_{k\in K_i} z_i^k \alpha^k + \sum_{m\in M_i} c_i u_i^m \lambda^m - \sum_{k\in K_i} x_i^k \beta^k - \sum_{k\in K_i} \sum_{m\in M_i^k} v_i^m \gamma^{km} - \sum_{m\in M_i} c_i v_i^m \delta^m \tag{4.22a}$$

$$\text{s.t.} \quad b_i^m \alpha^k + \lambda^m \leq \beta^k + \gamma^{km} + \delta^m \qquad \forall k \in K_i, m \in M_i^k, \tag{4.22b}$$

$$\alpha^k, \beta^k \geq 0 \qquad \forall k \in K_i, \tag{4.22c}$$

$$\gamma^{km} \geq 0 \qquad \forall k \in K_i, m \in M_i^k, \tag{4.22d}$$

$$\delta^m, \lambda^m \geq 0 \qquad \forall m \in M_i. \tag{4.22e}$$

Cuts from (4.10) are included after the linear relaxation of (2ICM), until no more valid inequalities are violated or the linear relaxation bound does not decrease, and in the nodes of the branching tree. Preliminary testing led us to include valid inequalities during the branch-and-bound in the nodes of depth less than or equal to 3.

Method MET5 requires a separation procedure to include inequalities from (4.16) dynamically in a branch-and-cut framework. Thus, if we are given a fractional feasible solution of formulation (2ICM), we calculate the r that minimizes the RHS of (4.16) for each k and i fixed simplifying the reasoning developed in Chapter 2 for (4.17). In essence, we study the variation of the RHS of (4.16) as r increases. Calling $W(r)$ the RHS of (4.16) for a given r, we have

$$W(r+1) - W(r) = \left(b^{r+1} - b^{r}\right)\left(\overline{x}_i^k - \sum_{m=r+1}^{\sigma_i(k)} \overline{v}_i^m\right). \qquad (4.23)$$

Clearly $b^{r+1} - b^r > 0$, and the sum $\sum_{m=r+1}^{\sigma_i(k)} \overline{v}_i^m$ has less terms as r increases. Therefore, the slope $\overline{x}_i^k - \sum_{m=r+1}^{\sigma_i(k)} \overline{v}_i^m$ is negative and for some r, the slope changes to a positive value. So the minimum in (4.23) is obtained for the r such that $x_i^k - \sum_{m=r+1}^{\sigma_i(k)} v_i^m \leq 0$ and $x_i^k - \sum_{m=r+2}^{\sigma_i(k)} v_i^m > 0$. Algorithm 3 depicts the separation used in MET5 to incorporate inequalities from set (4.16) in a branch-and-cut fashion.

Algorithm 3 MET5: Separation of inequalities (4.16)

Let $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{u}})$ be a fractional solution obtained after the linear relaxation of formulation (2ICM) or in a node of the branching tree of depth smaller than or equal to three.
For every product $i \in I$ and for every customer $k \in K_i$ do

Step 1. Set $r = 0$.

Step 2. If $r < \sigma_i(k)$ and $\overline{x}_i^k - \sum_{m=r+1}^{\sigma_i(k)} \overline{v}_i^m \leq 0$, update $r := r+1$ and repeat Step 2. Otherwise, go to Step 3.

Step 3. Incorporate constraint

$$z_i^k \leq b_i^r x_i^k + \sum_{m=r+1}^{\sigma_i(k)} \left(b_i^m - b_i^r\right) v_i^m$$

to formulation (2ICM) if and only if it is violated by solution $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{u}})$.

MET6 consists in solving (2ICM) including valid inequalities from set (4.20) in a branch-and-cut. It also requires a separation algorithm for set (4.20), but this algorithm is very similar to Algorithm 3. In this case, the calculation of r^k $\forall k \in K_i$ is analogous to that depicted in Algorithm 3. But here, at each iteration in a node of the branching tree we include one inequality per product (if violated). Like in MET3, inequalities from (4.20) are also included when fractional solutions are found in the nodes of the branching tree of depth smaller than or equal to three.

When valid inequalities from set (4.16) are separated, we include one inequality per customer and product at each iteration. However, in the case of inequalities from family (4.20), only one inequality per product is included at each step, and each of them combines variables associated to all the customers. Therefore, valid inequalities from (4.20) have a much larger number of non zero elements than those from (4.16). Moreover, and

Name	Description
MET1	(3ICM)
MET2	(3ICM) + branch-and-cut with valid inequalities from sets (4.3), (4.4), (4.5) and (4.6)
MET3	(2ICM)
MET4	(2ICM) + branch-and-cut with valid inequalities from set (4.10)
MET5	(2ICM) + branch-and-cut with valid inequalities from set (4.16)
MET6	(2ICM) + branch-and-cut with valid inequalities from set (4.20)
MET7	(2ICM) + branch-and-cut with valid inequalities from sets (4.16) and (4.20)

Table 4.1: Details of methods MET1-MET7 tested in the computational study

according to preliminary testing, the linear relaxation bound of model (2ICM) improves slowly when adding inequalities from (4.20), so a great number of iterations is required in this case. These are the reasons for incorporating an in-out algorithm to stabilize the inclusion of inequalities in the root node in the last two methods MET6 and MET7. This algorithm has been developed by Ben-Ameur and Neto (2007), used for other capacitated problems (e.g. in Fischetti et al. (2016)) and it is widely used to include Benders decomposition cuts, like in Chapter 3. It is of particular importance when only a few valid inequalities are separated at each iteration, like in this case. It allows to generate less cuts of better quality, hopefully reducing the tailing-off effect. For a detailed explanation on the in-out procedure, we refer the reader to subsection 3.4.2. As for the initial interior point, it is frequently obtained using the barrier algorithm with crossover. In our case, we built a non-degenerate convex combination of enough linearly independent points of the polytope and then obtained the centroid.

Finally, in MET7 we separate inequalities from families (4.16) and (4.20). Through preliminary computational experiments, we observed that the upper bound of the linear relaxation obtained separating inequalities from (4.20) was tighter than that obtained with valid inequalities from (4.16). However, inequalities from (4.16) require less iterations to reduce the bound. To combine them in the cut loop previous to the branching, we begin including only constraints from (4.16) until no more are found. Then, we include only inequalities from (4.20) and further reduce the bound. The separation oracles are the ones previously described, and the in-out algorithm is included in this step of MET7. We also separate inequalities from both sets in all the fractional solutions obtained in the nodes of the branching tree of depth less than or equal to three (including all the violated ones from both sets at every node).

We have depicted the seven resolution methods developed and their characteristics in Table 4.1, to help to identify them easily.

4.5 Computational study

In this section, we compare the performance of formulations (2ICM) and (3ICM) and the valid inequalities proposed testing the methods MET1-MET7. The computational experiments were carried out on a personal computer with Intel Xeon E3-1270, 3.40 GHz with 16 GB of RAM. The optimization problems were solved exactly by using the solver Xpress Optimizer Version: 29.01.10 and the methods were coded using Mosel Version: 4.0.3.

To test the algorithms, we created instances with $|K| = 50$, $|K| = 75$, $|K| = 100$, $|K| = 125$ and $|K| = 150$ customers. For each size of set K, we designed instances with $|I| = 5$, $|I| = 10$ and $|I| = 15$ products. For the instances with $|I| = 15$, the number of acceptable products for each customer is equal to 3, whereas for instances with $|I| = 10$ and $|I| = 15$ products, the number ascends to 5 acceptable products. In all instances, the number of copies of each product available (called C in the following) is the same for all the products in I. So for each combination of the previous parameters, we considered four different sizes for C. The reservation prices of the customers were randomly generated between 1 and $4|K|$, and their ranked lists of preferences were also randomly generated. We generated 5 instances of each combination of parameters, 300 in total. All the instances can be found at https://github.com/cdomsa/CRPP/. The time limit was set to 3600 seconds, and the default setting of Xpress was used.

We first compare the performance of MET1-MET7 using the instances of smaller size, namely those with $|K| = 50$, $|K| = 75$ and $|K| = 100$. The most significant information obtained is summarized by means of several figures.

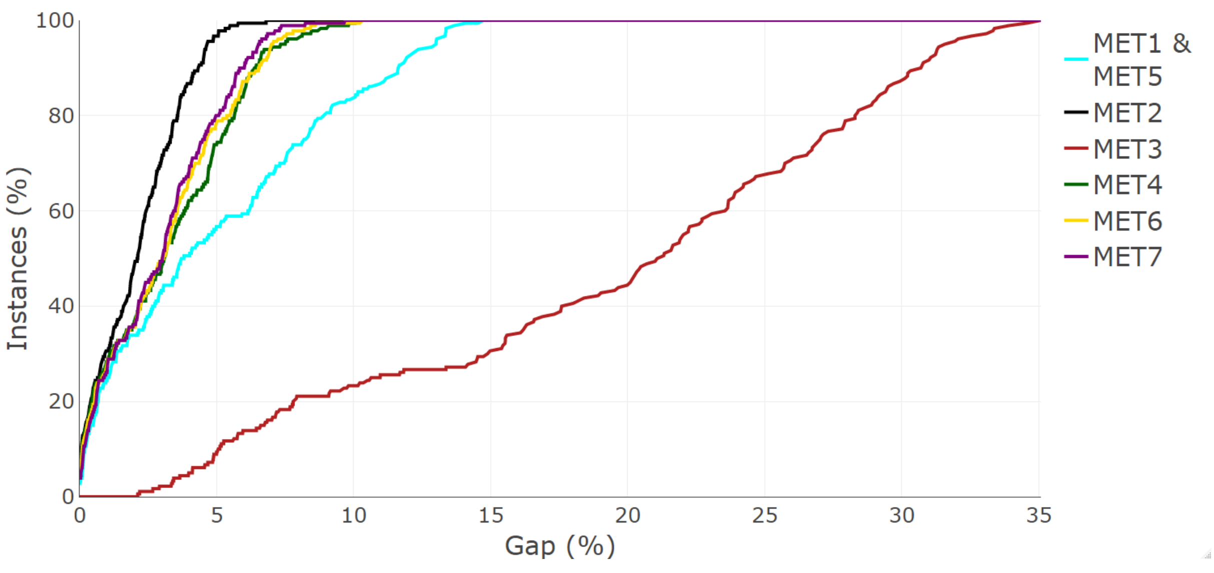

Figure 4.2: In the y-axis, the percentage of instances with an integrality gap less than or equal to that of the x-value is represented for MET1-MET7

First, we compare the integrality gaps obtained. Figure 4.2 is a performance profile that shows the percentage of instances having an integrality gap less than or equal to that on the x-axis. For MET1 and MET3, the gap depicted is the linear relaxation gap LRGap $= 100\frac{\text{UB-BV}}{\text{BV}}\%$ where UB is the upper bound given by the linear relaxation and BV is the best objective value found by any of the models (the optimal value almost always). For MET2, MET4-MET7, the figure shows the integrality gap obtained after adding the violated cuts in the linear relaxation, before branching: RGap $= 100\frac{\text{UBC-BV}}{\text{BV}}\%$, where UBC corresponds to the upper bound obtained after adding the violated cuts in the root node.

As expected, MET3 yields the worst linear relaxation gap of the seven, with gaps of up to 35%. MET1 (formulation (3ICM)) has the second worst linear relaxation bound, and this bound is the same for MET5. This is consistent with the results obtained in Section 4.4. Indeed, valid inequalities (4.16) are obtained projecting formulation (3ICM) on formulation (2ICM), because the parameters associated to valid inequalities (4.3), (4.4) and (4.9a) are set to 0 in the separation problem. As for MET4, MET6 and MET7, they provide similar integrality gaps, MET7 slightly outperforming the others. These methods yield gaps smaller than 10% for all the instances proposed. It is remarkable that the gap obtained adding valid inequalities to (2ICM) can outperform that of (3ICM), as it happens in four out of the seven methods developed. Finally, the best scheme in terms of gap is MET2. This is also consistent with our theoretical results, since this method includes a branch-and-cut with all the valid inequalities developed for (3ICM). The gaps provided by MET2 are always smaller than 7%.

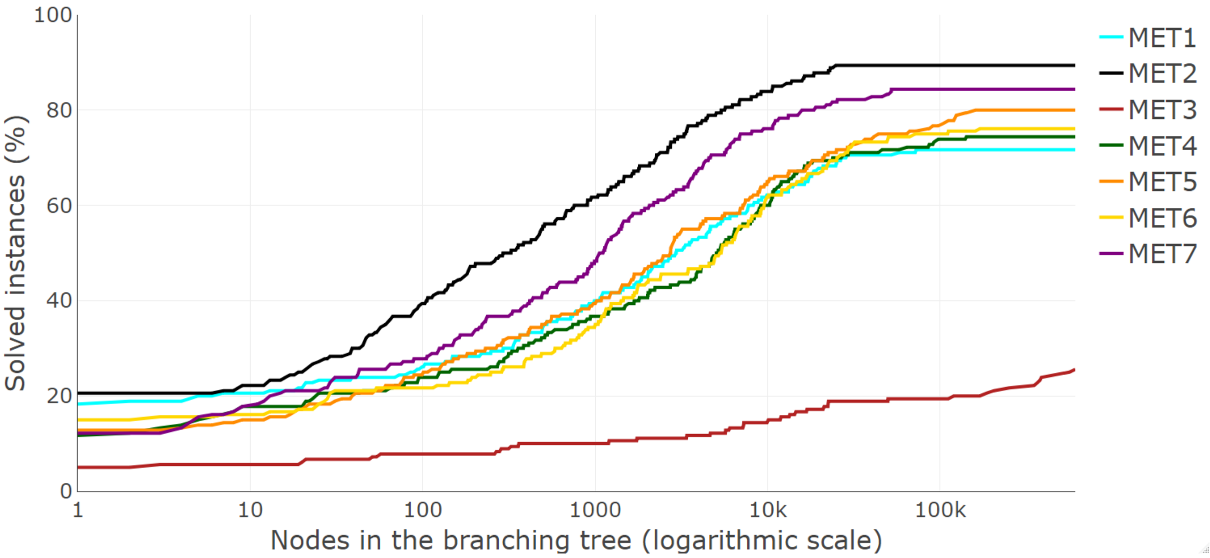

Figure 4.3: In the y-axis, the percentage of instances solved exploring an amount of nodes in the branching tree less than or equal to that of the x-value is represented for MET1-MET7

Figure 4.3 compares MET1-MET7 in terms of the number of nodes explored in the branching tree with respect to the percentage of instances solved to optimality. Once again, formulation (2ICM) (MET3) is clearly the worst in performance, it only solves about a 25% of the instances proposed. Methods MET4-MET6 perform similarly to MET1 (formulation (3ICM)), whereas MET7 performs a bit better than the rest of the approaches, solving up to an 81% of the instances. Finally, MET2 solves the greatest amount of instances and it does so with the least amount of nodes explored.

Finally, Figure 4.4 shows the number of instances solved with respect to the time (in

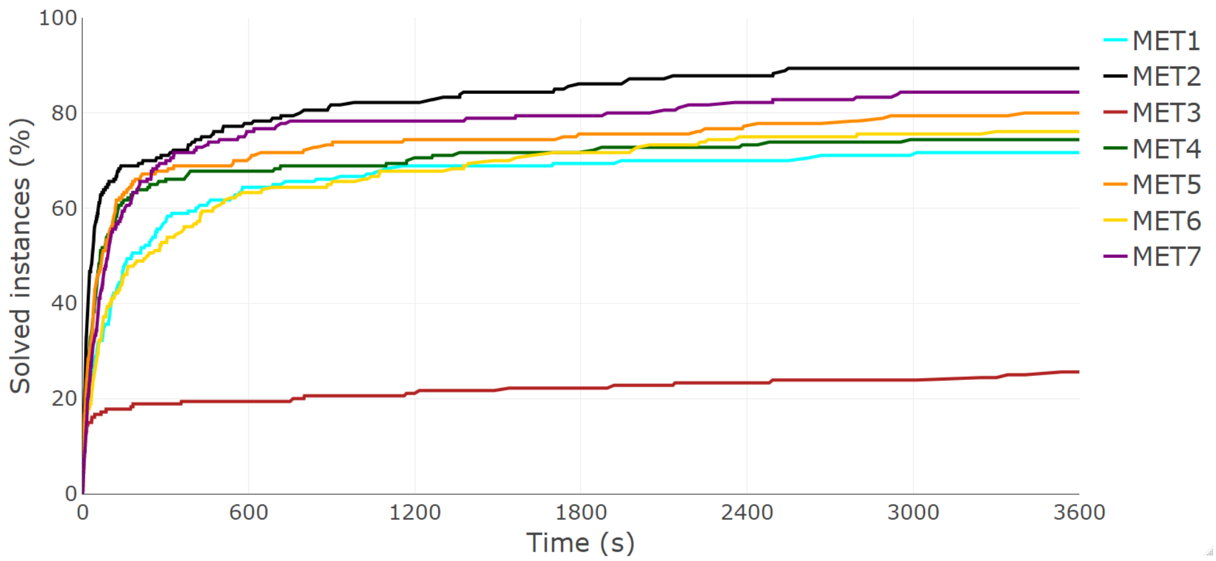

Figure 4.4: Percentage of instances solved with respect to the time (with a time limit of 3600 seconds) by MET1-MET7

seconds), up to 3600 seconds of time limit. This figure confirms the results already seen in the two previous figures, where MET2 and MET3 are the best and worst (respectively) in terms of solved instances. The addition of inequalities to formulation (2ICM) triples the number of instances that are solved to optimality, regardless of the method. Here, the difference in performance between MET4-MET7 is more pronounced than in the previous figures. Thus, we can see that valid inequalities (4.16) alone (in MET5) perform slightly better than (4.20) (in MET6), even though the upper bound provided by MET6 is smaller. Moreover, the combination of both types of valid inequalities in the branch-and-cut makes MET7 outperform the two previous approaches. MET4, the branch-and-cut obtained solving the Farkas separation problem (4.10), provides the worst results of the group.

In view of the results obtained, we decided to run the largest instances (those with $|K| = 125$ and $|K| = 150$) with the three best methods developed: MET2, MET5 and MET7. Table 4.2 summarizes the main results obtained. It shows the integrality gap of the linear relaxation (LRGap, the same for MET5 and MET7), the integrality gap of the linear relaxation after the cuts in the root node (RGap), the integrality gap after 3600 seconds (FGap), the average time in seconds needed to optimally solve the instances (t(s)) and the number of instances solved to optimality in less than the time limit of 3600 seconds. Note that RGap for MET5 is the same as LRGap for MET2.

We can see that the relationship between the number of customers, products and copies of each product determines the number of instances that can be solved to optimality within an hour. In some cases, the three methods can solve all the instances, whereas in other cases none of the instances is solved. Moreover, each method performs better than the rest for certain combinations of parameters. For instance, for a small number of products ($|I| = 5$), MET5 yields the best results in terms of time, even if all the instances are also solved by MET7 within the time limit. However, for $|I| = 15$, MET5 is worse than MET7 in terms of time, final bound and number of instances solved. Besides, for greater values of C, valid inequalities (4.20) added in MET7 reduce the gap RGap significantly. For instance, for $|K| = 150$, $|I| = 15$, $C = 30$, RGap for MET5 is equal to 14.6, whereas RGap for MET7 is equal to 4.2. As for MET2, it solves less instances than MET5 and MET7. Nonetheless, it provides better final bounds for the combinations of parameters that make the instances hard to solve in one hour. All in all, the inclusion of valid inequalities to models (2ICM) and (3ICM) in a branch-and-cut fashion highly improves their performance. Valid inequalities (4.16) are essential to reduce the bound of (2ICM) regardless of the instance, whereas (4.20) are particularly useful for instances with a greater number of products and/or copies of each product.

$\|K\|$	$\|I\|$	C	**MET2**					**MET5**					**MET7**			
			LRGap	RGap	FGap	t(s)	Sol.	LRGap	RGap	FGap	t(s)	Sol.	RGap	FGap	t(s)	Sol.
125	5	5	0.3	0.2	0.0	10	5	5.9	0.3	**0.0**	**4**	**5**	0.3	0.0	12	5
125	5	13	2.4	2.1	0.0	254	5	17.8	2.4	**0.0**	**41**	**5**	2.3	0.0	65	5
125	5	20	4.0	3.6	0.0	1479	5	29.8	4.0	**0.0**	**81**	**5**	3.9	0.0	196	5
125	5	25	6.2	4.5	1.0	3271	2	34.7	6.2	**0.0**	**206**	**5**	5.6	0.0	520	5
125	10	5	0.7	0.6	0.0	78	5	8.8	0.7	**0.0**	**53**	**5**	0.7	0.0	84	5
125	10	13	4.4	3.5	3.0	3600	0	23.3	4.4	**1.8**	**3529**	**1**	4.0	2.9	3600	0
125	10	20	11.3	5.6	**5.0**	**3600**	**0**	31.6	11.3	7.4	3600	0	7.8	6.9	3600	0
125	10	25	15.2	6.7	**6.3**	**3600**	**0**	36.6	15.2	8.9	3600	0	9.0	7.5	3600	0
125	15	5	1.1	0.8	0.0	266	5	11.1	1.1	0.0	231	5	1.0	**0.0**	**219**	**5**
125	15	13	8.3	4.0	**3.0**	**3101**	**1**	26.0	8.3	4.9	3600	0	5.8	4.5	3600	0
125	15	20	14.2	4.7	**3.7**	**3600**	**0**	32.6	14.2	7.7	3600	0	6.4	4.6	3600	0
125	15	25	15.0	4.2	2.2	3377	1	33.7	15.0	7.5	3600	0	4.9	**1.1**	**2763**	**2**
150	5	6	0.2	0.1	0.0	21	5	6.5	0.2	**0.0**	**4**	**5**	0.2	0.0	18	5
150	5	15	1.8	1.5	0.0	647	5	16.7	1.8	**0.0**	**57**	**5**	1.7	0.0	151	5
150	5	24	4.4	4.1	2.0	3334	1	30.8	4.4	**0.0**	**127**	**5**	4.3	0.0	226	5
150	5	30	7.1	4.7	4.0	3600	0	35.5	7.1	**0.0**	**395**	**5**	6.0	0.0	1259	5
150	10	6	0.8	0.7	0.0	229	5	9.3	0.8	**0.0**	**177**	**5**	0.8	0.0	183	5
150	10	15	4.4	3.8	3.4	3600	0	24.2	4.4	3.5	3600	0	4.2	**3.3**	**3600**	**0**
150	10	24	12.8	6.9	**6.4**	**3600**	**0**	34.0	12.8	10.5	3600	0	9.2	8.6	3600	0
150	10	30	15.7	6.9	**6.6**	**3600**	**0**	37.2	15.7	11.9	3600	0	9.5	8.7	3600	0
150	15	6	1.2	0.9	0.0	1064	5	12.1	1.2	0.0	1078	5	1.1	**0.0**	**683**	**5**
150	15	15	9.2	4.9	**4.4**	**3600**	**0**	27.8	9.2	7.3	3600	0	6.8	6.2	3600	0
150	15	24	13.7	4.6	**4.0**	**3600**	**0**	33.3	13.7	10.3	3600	0	6.4	4.9	3600	0
150	15	30	14.6	3.7	2.3	3421	1	33.7	14.6	10.4	3600	0	4.2	**0.9**	**2318**	**3**

Table 4.2: Results obtained testing MET2, MET5 and MET7 with the instances with 125 and 150 customers (5 instances averaged per line). The table includes the integrality gap of the linear relaxation (LRGap), the integrality gap of the linear relaxation after the cuts in the root node (RGap), the integrality gap after 3600 seconds (FGap), the average time in seconds needed to optimally solve the instances (t(s)) and the number of instances solved to optimality in less than the time limit of 3600 seconds

Chapter 5

Conclusions / *Conclusiones*

In this dissertation, we have focused on the study of a pricing optimization problem entitled The Rank Pricing Problem, and two generalizations that consist in adding ties in the list of preferences and capacities. We have introduced the first mixed-integer linear formulations for the three problems, enabling their resolution by means of off-the-shelf solvers. In many cases, we have developed valid inequalities tailored to the problem at hand that we have then separated, solving the problems using a branch-and-cut algorithm. In other cases, we have obtained theoretical results like the characterization of clique facets of model (2IM_1) in Section 2.4 or the complexity result establishing that the assignment problem associated to the CRPP with envy is NP-complete in Section 4.1.1. In what follows, we give a brief summary of the main contributions of this doctoral thesis and discuss possible future research lines.

In Chapter 2, we give two nonlinear formulations for the RPP, one that comes from a bilevel formulation and another one (model (2INLM)) that is directly formulated as a single-level one. We compare them theoretically, linearize their objective functions (which are the same) in two different ways and propose preprocessing techniques that effectively reduce the size of the instances by fixing variables to either zero or one. By studying the SPP associated to the binary variables of the strongest model, we prove that this formulation is very tight, since the majority of its constraints are facet-defining in the corresponding subproblem. The computational results are consistent with our theoretical comparison, given that they show the superiority of the single-level model (2INLM). They also disclose that the linearization carried out using variables z_i^k outperforms the one made through variables z^k regardless of the model, due to the enormous impact of the valid inequalities associated to z_i^k in the reduction of the bounds and the overall performance of the models. These results are carried out to the following chapters, where the linearizations always occur by means of variables z_i^k (the models are already linear in Chapters 3 and 4) and some of the preprocessing (specifically Proposition 2.24) is made in the definition of the variables at the beginning of the chapters.

In Chapter 3, we propose the first formulation with three-index variables for the RPPT (also valid for the RPP). Next, we propose a resolution method based on the introduction of a weaker model with much less variables and constraints and its strengthening by means of valid inequalities. Although the Benders decomposition is a widely used approach for solving mixed-integer programs, in this work we adapt the procedure in a few ways and make it more efficient for our problem. In the first place, we opt for solving a valid model from the beginning instead of the relaxed master problem. Indeed, our Benders Model has a very small set of variables and constraints but is valid for the RPPT, so it does not require the search for constraints when a feasible solution is found during the tree search. Furthermore, we take advantage of the properties of the Benders subproblems (and of the separation problem of the first resolution method) to transform them into min-cost flow problems. In the computational testing, each of the resolution algorithms proposed excels in a specific phase. In the linear relaxation phase, the upper bound of the linear relaxation of the Benders Model initially outperforms that of the two-index model. In this case, one inequality per customer is separated for the Benders Model, whereas an inequality per customer and product is computed for the two-index formulation. But it is interesting how, like in the RPP, the introduction of inequalities that can be separated by products (even if a bigger number of them are introduced) results in a faster separation procedure, that of the two-index model. In the integer phase, however, the Benders Model outperforms the two-index model in the tree search because the node exploration is faster, probably due to the small number of variables and constraints it has.

Chapter 4 begins with the introduction of a model with three-index variables for the CRPP. Unlike in the previous chapters, here the capacity constraints and the inclusion of new variables associated to the capacity allow to derive several sets of valid inequalities for the three-index formulation. These sets are then included in the projection of the three-index formulation into the two-index one, so the resultant inequalities are more difficult to separate because they depend on six sets of parameters. Besides solving the separation problem with Xpress, we theoretically separate a particular set of inequalities (family (4.16)) that proves to be very effective in practice, and link it with the family (2.14) proposed for the RPP. The combination of these inequalities with the last set proposed (4.20) results in a very efficient resolution method (MET7) that combines the best linear relaxation gap (that of (4.20)) with a rather quick inclusion of cuts in the root node and an effective node search provided by (4.16).

An interesting future line of research consists in modifying the problems' characterization of the customers to make them more realistic and the problem more robust. Examples of this include considering that the given rankings correspond to a possible behavior of a customer, instead of a given customer. In this setting, we could represent the customers' choice rule using a probability mass function over the set of rankings that would alter the objective function (very much in the spirit of Bertsimas and Mišić (2019)). Another option consists in considering near-optimal robust bilevel solutions (recently introduced in Besançon (2020)) of the RPP and the RPPT. The seek of a near-optimal robust solution relies on the fact that the lower level (customers in our setting) sometimes makes decisions that are not optimal but are close to the optimal one, so we might want to seek for a global solution that is robust with respect to near-optimal lower level solutions, thus *protecting* the upper level. It is a generalization of the pessimistic approach of the problem where the follower can choose any optimal or near-optimal solution.

Another challenging line of research is to study the pessimistic approach of the allocation of the products in the CRPP with envy. In the CRPP, we assume that the company can decide the allocation of a product to its clients if (and only if) it is sold out in the solution. In the pessimistic version, the company cannot decide the allocation of the products, so it seeks for an optimal pricing assuming the worst possible allocation (a worst-case scenario). Since the real allocation is unknown in general (it may e.g. come from the ordering in which the customers make their purchase), this assumption fits more realistic settings because it gives a lower bound of the expected profit. This setting would very likely need to be studied from a bilevel perspective, and the formulations would be drastically different to the ones introduced here for the CRPP with envy.

En esta tesis doctoral nos hemos centrado en el estudio de un problema de optimización titulado Problema de Tarificación basado en Preferencias, y en dos generalizaciones de este que consisten en la inclusión de empates en la lista de preferencias y capacidades. Hemos introducido las primeras formulaciones lineales enteras mixtas para los tres problemas, facilitando su resolución mediante optimizadores comerciales. En muchos casos, el estudio del problema ha dado lugar a desigualdades válidas que hemos añadido implementando algoritmos de ramificación y corte. En otros casos, hemos llevado a cabo estudios teóricos como la caracterización de las facetas clique del modelo ($2IM_1$) en la Sección 2.4 o el resultado de complejidad que establece que el problema de asignación asociado al CRPP con envidia es NP-duro en el Capítulo 4 (Sección 4.1.1). A continuación, resumimos las principales conclusiones y resultados obtenidos en el trabajo, y damos posibles líneas futuras de investigación.

En el Capítulo 2 introducimos dos formulaciones no lineales para el RPP, una que surge de una formulación binivel y otra (el modelo (2INLM)) que está formulada directamente en un solo nivel. Las comparamos teóricamente, linealizamos su función objetivo (que es la misma en ambos casos) de dos formas distintas y proponemos técnicas de preprocesamiento que reducen el tamaño de las instancias fijando variables binarias a cero o a uno. También estudiamos el problema de empaquetamiento asociado a las variables binarias del modelo más fuerte, con lo que probamos que esta formulación es muy fuerte porque la mayoría de las restricciones que contiene definen facetas del subproblema asociado. Los resultados computacionales son consistentes con nuestra comparación teórica, ya que muestran la superioridad del modelo uninivel (2INLM). También revelan que se obtienen mejores cotas y un mejor rendimiento en general con la linealización llevada a cabo usando variables z_i^k que con la linealización hecha con las variables z^k. Estos resultados se utilizan en los siguientes capítulos: las linealizaciones siempre se llevan a cabo utilizando las variables z_i^k (aunque los modelos se presentan ya linealizados en los capítulos 3 y 4) y parte de las técnicas de preprocesamiento (en concreto la Proposición 2.24) se lleva a cabo en la definición de las variables al principio de los capítulos.

En el Capítulo 3, proponemos la primera formulación con variables de tres índices para el RPPT (que también es válida para el RPP). Luego incluimos un método de resolución basado en la introducción de un modelo más débil con un conjunto de variables y restricciones de menor tamaño, formulación que a continuación fortalecemos añadiendo desigualdades válidas. Aunque la descomposición de Benders es un método de resolución ampliamente utilizado para problemas enteros mixtos, en este trabajo modificamos el procedimiento para adaptarlo a nuestro problema. En primer lugar, optamos por resolver un modelo válido desde el principio, en vez de una relajación de la formulación master. Así, nuestro Modelo de Benders (BDM) tiene un conjunto de variables y restricciones realmente pequeño, pero es válido para el RPPT, por lo que evitamos tener que buscar restricciones cada vez que se encuentra una solución entera en el árbol de ramificación. Además, aprovechamos las propiedades de los subproblemas de Benders (y de los problemas de separación del primer método de resolución) para transformarlos en problemas de flujo a coste mínimo. En los experimentos computacionales, observamos que cada uno de los algoritmos de resolución propuestos destaca en una fase de la resolución del problema. En la fase de la relajación lineal, la cota superior de la relajación lineal del Modelo de Benders es inicialmente mejor que la del modelo de dos índices. En este caso, en el Modelo de Benders se separa una desigualdad por cada cliente, mientras que en el modelo *de dos índices se separa una desigualdad por cliente y producto. Pero es interesante comprobar que, al igual que en el RPP, resulta más rápida la adición de cortes cuando las desigualdades se pueden separar por productos (incluso si se incluyen más cortes). En la fase entera, sin embargo, el Modelo de Benders lleva a cabo la exploración de nodos más rápido que el otro modelo, probablemente debido a su reducido número de variables y restricciones.*

El Capítulo 4 comienza introduciendo una formulación con variables de tres índices para resolver el CRPP. Al contrario que en los capítulos precedentes, en este se incluyen varios conjuntos de desigualdades válidas para dicha formulación que utilizan las restricciones de capacidad y nuevas variables binarias. Estos conjuntos se utilizan después cuando se proyecta el modelo de tres índices en el modelo de dos índices, por lo que las desigualdades obtenidas son más difíciles de separar porque dependen de seis conjuntos de parámetros. Además de resolver el problema de separación con Xpress, también separamos teóricamente un conjunto de desigualdades (el conjunto (4.16)*) que resulta muy efectivo en la práctica, y lo relacionamos con el conjunto de cortes* (2.14) *propuesto para el RPP. La combinación de estas desigualdades con las últimas que proponemos* (4.20) *da como resultado un algoritmo muy eficiente (el método de resolución MET7) que reúne las ventajas de ambas: proporciona unas de las mejores cotas de la relajación lineal (obtenidas usando* (4.20)*) y fases de inclusión de cortes en el nodo raíz y de ramificación muy rápidas (dadas por* (4.16)*).*

Una línea de investigación futura interesante consiste en modificar la caracterización de los clientes para hacerla más realista y para hacer el problema más robusto. Una forma de hacerlo consiste en asumir que los rankings representan en realidad posibles comportamientos de los clientes, en vez de a los propios clientes. En este contexto, la decisión de compra de los consumidores se representaría mediante una distribución de probabilidad en el conjunto de rankings (de forma similar a Bertsimas and Mišić (2019)). Otra opción consiste en considerar una solución binivel robusta casi óptima (recientemente acuñada en Besançon (2020)) del RPP y del RPPT. La búsqueda de una solución casi óptima se basa en el hecho de que los consumidores a menudo toman decisiones que no son óptimas pero están cerca de la óptima, por lo que buscar una solución óptima global robusta con respecto a las decisiones de los consumidores. Es una generalización de la posición pesimista binivel en la que los seguidores pueden elegir cualquier solución óptima o casi óptima.

Otra línea de investigación prometedora es el estudio del CRPP con envidia considerando la posición pesimista en la asignación de los productos a los clientes. En el CRPP, se asume que la compañía puede asignar un producto a los clientes si (y solo si) se agota en la solución. En la versión pesimista, la asignación no puede ser hecha por la compañía, por lo que esta busca una tarificación óptima asumiendo el peor de los casos, es decir, la asignación que da el menor beneficio. Como la asignación real es desconocida (puede venir dada, por ejemplo, por el orden en el que los clientes compran), esta suposición es más realista porque da una cota inferior del beneficio esperado. Muy probablemente, habría que estudiar este caso desde una perspectiva binivel, y las formulaciones resultantes serían drásticamente distintas a las aquí propuestas para el CRPP.

Bibliography

Aboussoror, A., & Adly, S. (2018). New necessary and sufficient optimality conditions for strong bilevel programming problems. *Journal of Global Optimization*, *70*(2), 309–327.

Aggarwal, G., Feder, T., Motwani, R., & Zhu, A. (2004). Algorithms for Multi-product Pricing. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. Díaz, J. Karhumäki, A. Lepistö, & D. Sannella (Eds.), *Automata, Languages and Programming* (pp. 72–83). Springer Berlin Heidelberg.

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1988). *Network flows.* Cambridge, Mass.: Alfred P. Sloan School of Management, Massachusetts.

Atamtürk, A., Nemhauser, G. L., & Savelsbergh, M. W. P. (2000). Conflict graphs in solving integer programming problems. *European Journal of Operational Research*, *121*(1), 40–55.

Balas, E. (1971). Intersection cuts—a new type of cutting planes for integer programming. *Operations Research*, *19*(1), 19–39.

Balas, E. (1972). Integer programming and convex analysis: Intersection cuts from outer polars. *Mathematical Programming*, *2*(1), 330–382.

Balas, E., Ceria, S., & Cornuéjols, G. (1993). A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, *58*(1), 295–324.

Balas, E., & Padberg, M. W. (1976). Set partitioning: A survey. *SIAM review*, *18*(4), 710–760.

Balas, E., & Pulleyblank, W. (1983). The perfectly matchable subgraph polytope of a bipartite graph. *Networks*, *13*(4), 495–516.

Balcan, M.-F., Blum, A., & Mansour, Y. (2008). Item Pricing for Revenue Maximization. *Proceedings of the 9th ACM Conference on Electronic Commerce*, 50–59.

Balcan, M.-F., Blum, A., Hartline, J. D., & Mansour, Y. (2005). Mechanism design via machine learning. *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, 605–614.

Bard, J. F. (1984). Optimality conditions for the bilevel programming problem. *Naval Research Logistics Quarterly*, *31*(1), 13–26.

Belloni, A., Freund, R., Selove, M., & Simester, D. (2008). Optimizing Product Line Designs: Efficient Methods and Comparisons. *Management Science*, *54*(9), 1544–1552.

Ben-Ameur, W., & Neto, J. (2007). Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, *49*(1), 3–17.

Benders, J. F. (1962). Partitioning Procedures for Solving Mixed-Variable Programming Problems. *Numerische Mathematik*, *4*(1), 238–252.

Bertsimas, D., & Mišić, V. V. (2019). Exact First-Choice Product Line Optimization. *Operations Research*, *67*(3), 651–670.

Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization* (Vol. 6). Athena Scientific Belmont, MA.

Besançon, M. (2020). *Bilevel models for demand response in smart grids* (Doctoral dissertation). Centrale Lille; Polytechnique Montréal.

Bialas, W. F., & Karwan, M. H. (1984). Two-level linear programming. *Management Science*, *30*(8), 1004–1020.

Bonami, P., Salvagnin, D., & Tramontani, A. (2020). Implementing automatic Benders decomposition in a modern MIP solver. *International Conference on Integer Programming and Combinatorial Optimization*, 78–90.

Briest, P. (2008). Uniform Budgets and the Envy-Free Pricing Problem. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, & I. Walukiewicz (Eds.), *Automata, Languages and Programming* (pp. 808–819). Springer Berlin Heidelberg.

Briest, P., & Krysta, P. (2006). Buying Cheap is Expensive: Hardness of Non-Parametric Multi-Product Pricing. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 716–725.

Bucarey, V., Elloumi, S., Labbé, M., & Plein, F. (2021). Models and algorithms for the product pricing with single-minded customers requesting bundles. *Computers & Operations Research*, *127*, 105139.

Calvete, H. I., Domínguez, C., Galé, C., Labbé, M., & Marín, A. (2019). The rank pricing problem: Models and branch-and-cut algorithms. *Computers & Operations Research*, *105*, 12–31.

Cánovas, L., García, S., Labbé, M., & Marín, A. (2007). A strengthened formulation for the simple plant location problem with order. *Operations Research Letters*, *35*(2), 141–150.

Cánovas, L., Landete, M., & Marín, A. (2000). New facets for the set packing polytope. *Operations Research Letters*, *27*(4), 153–161.

Chen, K. D., & Hausman, W. H. (2000). Technical Note: Mathematical Properties of the Optimal Product Line Selection Problem Using Choice-Based Conjoint Analysis. *Management Science*, *46*(2), 327–332.

Cheng, E., & Cunningham, W. H. (1997). Wheel inequalities for stable set polytopes. *Mathematical Programming*, *77*(2), 389–421.

Chvàtal, V. (1973). Edmonds Polytopes and a Hierarchy of Combinatorial Problems. *Discrete Mathematics*, *4*, 305–337.

Conforti, M., Cornuéjols, G., & Zambelli, G. (2014). *Integer Programming* (Vol. 271). Springer.

Dantzig, G. B., & Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, *8*(1), 101–111.

Delorme, M., García, S., Gondzio, J., Kalcsics, J., Manlove, D., & Pettersson, W. (2019). Mathematical models for stable matching problems with ties and incomplete lists. *European Journal of Operational Research*, *277*(2), 426–441.

Dempe, S., Dinh, N., & Dutta, J. (2010). Optimality conditions for a simple convex bilevel programming problem. *Variational Analysis and Generalized Differentiation in Optimization and Control* (pp. 149–161). Springer.

Dempe, S. (1992a). A necessary and a sufficient optimality condition for bilevel programming problems. *Optimization*, *25*(4), 341–354.

Dempe, S. (1992b). Optimality conditions for bilevel programming problems. *System Modelling and Optimization* (pp. 17–24). Springer.

Dempe, S. (2002). *Foundations of Bilevel Programming.* (Vol. 61). Kluwer Academic Publishers.

Dempe, S., Mordukhovich, B. S., & Zemkoho, A. B. (2014). Necessary optimality conditions in pessimistic bilevel programming. *Optimization*, *63*(4), 505–533.

Dobson, G., & Kalish, S. (1988). Positioning and Pricing a Product Line. *Marketing Science*, *7*(2), 107–125.

Dobson, G., & Kalish, S. (1993). Heuristics for Pricing and Positioning a Product-Line Using Conjoint and Cost Data. *Management Science*, *39*(2), 160–175.

Domínguez, C., Labbé, M., & Marín, A. (2021a). *The Capacitated Rank Pricing Problem with envy* (tech. rep.).

Domínguez, C., Labbé, M., & Marín, A. (2021b). The rank pricing problem with ties. *European Journal of Operational Research*, *294*(2), 492–506.

Escudero, L. F., Landete, M., & Marín, A. (2009). A branch-and-cut algorithm for the Winner Determination Problem. *Decision Support Systems*, *46*(3), 649–659.

Espejo, I., Marín, A., & Rodríguez-Chía, A. M. (2012). Closest assignment constraints in discrete location problems. *European Journal of Operational Research*, *219*(1), 49–58.

Farkas, J. (1902). Theorie der einfachen Ungleichungen. *Journal für die reine und angewandte Mathematik*, *1902*(124), 1–27.

Fernandes, C. G., Ferreira, C. E., Franco, Á. J. P., & Schouery, R. C. S. (2016). The envy-free pricing problem, unit-demand markets and connections with the network pricing problem. *Discrete Optimization*, *22*, 141–161.

Fischetti, M., Ljubić, I., & Sinnl, M. (2016). Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, *253*(3), 557–569.

Fischetti, M., Salvagnin, D., & Zanette, A. (2010). A note on the selection of Benders' cuts. *Mathematical Programming*, *124*(1), 175–182.

Fulkerson, D. R. (1971). Blocking and anti-blocking pairs of polyhedra. *Mathematical Programming*, *1*(1), 168–194.

Gale, D., & Shapley, L. S. (1962). College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, *69*(1), 9–15.

García-Herreros, P., Zhang, L., Misra, P., Arslan, E., Mehta, S., & Grossmann, I. E. (2016). Mixed-integer bilevel optimization for capacity planning with rational markets. *Computers & Chemical Engineering*, *86*, 33–47.

Garey, M. R., & Johnson, D. S. (1978). "Strong" NP-Completeness Results: Motivation, examples, and implications. *Journal of the ACM (JACM)*, *25*(3), 499–508 PUB27 New York, NY, USA.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman.

Geoffrion, A. M. (1974). Lagrangean relaxation for integer programming. *Approaches to Integer Programming* (pp. 82–114). Springer.

Goldberg, A. V., & Hartline, J. D. (2001). Competitive Auctions for Multiple Digital Goods. In G. Goos, J. Hartmanis, J. van Leeuwen, & F. M. auf der Heide (Eds.), *European Symposium on Algorithms* (pp. 416–427). Springer Berlin Heidelberg.

Goldberg, A. V., Hartline, J. D., Karlin, A. R., Saks, M., & Wright, A. (2006). Competitive auctions. *Games and Economic Behavior*, *55*(2), 242–269.

Gomory, R. E. (1963). An Algorithm for Integer Solutions to Linear Programs. In R. L. Graves & P. Wolfe (Eds.), *Recent Advances in Mathematical Programming* (pp. 269–302). McGraw-Hill.

Green, P. E., & Krieger, A. M. (1985). Models and Heuristics for Product Line Selection. *Marketing Science*, *4*(1), 1–19.

Green, P. E., & Krieger, A. M. (1993). Conjoint analysis with product-positioning applications. *Handbooks in Operations Research and Management Science* (pp. 467–515). Elsevier.

Guruswami, V., Hartline, J. D., Karlin, A. R., Kempe, D., Kenyon, C., & McSherry, F. (2005). On Profit-maximizing Envy-free Pricing. *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1164–1173.

Gusfield, D., & Irving, R. W. (1989). *The stable marriage problem: Structure and algorithms*. MIT press.

Hanjoul, P., & Peeters, D. (1987). A facility location problem with clients' preference orderings. *Regional Science and Urban Economics*, *17*(3), 451–473.

Hansen, P., Kochetov, Y., & Mladenovic, N. (2004). Lower bounds for the uncapacitated facility location problem with user preferences. *Groupe d'études et de recherche en analyse des décisions, HEC Montréal.*, 6.

Hartline, J. D., & Koltun, V. (2005). Near-Optimal Pricing in Near-Linear Time. In F. Dehne, A. López-Ortiz, & J.-R. Sack (Eds.), *Algorithms and Data Structures* (pp. 422–431). Springer.

Heilporn, G., Labbé, M., Marcotte, P., & Savard, G. (2010). A parallel between two classes of pricing problems in transportation and marketing. *Journal of Revenue and Pricing Management*, *9*(1-2), 110–125.

Hemmati, M., & Smith, J. C. (2016). A mixed-integer bilevel programming approach for a competitive prioritized set covering problem. *Discrete Optimization*, *20*, 105–134.

Ishizuka, Y. (1988). Optimality conditions for quasi-differentiable programs with application to two-level optimization. *SIAM Journal on Control and Optimization*, *26*(6), 1388–1398.

Jeroslow, R. G. (1985). The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, *32*(2), 146–164.

Khandekar, R., Kimbrel, T., Makarychev, K., & Sviridenko, M. (2009). On Hardness of Pricing Items for Single-Minded Bidders. In I. Dinur, K. Jansen, J. Naor, & J. Rolim (Eds.), *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* (pp. 202–216). Springer.

Kraus, U. G., & Yano, C. A. (2003). Product line selection and pricing under a share-of-surplus choice model. *European Journal of Operational Research*, *150*(3), 653–671.

Kwanashie, A., & Manlove, D. F. (2014). An Integer Programming Approach to the Hospitals/Residents Problem with Ties. *Operations Research Proceedings 2013*, 263–269.

Landete, M. (2001). *Obtención de facetas de poliedros asociados a problemas de empaquetamiento* (Doctoral dissertation). University of Murcia.

Magnanti, T. L., & Wong, R. T. (1981). Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. *Operations Research*, *29*(3), 464–484.

Manlove, D. F. (2013). *Algorithmics of Matching Under Preferences*. World Scientific.

Manlove, D. F., Irving, R. W., Iwama, K., Miyazaki, S., & Morita, Y. (2002). Hard variants of stable marriage. *Theoretical Computer Science*, *276*(1), 261–279.

McBride, R. D., & Zufryden, F. S. (1988). An integer programming approach to the optimal product line selection problem. *Marketing Science*, *7*(2), 126–140.

Minkowski, H. (1896). Geometrie der zahlen (2 vol.) *Teubner, Leipzig*, *1910*(1896).

Myklebust, T., Sharpe, M., & Tunçel, L. (2016). Efficient heuristic algorithms for maximum utility product pricing problems. *Computers & Operations Research*, *69*, 25–39.

Naoum-Sawaya, J., & Elhedhli, S. (2013). An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, *210*(1), 33–55.

Nemhauser, G. L., & Trotter, L. E. (1974). Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, *6*(1), 48–61.

Nemhauser, G. L., & Wolsey, L. A. (1999). *Integer and Combinatorial Optimization.* John Wiley & Sons.

Nemhauser, G. L., & Wolsey, L. A. (1990). A recursive procedure to generate all cuts for 0–1 mixed integer programs. *Mathematical Programming*, *46*(1), 379–390.

Outrata, J. V. (1993). Necessary optimality conditions for Stackelberg problems. *Journal of Optimization Theory and Applications*, *76*(2), 305–320.

Padberg, M. W. (1973). On the facial structure of set packing polyhedra. *Mathematical Programming*, *5*(1), 199–215.

Padberg, M. W. (1975). A note on zero-one programming. *Operations Research*, *23*(4), 833–837.

Papadimitriou, C. H. (1981). On the complexity of integer programming. *Journal of the ACM (JACM)*, *28*(4), 765–768.

Paruchuri, P., Pearce, J. P., Kraus, S., & Marecki, J. (2008). Playing Games for Security: An Efficient Exact Algorithm for Solving Bayesian Stackelberg Games. *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, *2*, 895–902.

Rothblum, U. G. (1992). Characterization of stable matchings as extreme points of a polytope. *Mathematical Programming*, *54*(1-3), 57–67.

Rusmevichientong, P. (2003). *A non-parametric approach to multi-product pricing: Theory and application.* (Doctoral dissertation). Stanford University.

Rusmevichientong, P., Van Roy, B., & Glynn, P. W. (2006). A non-parametric approach to multiproduct pricing. *Operations Research*, *54*(1), 82–98.

Santos, F. (2012). A counterexample to the Hirsch conjecture. *Annals of Mathematics*, *172*, 383–412.

Savard, G., & Gauvin, J. (1994). The steepest descent direction for the nonlinear bilevel programming problem. *Operations Research Letters*, *15*(5), 265–272.

Schön, C. (2010a). On the Optimal Product Line Selection Problem with Price Discrimination. *Management Science*, *56*(5), 896–902.

Schön, C. (2010b). On the product line selection problem under attraction choice models of consumer behavior. *European Journal of Operational Research*, *206*(1), 260–264.

Schrijver, A. (1998). *Theory of Linear and Integer Programming*. John Wiley & Sons.

Sherali, H. D., & Adams, W. P. (1998). Reformulation-Linearization Techniques for Discrete Optimization Problems. In D.-Z. Du & P. M. Pardalos (Eds.), *Handbook of Combinatorial Optimization: Volume1–3* (pp. 479–532). Springer US.

Shioda, R., Tunçel, L., & Myklebust, T. G. J. (2011). Maximum Utility Product Pricing Models and Algorithms Based on Reservation Prices. *Computational Optimization and Applications*, *48*(2), 157–198.

Sipser, M. (2012). *Introduction to the Theory of Computation* (2nd Edition). Cengage Learning.

Trotter, L. E. (1975). A class of facet producing graphs for vertex packing polyhedra. *Discrete Mathematics*, *12*(4), 373–388.

van Roy, T. J., & Erlenkotter, D. (1982). A Dual-Based Procedure for Dynamic Facility Location. *Management Science*.

Vande Vate, J. H. (1989). Linear programming brings marital bliss. *Operations Research Letters*, *8*(3), 147–153.

Vemuganti, R. R. (1998). Applications of Set Covering, Set Packing and Set Partitioning Models: A Survey. In D.-Z. Du & P. M. Pardalos (Eds.), *Handbook of Combinatorial Optimization: Volume1–3* (pp. 573–746). Springer US.

Vicente, L. N., & Calamai, P. H. (1995). Geometry and local optimality conditions for bilevel programs with quadratic strictly convex lower levels. *Minimax and Applications* (pp. 141–151). Springer.

Weyl, H. (1950). The Elementary Theory of Convex Polyhedra. In H. Kuhn & A. Tucker (Eds.). Princeton University Press.

Wolsey, L. A. (1998). *Integer Programming*. John Wiley & Sons.