University of Murcia

END OF DEGREE THESIS

The Rank Pricing Problem

By
Marcos Gil Espinosa

Supervised by
Alfredo Marín Pérez

May 2023

Contents

Nomenclature

$\sigma : K \to \{1, \ldots, M\}$ Function assigning each client the index of its budget

B	Ordered set of budgets
b^j	j'th budget
I	Set of products
K	Set of customers
M	Number of different budgets
S^k	Set of products in which customer k is interested
s_i^k	Preference value of product i for customer k

Resumen (Español)

El objeto del presente trabajo es el Problema de Asignación de Precios con Rango (en inglés, *Rank Pricing Problem*, RPP). Éste es un problema de optimización consistente en determinar de qué forma debe un productor asignar precios a sus productos de manera que se maximice su beneficio. A continuación, presentaremos el RPP y daremos al lector un resumen de la estructura del trabajo.

El Problema de Asignación de Precios con Rango supone la existencia de un vendedor y de un conjunto de clientes. El vendedor ofrece diversos productos, y desea asignar un precio a cada uno de manera que se maximice su beneficio. A su vez, cada cliente viene caracterizado por una lista de preferencias. Esta lista está conformada por un conjunto de productos de entre los ofertados por el vendedor. Los productos de estas listas de preferencias están totalmente ordenados, de manera que, para cada par de productos i_1 e i_2 deseados por un cliente k, bien i_1 es estrictamente mejor que i_2 para k, bien i_2 es estrictamente mejor que i_1 para k. Todo cliente tiene además un presupuesto, que es el máximo gasto que se puede permitir. Así pues, cada cliente desea adquirir exactamente un producto, que sea de máxima preferencia para él, pero cuyo precio sea menor o igual que su presupuesto.

El RPP ha sido tratado en profundidad en el artículo [Calvete et al., 2019]. Es de esta fuente de donde se han extraído la mayor parte de resultados de los capítulos segundo, tercero y cuarto. A menos que se indique lo contrario, todo resultado se debe entender contenido en el artículo de Calvete y otros.

Este trabajo se encuentra dividido en seis capítulos, cuyos contenidos se resumen a continuación.

El desarrollo del trabajo requiere de resultados y definiciones pertenecientes a tres áreas diferentes de la optimización: programación lineal, programación entera y programación binivel. Éstos se presentan en el primer capítulo.

Un entendimiento de programación lineal es requisito necesario para comprender los modelos de programación entera que forman el núcleo de este trabajo. Por este motivo se le dedica la primera sección de la introducción. Tras una breve reseña histórica de este paradigma, se dan varias de sus definiciones elementales. Además, se presentan brevemente varios resultados referidos a teoría de la dualidad. Éstos son usados en el segundo capítulo, donde permiten derivar una formulación de un solo nivel a partir de las formulaciones binivel inicialmente definidas.

La programación entera abarca la segunda sección de la introducción. Este paradigma permite definir la mayoría de las formulaciones del trabajo. Se presentan pues, varias definiciones elementales, y se analiza su complejidad computacional y dos métodos de resolución: ramificación y acotación, y ramificación y corte. Además, se introducen varios resultados relacionados con matrices totalmente unimodulares que permiten resolver algunos problemas enteros como si fueran lineales. Estos resultados serán de interés en el segundo capítulo.

Finalmente, la programación binivel abarca la tercera sección de la introducción. Esta rama de la optimización estudia la definición de problemas con una estructura en dos niveles. En el problema de primer nivel, se desea resolver un problema de programación lineal o entera. Además, entre las restricciones de este problema de primer nivel se requiere que algunas variables sean solución de un problema de segundo nivel. La programación binivel permite dar en el segundo capítulo dos formulaciones del Problema de Asignación de Precios con una interpretación muy natural de los conceptos de cliente y vendedor.

La introducción contiene, además, una introducción al RPP, y una sección presentando varios resultados de teoría de grafos. Estos resultados serán la base del cuarto capítulo. Su interés se debe a que proporcionan una manera de traducir afirmaciones sobre desigualdades válidas para un problema de optimización, el problema de empaquetamiento de conjuntos (SPP), en afirmaciones sobre subgrafos de un cierto grafo asociado al SPP.

El segundo capítulo presenta cuatro formulaciones para el RPP. En primer lugar se presentan dos formulaciones binivel en que se asigna al vendedor un problema de primer nivel, y a cada uno de los clientes un problema de segundo nivel. Así pues, el vendedor elige los precios de los productos maximizando su beneficio, mientras que los clientes buscan elegir un producto dentro de su presupuesto con *nivel de preferencia* máximo. La elección concreta de variables da lugar a dos formulaciones, (BNL^p) y (BNL^v). (BNL^p) nace de representar el precio del producto i mediante la variable p_i, que es una variable real indicando el precio de i. La formulación (BNL^v) se obtiene a partir de (BNL^p) notando la aplicabilidad de un resultado dado en [Rusmevichientong et al., 2006]. Éste asegura que en la solución óptima del RPP los precios asignados pertenecen al conjunto de diferentes presupuestos de los clientes. Definiendo B como el conjunto de diferentes presupuestos, podemos entonces representar el precio del producto i mediante v_i^ℓ, que es una variable binaria indicando si el producto i recibe el precio $\ell \in B$.

Deseando eludir la alta complejidad computacional de las formulaciones binivel, se deriva de (BNL^v) una formulación de un solo nivel, (BNL), fruto de caracterizar las soluciones del problema de segundo nivel mediante varias restricciones lineales. A partir de (BNL) se obtiene una segunda formulación de un solo nivel (SLNL), que servirá de base al resto del trabajo.

El tercer capítulo parte de la formulación (SLNL) para obtener dos formulaciones lineales. El motivo de este cambio vuelve a ser el deseo de resolver el RPP con un método de la menor complejidad computacional posible pues, efectivamente, la resolución de los problemas no lineales es notoriamente más compleja que la de los lineales. El elemento problemático en (SLNL) es su función objetivo, que contiene productos de variables. Así pues, la linealización de esta formulación se realiza transformando la función objetivo en una función lineal. Se define para ello una nueva familia de variables que permiten expresar la función objetivo de manera lineal. Además, es necesario incluir varias restricciones que permiten dar sentido a las nuevas variables.

La elección concreta de nuevas variables da lugar a dos formulaciones lineales distintas: (SLL$_1$) y (SLL$_2$). Las nuevas variables de la primera son z^k, el beneficio obtenido por el vendedor del cliente k. Las de la segunda son más granulares, z_i^k, representando el beneficio obtenido por el vendedor de vender el producto i al cliente k. Nótese que si en la solución óptima un cliente k_0 no adquiere un producto i_0, entonces $z_{i_0}^{k_0} = 0$.

También con ánimo de mejorar el tiempo de resolución se definen dos familias de desigualdades válidas, una para cada formulación lineal. Éstas resultan tener un tamaño exponencial, por lo que se hace necesario estudiarlas minuciosamente y dar un algoritmo de separación para cada familia. Estos algoritmos toman como entrada la solución fraccionaria de alguna relajación lineal del problema y devuelven una restricción de la familia que no sea cumplida por la solución fraccional. Tal procedimiento resulta útil para resolver el RPP mediante ramificación y corte.

A diferencia de los dos anteriores, el objetivo del cuarto capítulo no es mejorar las formulaciones obtenidas, sino determinar cuán buenas son. Se comienza notando que varias de las restricciones de las formulaciones (SLL_1) y (SLL_2) forman lo que se conoce como un Problema de Empaquetamiento de Conjuntos (en inglés, *Set Packing Problem*, SPP). Tal descubrimiento nos permite emplear las técnicas y resultados presentados al término del Capítulo 1 para determinar cuáles de las restricciones de (SLL_1) y (SLL_2) son facetas para sus subproblemas de empaquetamiento de conjuntos. Para ello necesitamos considerar un objeto, llamado grafo intersección G, asociado al problema de empaquetamiento de conjuntos. Un resultado debido a Padberg [Padberg, 1973] nos permite relacionar facetas de SPP con cliqués de G. Este cambio de lenguaje de problema de empaquetamiento a teoría de grafos nos permite concluir que la mayor parte de las restricciones del SPP son facetas del mismo, lo que implica que las formulaciones (SLL_1) y (SLL_2) no admiten una mejora trivial.

Finalmente, el quinto capítulo presenta las contribuciones originales del autor. Éstas se resumen en dos: extensión de las formulaciones (SLL_1) y (SLL_2) al problema CRPP sin envidia y estudio computacional de dichas extensiones.

El Problema de Asignación de Precios con Rango y Capacidades (en inglés, *Capacitated Rank Pricing Problem*, CRPP) es una extensión del RPP. Las adiciones introducidas por CRPP son el permitir definir limitaciones de inventario y el permitir a cada cliente tener un presupuesto diferente para cada producto. Las limitaciones de inventario permiten modelar situaciones en que no se puede producir un bien de manera ilimitada. Por ejemplo, un teatro pude ofertar 50 localidades en palco y 200 en la platea, pero ninguna más, por limitaciones de espacio. El CRPP tiene dos variantes: con y sin envidia. Se dice que una solución presenta envidia si algún cliente k desea adquirir un producto i cuyo precio es menor o igual que su presupuesto, pero no puede hacerlo porque se han agotado sus existencias. El CRPP con envidia es aquel en que se permiten soluciones con envidia. A su vez, CRPP sin envidia se refiere a aquel en que no se permiten soluciones que presenten envidia. Nosotros presentamos dos formulaciones para el CRPP sin envidia basadas en las formulaciones (SLL_1) y (SLL_2) antes aludidas.

Por último, el quinto capítulo presenta también un estudio computacional en que se estudia la eficiencia de las formulaciones propuestas para el CRPP. Las instancias empleadas para este estudio han sido generadas aleatoriamente.

Abstract (English)

The object of study of the present work is the Rank Pricing Problem (RPP), an optimisation problem in which some products must me priced so as to maximise their seller's gains. Now, we will present the RPP, and provide the reader a summary of the structure of this work.

The Rank Pricing Problem presupposes the existence of a seller and of a set of clients. The seller offers various goods and wishes to prize them so as to maximise his benefits. In turn, each client is characterised by a list of preferences. These lists consist of products from those offered by the seller. Products in them are totally ordered, so that for each pair of distinct products i_1 and i_2 in the preference list of some customer k, either i_1 is strictly better than i_2 for k, or i_2 is strictly better than i_1 for k. Furthermore, each client has a budget, which is the maximum amount of money he can afford to expend. In this setting, each client wishes to purchase exactly one product of maximum preference for him, and whose prices be less than or equal to his budget.

The RPP has been treated in depth in the article [Calvete et al., 2019]. Most of the results of chapters 2, 3 and 4 have been extracted from this source. Unless otherwise stated, all results should be considered to have been extracted from the article of Calvete et al.

This work is divided in six chapters, whose contents we will now summarise.

The present work requires several definitions and results from three different areas of optimisation. Namely linear, integer, and bilevel programming. These are presented in the first chapter.

An understanding of linear programming is a necessary requirement for understanding the integer programming models which conform the core of this work. For this reason, it is treated in the first section of the introduction. After giving a short historical overview of this paradigm, several of its elementary definitions are presented. Furthermore, several results from duality theory are shortly presented. These are needed in the second chapter, where they are used to derive a single-level formulation from the bilevel formulations initially defined.

Integer programming is introduced in the second section of the introduction. This paradigm is used to define most formulations in this work. Thus, several of its elementary definitions are presented, as well as comments on its computational complexity and on two of its resolution methods: branch and bound, and branch and cut. Furthermore, we introduce several results on totally unimodular matrices, which allow solving some integer problems as linear ones. These results are used in the second chapter.

Finally, bilevel programming is presented in the third section of the introduction. This branch of optimisation studies the definition of problems with a bilevel structure. On the first level, one wishes to solve a linear or integer programming problem. Furthermore, amongst the restrictions of this first level problem stands the requirement that some variables be solution to a second problem. Bilevel programming permits giving some interpretations of the RPP in which the concepts of *seller* and *client* find quite natural translations as levels of a bilevel program. These matters are discussed in the second chapter.

The introduction also contains a presentation of the RPP, and a section listing various results from graph theory. Their interest lies in that they allow translating statements about valid inequalities for some optimisation problem, the Set Packing Problem (SPP), into statements about subgraphs of a certain graph associated to the SPP.

The second chapter presents four formulations for the RPP. Two bilevel formulations are first introduced. In these, the seller is represented by the first level problem, and each client is represented by one of the second level problems. In this situation, the seller prices its products so as to maximise his revenue, whereas clients choose a maximum-preference product of price less than or equal to their budgets. Which variables to use gives rise to two different formulations, (BNL^p) and (BNL^v). In (BNL^p), the price of product i is represented by p_i, a real variable explicitly indicating i's price. Formulation (BNL^v) can be obtained from (BNL^p) by employing a result from [Rusmevichientong et al., 2006]. This result states that in an optimal solution to the RPP, prices assigned to items belong to the set of all client budgets. Representing this set by B, the price of item i can then be represented by v_i^ℓ. v_i^ℓ is a binary variable indicating whether product i is assigned the price $\ell \in B$.

Wishing to elude the high computational complexity of bilevel formulations, a single-level formulation (BNL) is derived from (BNL^v) by characterising solutions to (BNL^v) 's second level problem through linear restrictions. From (BNL), a second single-level formulation (SLNL) is derived. This last formulation will serve as basis for the rest of the work.

The third chapter derives two linear formulations from (SLNL) . The reason for this change is again the desire to develop a method of the least possible computational complexity. Indeed, solving non linear problems is generally much harder than solving linear ones. The problematic element in (SLNL) is its objective function, which contains products of variables. Therefore, linealisation of (SLNL) is performed by transforming its objective function into a linear function. With this purpose, a new family of variables is defined which allows expressing the objective function in a linear manner. Furthermore, it is necessary to include several restrictions that give the new variables their intended meaning.

The concrete election of which variables to add gives rise to two distinct formulations: (SLL_1) and (SLL_2). The new variables in (SLL_1) are z^k, the benefit obtained by the seller from client k. New variables in (SLL_2) are more granular: z_i^k, representing the benefit obtained by the seller by selling product i to client k. Please note that if some client k_0 does not purchase a particular product i_0, then $z_{i_0}^{k_0} = 0$.

Two families of valid inequations, one for each linear formulation, are defined. These are of exponential size, so that it becomes necessary to study them minutely in order to derive a separation algorithm for each family. These algorithms are provided as entry a fractional solution for some linear relaxation of the problem, and return a restriction from the family not satisfied by the fractional solution. Such methods are useful for solving the RPP by the branch and bound procedure.

The objective of the fourth chapter, is to determine how good provided formulations are. This is in stark contrast with chapters 2 and 3, which merely tried to improve the existing formulations, without trying to assess how tight their restrictions were. First, it is noted that several restrictions from formulations (SLL_1) and (SLL_2) give rise to the feasible region of a Set Packing Problem (SPP). This finding allows us to employ results presented at the end of Chapter 1 in order to determine which restrictions of (SLL_1) and (SLL_2) are facets of the corresponding set packing subproblems. In order to do this, we need to consider the so-called intersection graph G of the set packing subproblem. An important result due to Padberg [Padberg, 1973] allows us to relate facets of the SPP to cliques in G. This change of language from set packing to graph theory allows us to conclude that most restrictions of the SPP are facets. This implies that no trivial improvement can be made to formulations (SLL_1) and (SLL_2).

Finally, the fifth chapter presents the author's original contributions. These are the extension of formulations (SLL_1) and (SLL_2) to the envy-free CRPP and a computational study of these formulations.

The Capacitated Rank Pricing Problem (CRPP) is an extension of the RPP. The additions it introduces are that it permits defining stock limitations and that it allows clients to have different budgets for different products. Stock limitations allow modelling situations in which a certain product cannot be produced without limitation. For instance, a theatre may offer 50 box seats and 200 seats in the stalls, but no more because of lack of space. CRPP has two variants: with and without envy. A solution is said to present envy if some customer k wishes to acquire a product i whose price is less than or equal to his budget, but cannot do so because it has run out of stock. CRPP with envy is the CRPP version allowing solutions to present envy. Conversely, the envy-free version does not allow solutions that present envy. We have developed two formulations to the envy-free CRPP based in formulations (SLL_1) and (SLL_2) .

The fifth chapter also presents a computational study, in which the efficiency of the proposed CRPP formulations is assessed. For this purpose, several CRPP instances have been randomly generated.

Chapter 1

Introduction

This chapter will present the theoretical foundations and mathematical prerequisites of the present work. It consists of five sections: Section 1.1 introduces linear programming and duality theory. Even though the formulations considered will mostly contain integer variables, an understanding of linear programming is a necessary prerequisite to follow our discussion. Section 1.2 presents integer programming, its resolution methods, and several practical matters, mainly considerations in computational complexity and restriction reinforcement. Section 1.3 presents bilevel programming, an optimisation paradigm which will allow for the natural derivation of a formulation in Chapter 2. Section 1.4 presents the Rank Pricing Problem, the theme of this work. Finally, Section 1.5 presents the Set Packing Problem and some concepts of Graph Theory which will be useful for Chapter 4.

In the first section, we will follow the first chapter of [Bazaraa et al., 2009] and Dantzig's review of the first years of linear programming [Dantzig, 1981]; in the second section, the first, seventh, and eighth chapters of [Wolsey, 1988] will be used; in the third section we will follow the introduction of [Dempe, 2002]; in the fourth section, the paper [Calvete et al., 2019]; and, finally, material in Section 1.5 is taken from [Padberg, 1973].

1.1 Linear programming

1.1.1 Historical introduction

The origins of linear programming

The initial conception of linear programming is usually credited to George B. Dantzig, who discovered it during his work as counsellor for the U.S. Air Force in 1947. Several individuals who realised the potential of the area made prior independent discoveries, but most of their advances were eventually forgotten. The most notorious work was that of Soviet mathematician Leonid Kantorovich in 1939, which became known in the west in 1959.

Dantzig's discovery was a result of working for the U.S. Military on task assignment problems. By 1946 he had developed a model for the problem, but had no resolution method. In 1947, he modelled this same problem as a set of linear restrictions over a set of real decision variables, alongside a function to be maximised subject to those restrictions. The inclusion of an objective function was a novel feature, as was the clear distinction of requirements and objectives. As Dantzig exemplifies [Dantzig, 1981], officers in the U.S. Army would state vague objectives, such as *win the war*, or confuse goals and means in assertions like *The way to win is to build a great fleet of bombers*. Thus, distinguishing both concepts was a conceptually relevant step in the road towards developing clear mathematical formulations. In Dantzig's own words, *The ability to state general objectives and then find optimal policy solutions to practical decision problems of great complexity is a revolutionary development.*

After arriving at this formulation, the question arose as to whether a solution method existed. To get possible hints, he resorted to economist Tjalling C. Koopmans, of the University of Chicago. Koopmans was unable to provide him with hints of any method, but was excited at Dantzig's discoveries, realising their value for economic planning. Koopmans played an important role in disseminating Linear Programming among young economists, resulting in works honoured with several Nobel prizes in Economy. Not having found any existing solving method among economists, Dantzig was forced to develop a method of his own. In the summer of 1947, he finally arrived at what would become known as the Simplex method.

A note on terminology

The term *programming* comes from *program*, which was the term employed by the U.S. Military for referring to their plans and schedules for training, logical supply, and deployment of combat units. The original internal report of Dantzig in the U.S. Air Force was titled *Programming in a linear structure*. It was Koopmans who, taking a walk with Danzig in Santa Monica Beach, suggested shortening the name to *Linear programming*. Since then, the term *programming* has been used interchangeably with *optimisation* for referring to various areas of mathematical optimisation, like integer optimisation or stochastic optimisation.

The term *Simplex Method* resulted from a discussion of Dantzig with Theodore Motzkin, who thought Dantzig's method would be best geometrically described as a series of movements among neighbouring simplices in the feasible region. Finally, even though the term *dual* has ancient roots in mathematics, the antonym *primal* was an invention of Dantzig's father, Tobias Dantzig, also a mathematician, around 1959.

1.1.2 Definition of the general linear programming problem

Definition

Optimisation is the branch of applied mathematics concerned with maximisation and minimisation of functions over some given space. The concrete nature of these functions and their domains greatly influences the kind of methods applicable to their optimisation. In this section, we will consider the case of optimising a linear function over a convex polyhedron in $\mathbb{R}^n$ implicitly defined by the inequalities associated to its outer faces. More concretely, we give the following definition:

Definition 1 (Linear programming problem)**.** *By* linear problem *or* linear programming formulation*, we mean the problem of maximising some linear function, $\sum_i c_i x_i$, with $c_i \in \mathbb{R}$, over the set of all $x \in \mathbb{R}^n$ with non negative components and satisfying some set of linear inequalities, called* constraints*. Specifically, if $(a_{ij})_{1\le i\le n 1\le j\le m}$, is a matrix of real numbers and $b \in \mathbb{R}^m$, then the following is a linear programming problem:*

$$\begin{array}{lccccccccc}
\min_x & c_1x_1 & + & c_2x_2 & + & \cdots & + & c_nx_n & & \\
s.t. & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & \le & b_1 \\
 & a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & \le & b_2 \\
 & \vdots & & \vdots & & & & \vdots & & \vdots \\
 & a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & \le & b_m \\
 & x_1, & & x_2, & & \ldots, & & x_n & \ge & 0.
\end{array}$$

Variables x_i are called decision variables*, matrix $A = (a_{ij})$ is the* constraint matrix*, the c_i are the* cost coefficients*, and $z = \sum_i c_i x_i$ is the objective function. Finally, the set of points $x \in \mathbb{R}^n$ satisfying all restrictions is called the* feasible set*.*

We can also express a linear problem in the following short form: $\max\{cx \mid Ax \leq b, x \geq \mathbf{0}\}$*. The term* continuous linear problem *may be used to distinguish this kind of problem from the integer linear problems presented in Section 1.2.*

Frequently, we will use the phrase *formulate a problem* for talking about the process of finding a suitable formulation for a given real-world problem. Albeit we will henceforth only consider maximisation problems, equivalent results may be obtained for minimisation problems by making minor adjustments to our discourse. The key for changing between minimisation and maximisation problems is realising that the minimum of $\sum_i c_i x_i$ equals minus the maximum of $\sum_i -c_i x_i$. This very insight of multiplying by -1 allows us to use $\geq$ inequalities in linear formulations, for $\sum_i a_{ij} x_i \geq b_j \iff \sum_i -a_{ij} x_i \leq -b_j$. Equality restrictions may now be introduced, being equivalent to the inclusion of two reverse inequalities.

Efficient methods exist for solving the general linear programming problem, the first discovered one being the Simplex Method. Being acquainted with the inner workings of these algorithms is not a prerequisite for this work. However, we direct the interested reader to the comprehensive introduction to the Simplex method given in the first five chapters of [Bazaraa et al., 2009].

1.1.3 Duality theory

A remarkable fact of linear programming is that any linear problem has an associated *dual* problem. Solving this dual problem gives us then the solution to our original problem. Furthermore, we may use this dual formulation so as to obtain useful information regarding our original problem. We will call our original problem *primal*. If the primal is of the form

$$\begin{array}{lll} \text{(P)} & \min & cx \\ & \text{s.t.} & Ax \geq b \\ & & x \geq \mathbf{0}, \end{array}$$

then we define its dual as

$$\begin{array}{lll} \text{(D)} & \max & ub \\ & \text{s.t.} & uA \leq c \\ & & u \geq \mathbf{0}. \end{array}$$

Please note that the dual has as many variables as the primal has restrictions, and vice versa. Also, cost vector and restriction vector exchange their places. Through the introduction of the appropriate modifications, any formulation may be taken into the form (P). Therefore, we may transform a general linear problem to the standard form (P) and then apply the transform from (P) to (D) to obtain a dual for the original problem. The following result partly justifies the use of the term *dual*:

Proposition 1. *The dual of a dual problem is the original primal problem.*

As stated, duality allows us to derive a wealth of useful results for solving linear problems. We will present, in particular, a result characterising the optimal solutions of a problem and its dual which will be used in Subsection 2.1.3. This result has been adapted from [Sierksma and Zwols, 2015], theorems 4.2.4 and 4.2.5.

Proposition 2. *A primal problem has an optimal solution iff its dual problem does. In this case, feasible solutions x and u for primal and dual are optimal iff $cx = ub$.*

The direct implication is known as the *strong duality theorem*. In particular, we obtain the following result for problems of the form (P):

Corollary 1. *x and u are optimal solutions for problems* (P) *and* (D)*, respectively, iff*

$$\begin{aligned} cx &= ub \\ Ax &\leq b \\ uA &\leq c \\ x &\geq \mathbf{0} \\ u &\geq \mathbf{0} \end{aligned}$$

1.2 Integer programming

1.2.1 Definition

Recall we defined a linear programming problem as being of the form $\min\{cx \mid Ax \leq b, x \geq \mathbf{0}\}$. If we also add the requirement that some of its variables be integer, we get an integer programming problem. The following definitions will make this more precise, and introduce some terminology for distinguishing between integer problems.

Definition 2 (Integer formulation)**.** *By a* (linear) integer problem*, we mean a linear problem with the added restriction that all of its variables be integers. Symbolically, these problems take the form*

$$\begin{array}{ll} \min_x & cx \\ s.t. & Ax \leq b \\ & x \geq \mathbf{0} \\ & x \in \mathbb{Z}. \end{array}$$

Definition 3 (Mixed integer formulation)**.** *By a* mixed integer problem*, we mean a linear problem with the added restriction that some of its variables be integers. Symbolically, these problems take the form*

$$\begin{array}{ll} \min_x & cx + dy \\ s.t. & Ax + By \leq b \\ & x, y \geq \mathbf{0} \\ & x \in \mathbb{Z}. \end{array}$$

Definition 4 (Binary integer formulation)**.** *By a* mixed integer problem*, we mean a linear problem with the added restriction that all of its variables take the values 0 or 1. Symbolically, these problems take the form*

$$\begin{array}{ll} \min_x & cx \\ s.t. & Ax \leq b \\ & x \in \{0,1\}^n. \end{array}$$

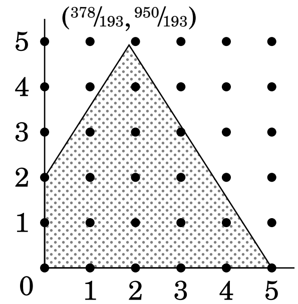

Figure 1.1: Comparison of solutions of an integer problem and its linear relaxation

1.2.2 Computational complexity

Even though integer programs look formally very similar to continuous linear programs, their resolution is computationally much more complicated. Intuitively, one might argue that a good heuristic might be to eliminate the integrity constraints, solve the resulting linear problem, and then round up the result. However, this naive procedure leads to very poor approximations. As an example, consider the following integer problem taken from [Wolsey, 1988]:

$$
\begin{array}{lrcrcr}
\max & 1.00x_1 & + & 0.64x_2 & & \\
\text{s.t.} & 50x_1 & + & 31x_2 & \leq & 250 \\
 & 3x_1 & - & 2x_2 & \geq & -4 \\
 & x_1, & & x_2 & \geq & 0 \\
 & x_1, & & x_2 & \in & \mathbb{Z}.
\end{array}
$$

Its feasible region is drawn in Figure 1.1, alongside the $\mathbb{Z}^2$ point cloud. We can see that the linear programming solution $(^{376}/_{193}, ^{950}/_{193})$ lies quite far from the solution to the integer problem, $(5, 0)$. However discouraging Figure 1.1 may be, it suggests a concept of great relevance in the resolution of integer problems:

Definition 5 (Linear relaxation)**.** *The* linear relaxation *of integer problem* $\min\{cx \mid Ax \leq b, x \geq \mathbf{0}, x \in \mathbb{Z}^n\}$ *is the linear problem* $\min\{cx \mid Ax \leq b, x \geq \mathbf{0}\}$.

1.2.3 Branch and bound

The most commonly used method for solving integer linear programming problems is *branch and bound*, also called *implicit enumeration*. This method searches for a problem's integer solution by dividing its feasible region into smaller subregions. The linear relaxation of the original problem is then solved over those subregions, until eventually an integer feasible solution is found. Furthermore, the optimal values of the linear relaxations can be compared to the objective value of the current best integer solution, thus being able to prune the search on those subregions unable to improve the current solution. More specifically, for a given integer minimisation problem (P), *branch and bound* solves (P) by building a tree in the following manner (adapted from [Sierksma and Zwols, 2015, Algorithm 7.2.1]):

---

1. **Initialisation**. First, an initial node (P_1) is defined and associated to (P)'s linear relaxation. This relaxation is then solved, and its solution $\tilde{x}^1$ and objective value $\tilde{z}^1$ are added to node (P_1). Furthermore, we define $z_U = -\infty$ and mark (P_1) as *unvisited*.

2. **Main loop**. Then, the following steps are repeated:
   (a) **Stopping rule**. If there are no more unvisited nodes and there is no current best solution, then the problem is unfeasible. If there are no more unvisited nodes, but there is some current best solution, then that solution is optimal. Otherwise, continue.
   (b) **Node selection**. Select some unvisited node ($\mathrm{P_k}$) and mark it as *visited*.
   (c) **Bounding**. Let z_L be the optimal value of the linear relaxation of ($\mathrm{P_k}$)'s father. If z_L is higher than z_U, then further studying this node cannot lead to any improvement on z_U. Thus, stop considering this node by going back to step 2.a.
   (d) **Solution**. Determine an optimal solution $\tilde{x}$ and the optimal value $\tilde{z}$ of the linear relaxation of node ($\mathrm{P_k}$). Then,
      i. If ($\mathrm{P_k}$) has no optimal solution, or $\tilde{z} \geq z_U$, prune this node by going back to step 2.a.
      ii. If $\tilde{z} < z_U$ and $\tilde{x}$ is an integer vector, then it is considered our current best solution, and we set $z_U = \tilde{z}$.
      iii. If $\tilde{z} < z_U$, but $\tilde{x}$ has some non-integer components, then go to step 2.e.
   (e) **Branching**. Select some non-integer component of $\tilde{x}$, say $\tilde{x}_i$, and define nodes ($\mathrm{P_{k_1}}$) and ($\mathrm{P_{k_2}}$) labelling them with the linear problem of ($\mathrm{P_k}$) alongside restriction $x_i \leq \lceil \tilde{x}_i \rceil$ for ($\mathrm{P_{k_1}}$) and $x_i > \lceil \tilde{x}_i \rceil$ for ($\mathrm{P_{k_2}}$).

Note that some steps were not perfectly defined. Namely, the node selection phase of 2.b and the component selection in the branching phase 2.e. These are important to tune, for choosing appropriate selection rules may result in much faster executions.

1.2.4 Branch and cut

As we have seen, the efficiency of branch and bound critically depends on the quality of the lower and upper bounds it can find for the objective function. Therefore, a natural question to ask is whether it would be beneficial to improve the formulation corresponding to a node prior to investigating its children. If this were done, the children's formulations would be tighter, and thus, in principle, have better bounds for their objective functions. This is the idea leading to branch and cut, a method extending the base branch and bound.

After finding each fractional solution, and prior to expanding a node, we may try to obtain an inequality that is fulfilled by each feasible solution, but not by this fractional solution. Such an inequality is said to *separate* the fractional solution from the feasible set, as the following definitions specify:

Definition 6 (Valid inequality). *An inequality is said to be a valid inequality for a set X if it is satisfied by all elements of X.*

Definition 7 (Separating inequality). *Given a point x_0 not in a set X, a valid inequality for X is said to separate x_0 from X if it is not satisfied by x_0.*

1.2.5 Useful results

As already stated, solving a general integer problem may be quite hard. However, some special classes of integer problems verify that their solutions are precisely those of their linear relaxations. One specially important such class is that with *totally unimodular* constraint matrices. The next definition specifies which matrices satisfy this property.

$$\begin{pmatrix} 1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

Figure 1.2: The left matrix is totally unimodular; the right one is not

Definition 8 (Total unimodularity)**.** *We say a matrix is* totally unimodular *(TU), if all of its square submatrices have determinant +1, -1 or 0.*

By *submatrix* of a matrix A we mean here the matrix resulting from eliminating from A any number of rows and/or columns. A *square submatrix* is then a square matrix thus obtained.

Since single matrix elements can be regarded as square submatrices of size one, totally unimodular matrices must have +1, -1 or 0 for entries. However, this simple condition is not a necessary one, as Figure 1.2 illustrates. Whereas the left matrix may be checked to be TU, the right one has determinant 2, and thus is not TU. We may now present a characterisation of total unimodularity which can be found in [Nemhauser and Wolsey, 1988], as well as a lemma necessary for its proof:

Lemma 1. *Let A be a totally unimodular matrix, and b,b',d,d' be vectors with integer coordinates. If the set $P(d,d',b,b') = \{x \in \mathbb{R}^n \mid b \leq Ax \leq b', d \leq x \leq d'\}$ is not empty, then its vertices have integer coordinates.*

Proposition 3. *A matrix A is totally unimodular iff every subset of rows J can be partitioned into two sets J_1 and J_2 such that*

$$\left| \sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij} \leq 1 \right|$$

for all $1 \leq i \leq n$.

Proof. ($\Longrightarrow$) This implication will be proved through an application of Lemma 1. With this aim, let us consider J as an arbitrary set of columns of A, $J \subseteq \{1,\ldots,n\}$. We will define several integral vectors from J and A in the following manner: z is a binary vector defined by $z_j = 1$ iff $j \in J$; d is the zero vector $\mathbf{0}$; $d' = z$; $g = Az$; and, finally, b and b' are defined by:

$$b_i = \begin{cases} \frac{g_i}{2}, & g_i \text{ is even} \\ \frac{g_i - 1}{2}, & \text{otherwise} \end{cases}, \quad b'_i = \begin{cases} \frac{g_i}{2}, & g_i \text{ is even} \\ \frac{g_i + 1}{2}, & \text{otherwise} \end{cases}$$

If we define $P = P(d,d',b,b')$ as in Lemma 1, then P is nonempty, for $\frac{z}{2} \in P$. Therefore, Lemma 1 allows us to conclude that P has integer vertices. Therefore, we may choose a binary vector $x \in P$ with $x_j = 0$ for all columns j not in J, and $x_j \in \{0,1\}$ for all other j. Furthermore, since $x_j = 0 \Longrightarrow z_j = 0$, it holds that $z_j - 2x_j \in \{+1,-1\}$ for all j. Defining $J_1 = \{j \in J \mid z_j - 2x_j = +1\}$ and $J_2 = \{j \in J \mid z_j - 2x_j = -1\}$, we get

$$\begin{aligned} \sum_{j \in J_1} a_{ij} + \sum_{j \in J_2} a_{ij} &= \sum_{j \in J} a_{ij}(z_j - 2x_j) \\ &= \sum_{j \in J} a_{ij} z_j - 2 \sum_{j \in J} a_{ij} x_j \\ &= (Az)_i - 2(Ax)_i \\ &= g_i - 2(A_x)_i \\ &= (*) \end{aligned}$$

If g_i is even, $b_i = b'_i = \frac{g_i}{2}$, and therefore $b_i \leq (Ax)_i \leq b'_i$ implies $(*) = 0$. If g_i is odd, the same relation implies $(*) \in \{-1, 0, +1\}$. Therefore,

$$\left| \sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij} \right| \leq 1$$

for every row i.

($\Longleftarrow$) We will prove by induction in k that every $k \times k$ nonsingular matrix of A has determinant $r \in \{+1, -1\}$. For $k = 1$ the result is trivial. Let us now suppose that every $(k-1) \times (k-1)$ nonsingular submatrix of A has determinant ± 1, let B be a $n \times n$ nonsingular submatrix, and let $r = |\det B|$.

An elementary result from linear algebra assures us that $B^{-1} = \frac{B^*}{r}$, where B^* represents B's adjoint. Then, $Bb_1^* = re_1$. If $J = \{i \mid b_{i1}^* \neq 0\}$, $\tilde{J}_1 = \{i \in J \mid b_{i1}^* = 1\}$ and $\tilde{J}_2 = J - \tilde{J}_1$, then $b_1^* = re_1$ translates into

$$(Bb_1^*)_i = \sum_{j \in \tilde{J}_1} b_{ij} - \sum_{j \in \tilde{J}_2} b_{ij} = 0, \quad \forall i \in \{2, \ldots, k\}.$$

This implies that, for any $i = 2, \ldots, k$, J holds an even number of j such that $b_{ij} = 0$. Therefore, $\sum_{j \in J_1} b_{ij} - \sum_{j \in J_2} b_{ij}$ is even for any partition $\{J_1, J_2\}$ of J. However, by hypothesis, there exists a partition $\{J_1, J_2\}$ with

$$|\sum_{j \in J_1} b_{ij} - \sum_{j \in J_2} b_{ij}| <= 1.$$

Therefore,

$$\sum_{j \in J_1} b_{ij} - \sum_{j \in J_2} b_{ij} = 0, \quad \forall i \in \{2, \ldots, k\}$$

for this partition. Let us now define $\alpha = |\sum_{j \in J_1} b_{1j} - \sum_{j \in J_2} b_{1j} = 0|$. If $\alpha = 0$, then we could define $y \in \mathbb{R}^k$ by $y_i = 1$ for all $i \in J_1$, $y_i = -1$ for all $i \in J_2$, and $y_i = 0$ for all other i. Then, $By = 0$. Since B is nonsingular, $y = 0$, which contradicts $J \neq \emptyset$. Therefore, it must be that $\alpha = 1$, and $By \in \{+e_1, -e_1\}$. Since $Bb_1^* = re_1$,

$$B(\pm y) = B(\frac{b_1^*}{r}) \implies \pm y = \frac{b_1^*}{r}.$$

Taking into account that b_1^* and y have entries $\{0, +1, -1\}$, it follows that $|r| = 1$, as desired. □

Finally, we may formally state the main result about totally unimodular matrices, which can also be found in [Nemhauser and Wolsey, 1988]:

Proposition 4. *If A is totally unimodular, b has integer components, and the linear problem* $\min\{cx \mid Ax \leq b, x \geq \mathbf{0}\}$ *has optimal solutions, then these are integer.*

1.3 Bilevel programming

Bilevel programming was first introduced in 1934 by H. von Stackelberg in relation to the modelling of financial markets [von Stackelberg, 1934]. Within the Game Theory literature, von Stackelberg's special case led to the so-called *Stackelberg's games*. Thus, this field of optimisation bears witness to the fruitful consequences of interaction between mathematics and other sciences.

Although a description of bilevel programming formulations may be given for the more general nonlinear optimisation case, we will here restrict ourselves to linear integer bilevel formulations. For the general description, please consult Dempe's comprehensive monograph on Bilevel Programming [Dempe, 2002], on which this section is based.

A bilevel program is a maximisation problem in two sets of variables, x and y, and hierarchically divided in two subproblems. For any value of y, there is an ancillary problem of the form:

$$\begin{aligned} \max_x \quad & c_{21}y + c_{22}x \\ \text{s.t.} \quad & A_{21}y + A_{22}x \le b_2 \\ & x \ge \mathbf{0}. \end{aligned}$$

Here, c_{21}, c_{22} and b_2 represent vectors of the appropriate dimensions, and A_{21} and A_{22} represent matrices. We will need to require for this family of ancillary subproblems to have an unique solution $x(y)$ for every y. Note that, on every particular bilevel formulation, this is a supposition that will need to be checked. The master subproblem takes the following form:

$$\begin{aligned} \max_y \quad & c_{11}y + c_{12}x(y) \\ \text{s.t.} \quad & A_{11}y + A_{12}x(y) \le b_1 \\ & y \ge \mathbf{0}. \end{aligned}$$

This framework naturally fits situations in which two agents interact, the benefits obtained or costs incurred by an agent being determined by choices taken by a second agent. In the case of von Stackelberg, these agents were decision makers in some market. In our work, the bilevel framework will serve for modelling the interaction of a company offering some goods and a set of buyers interested in certain subsets of those goods.

1.4 Introduction to the Rank Pricing Problem

1.4.1 Definition and notation

In the Rank Pricing Problem (henceforth, RPP), a company offers a finite set I of products to a finite set K of customers. Each customer wants to buy one product from this set, having some strict preferences among products. More specifically, each client $k \in K$ wants to buy one element from the subset $S^k \subseteq I$. Each $i \in S^k$ has an associate preference value s_i^k, such that k would rather buy a product with the greatest possible preference value. Preferences are strict, in the sense that for any $i, j \in S^k$ with $i \neq j$, either $s_i^k < s_j^k$ or $s_i^k > s_j^k$ hold, but not $s_i^k = s_j^k$. Each client also has an associated budget, so that k can only acquire those products with a price below or equal to this budget. We suppose the company has an unlimited supply of each product, so that every $i \in I$ may be bought by multiple clients. The company's problem is then to assign a nonnegative real price p_i to each of its products so as to maximise its revenue.

Note that there is a trade off between low and high prices: if the prices are very high, the company will get a notable profit from each item, but few customers will be able to afford buying any product, and thus few items will be sold. On the other hand, if prices are very low, many customers will be able to buy some item, but the obtained revenue will be quite meagre. Note also that this problem is naturally phrased as two interrelated problems: the company wants to maximise its revenues by pricing its products, and customers want to buy a maximally preferred product of affordable price. This line of thought will be the starting point of Chapter 2.

We will now specify the notation we will use in this work. For a summary of this notation, please consult the nomenclature table at the begin of this document. $K = \{1, \dots, |K|\}$ will stand for the finite set of customers, and $I = \{1, \dots, |I|\}$ for the finite set of offered products. For any $k \in K$, $S^k \subseteq I$ will be the set of products that k is willing to buy. For any $i \in I, k \in K$, $s_i^k \in I$ will stand for the preference value of product i by k. $B = \{b^1, \dots, b^M\}$, with $M \le |k|$, will be the set of possible budgets ordered like

$b^1 < b^2 < \cdots < b^M$. Function $\sigma : K \to \{1,\ldots,M\}$ will assign to each client the index of its budget. That is, the budget of customer k is $b^{\sigma(k)}$. Note that clients k with a greater value of $\sigma(k)$ are the richest.

We may assume, without loss of generality, that $S^k \neq \varnothing$ for all $k \in K$. Otherwise, we could simply remove k from the problem. Similarly, we may assume each product $i \in I$ is in the preference list of at least one customer. Otherwise, it may simply be removed from the problem. We will furthermore define $b^0 = 0$, for it will be useful in the following chapters.

1.5 Graph Theory and its application to the Set Packing Problem

1.5.1 The Set Packing Problem

Now must we present some concepts on Graph Theory and Geometry that will be needed during our discussion in Chapter 4. We will firstly introduce one of the most extensively studied integer programming problems:

Definition 9 (Set Packing Problem)**.** *If $c \in \mathbb{R}^u$, $c \geq \mathbf{0}$ is a cost vector, A a $w \times u$ binary matrix and $\mathbf{1}_w \in \mathbb{R}^w$ a vector of ones, then we may define the Set Packing Problem (SPP) as the binary maximisation problem*

$$\begin{aligned} \max_t \quad & ct \\ & At \leq \mathbf{1}_w \\ & t_i \in \{0,1\}, \forall 1 \leq i \leq u. \end{aligned}$$

In order to give an interpretation to this formulation, let us consider a set X with u elements. We would like to select some elements in this set so that w restrictions be fulfilled. Namely, the i'th of these restrictions states that only one element in a certain subset may be chosen. This subset is defined by the binary vector a_i, so that at most one element among those with positive entries in a_i may be selected. By identifying A with the matrix with rows a_i so defined, this set element selection problem corresponds to the SPP above defined.

1.5.2 Elementary definitions from Graph Theory

In Chapter 4, we will be using some concepts from graph theory. These are presented below.

Definition 10 (Graph)**.** *We define a* graph*, or, more specifically, an* undirected graph*, as a pair of finite sets $G = (V,E)$, where E is a family of 2-element subsets of V. V is called the set of* vertices*, or* nodes*, of G, and E the set of* edges *of G. Elements of E are usually represented as (u,v), instead of $\{u,v\}$, where $u,v \in V$. If $(u,v) \in E$, then we say that nodes u and v are* adjacent*.*

Definition 11 (Subgraph)**.** *We say that a graph $F = (U,I)$ is a* subgraph *of graph $G = (V,E)$ if $U \subseteq V$ and $I \subseteq E$. Furthermore, if $U \neq V$ or $I \neq E$, then we say F is a* proper *subgraph of G.*

Definition 12 (Complete graph)**.** *We say that a graph is* complete *if every pair of nodes on it are adjacent.*

Definition 13 (Neighbourhood)**.** *The neighbourhood of a node in a graph is the set of all nodes adjacent to it.*

Definition 14 (Clique)**.** *A subgraph G' of a graph G is said to be a* clique *if it is complete, and is not a proper subgraph of any other complete subgraph of G.*

1.5.3 Application to the SPP

Finally, our discussion in Chapter 4 will employ some results linking Graph Theory with linear programming problems. Before presenting the main theorem linking these two fields, some definitions must be presented.

Definition 15 (Intersection graph)**.** *We define the* intersection graph *of a set packing problem instance* $\max_t\{ct \mid At \leq \mathbf{1}_w, t \in \{0,1\}^u\}$ *to be the graph* $G = (V,E)$ *whose nodes are the problem variables* $t_i, 1 \leq i \leq u$*, and in which nodes* t_i *and* t_j *are adjacent iff* $a_{ki} = a_{kj} = 1$ *for some row* k *of* A*.*

That is, any two nodes in the intersection graph are adjacent iff the corresponding variables are not allowed to be simultaneously set to 1 in a feasible solution of the SPP instance.

Definition 16 (Incidence vector)**.** *If* $V' \subseteq V$ *is a set of nodes of a graph* $G = (V,E)$*, then we define its* incidence vector *to be* $(t_1,\ldots,t_{|V|})$*, where the* t_i *are binary values with* $t_i = 1$ *iff the i'th node of* G *belongs to the set* V'*.*

Definition 17 (Convex hull)**.** *The convex hull of a set of points in* $\mathbb{R}^n$ *is the intersection of all convex sets containing that set.*

Definition 18 (Facet)**.** *Let* X *be a subset of* $\mathbb{R}^n$ *implicitly defined as* $At \leq b$, $t \in \mathbb{R}^n$*. Then, by* facet *of* X *we refer to a linear constraint defining an* $(n-1)$*-dimensional face of the* $At \leq b$ *polyhedron.*

Lastly, the prime result over which Chapter 4's study is based is the following, first given in [Padberg, 1973]:

Theorem 1. *Let* (S) be a set packing problem, G its intersection graph, and $P(G)$ the convex hull of all its incidence vectors (that is, the convex hull of its feasible solutions). Then, an inequality $\sum_{j \in J} t_j \leq 1$ is a facet of $P(G)$ iff the variables t_j, $j \in J$ form a clique of G.

Chapter 2

Formulations for the Rank Pricing Problem

As said in the introduction, the resolution of linear programming problems requires developing good formulations. Such a requirement is this chapter's main motivation, driving the description of its formulations, each one being a small improvement over the previous one. We will build on the introduction to the RPP given in Section 1.4, and follow sections 2 and 3 of [Calvete et al., 2019], from which the formulations of this chapter originate. The first formulation, (BNL^p) , is a conceptually natural treatment of the problem, explicitly representing the bilevel structure of the RPP as an interaction of a seller pricing the items and several customers buying them. From (BNL^p) we obtain by introducing a new set of decision variables the formulation (BNL^v) .

The introduction of both bilevel formulations will aid the reader to get acquainted with the RPP. Furthermore, the second of these formulations, (BNL^v) , will also be the starting point for obtaining the single level formulations (BNL) and (SLNL) .

2.1 Bilevel formulation

2.1.1 Formulation with variables p_i

Recall that in the RPP a seller produces some finite set of products, I, and wants to price them so as to maximise the gains he will obtain through their sale. These products are in turn bought by some customers, which are represented as elements of a finite set K, trying to acquire the most preferred items lying within their budget. It is thus conceptually natural to model the RPP as a bilevel problem, so that each level represents one of the mentioned types of actors, seller and client. On the upper level, the seller prices the items, maximising his gains. On the lower level, each client tries to acquire his most preferred item, always respecting his budget. Note that these two problems are interrelated: by pricing the products, the seller determines which items fall within each client's budget. Likewise, by purchasing one item or another, the clients determine the seller's profits.

Turning now to our first formulation, $i \in I$ will represent an arbitrary item and $k \in K$ an arbitrary product. Variable p_i is then a real positive value representing i's price. These p_i will be the decision variables of the upper-level problem, for their value is decided by the seller. We furthermore need to consider binary variables x_i^k, determining that customer k acquires item i iff $x_i^k = 1$. As said in the introduction, s_i^k indicates the preference level of client k for product i, being $s_i^k = 0$ iff k does not want to buy i in any case. Lastly, $b^{\sigma(k)}$ will take the value of k's budget. With this notation, formulation (BNL^p) then the form:

$$
\begin{aligned}
(\mathrm{BNL^p}) \quad \max_p \quad & \sum_{k\in K}\sum_{i\in S^k} p_i x_i^k && (2.1)\\
\text{s.t.} \quad & p_i \geq 0, \quad \forall i \in I && (2.2)
\end{aligned}
$$

where $\forall k \in K$, $x^k = (x_i^k)_{i\in I}$ is an optimal solution of the problem:

$$
\begin{aligned}
\max_{x^k} \quad & \sum_{i\in S^k} s_i^k x_i^k && (2.3)\\
\text{s.t.} \quad & \sum_{i\in S^k} x_i^k \leq 1 && (2.4)\\
& \sum_{i\in S^k} p_i x_i^k \leq b^{\sigma(k)} && (2.5)\\
& x_i^k \in \{0,1\}, \forall i \in S^k && (2.6)
\end{aligned}
$$

In this formulation, (2.1) is the sum of prices of each product acquired, and thus represents the seller's total revenue. (2.2) state that prices are nonnegative real variables. In the lower level problem, (2.6) state that the x_i^k are binary variables, and (2.4) ensures that client k purchases at most one item. The left hand side of (2.5) then takes the value of the price of the item i acquired by x, so that the inequality restricts the bought items to be among those with price falling within k's budget. The meaning of (2.3) becomes then clear: it is the preference level of the item bought by k. Thus, maximising this sum, we assure that $x_i^k = 1$ just for the i most preferred by k and falling within his budget.

However, in order to prove that this formulation is well-posed, we need to assure that the lower level problems have a unique solution. Suppose, for instance, that for some customer k the maximum of the lower level problem were attained at two different vectors, x^k and $\tilde{x}^k$. This corresponds to k being able to select two most-preferred products for a given assignment of prices p. This would be problematic, for the employment of x^k and $\tilde{x}^k$ would give rise to different values for the upper level objective function (2.1). Next result assures that this situation cannot arise.

Proposition 5. *The lower level problems of formulation* (BNL^p) *have a unique optimal solution.*

Proof. The proof consists on explicitly giving the optimum value of the x^k vector, and noting that this value is unique. Let us consider for this matter the lower level problem associated to some customer $k \in K$ and price assignment p. If $p_i > b^{\sigma(k)}$ for all products $i \in I$, then no x_i^k could take a value of 1, for then (2.5) would not hold. Thus, $x_i^k = 0, \forall i \in S^k$ is the only solution in this case. Otherwise, constraint (2.4) asserts that at most one x-variable x_i^k may be non-zero, and the optimal solution is then given by $x_j^k = 0$ for all $j \in S^k$ except for the unique i such that $s_i^k = \max\{s_j^k \mid j \in S^k, p_j \leq b^{\sigma(k)}\}$. Note that the uniqueness of solution directly depends on the product preferences s_i^k being all distinct. □

2.1.2 Formulation with variables v_i^ℓ

Formulation (BNL^p) contains both integer (the x_i^k) and real (the p_i) variables. However, the possible values for the p_i may be restricted, so that only a finite set of prices must be considered, thus converting the problem to a purely integer one. The key insight for this was noted by the authors of [Rusmevichientong et al., 2006], and we phrase it here as a lemma:

Lemma 2. *There exists solutions of* (BNL^p) *with* $p_i \in B, \forall i \in I$.

Proof. This proof is adapted from that of [Domínguez et al., 2022]. Suppose that there were an optimal p and a product $i \in I$ such that p_i were strictly between two prices in B. That is, $b^m < p_i < b^{m+1}$ for some $1 \le m \le M-1$. If in this solution no client buys i, then we may change its price as we please, and thus obtain a valid optimal solution where $p_i = b^{m+1}$.

We will prove by contradiction that the reverse case (some client buys i) cannot arise. If such a client existed, we could increase p_i up to b^{m+1}, and the resulting price assignment would leave invariant the sets of products that customers can afford. Indeed, all clients $k \in K$ with $\sigma(k) \le m$ could not afford i before the change, and cannot afford i after it. Likewise, all $k \in K$ with $\sigma(k) \ge m+1$ could afford i before the change, and after it they still do. Therefore, the increase in p_i is possible. However, this would imply obtaining a feasible solution with an optimal value strictly greater than that of the original optimal solution, which is absurd. □

Therefore, the set of product prices may be chosen to equal the set of client budgets. We will exploit this remark by defining a new set of binary variables, $v_i^\ell, \forall i \in I, \forall \ell \in \{1,\dots,M\}$, which will represent the price assigned to the i'th item, thus replacing the continuous p_i variables. Namely, $v_i^\ell = 1$ iff product i is assigned price b^ℓ. Using these variables, we can replace (2.2) with

$$\begin{aligned} &\sum_{\ell=1}^{M} v_i^\ell \le 1, && \forall i \in I, \\ &v_i^\ell \in \{0,1\}, && \forall i \in I, \ell \in \{1,\dots,M\} \end{aligned}$$

The first constraints assert, in the same manner as (2.4), that each product can be assigned at most one price. The second constraints assert that the v_i^ℓ are binary variables. We also need to replace constraint (2.5) of the lower level problem, restricting the price of the products affordable by k, with the following constraint:

$$x_i^k \le \sum_{\ell=1}^{\sigma(k)} v_i^\ell, \quad \forall k \in K, i \in S^k$$

This constraint asserts that x_i^k may only take a value of 1 if $v_i^\ell = 1$ for some ℓ such that $1 \le \ell \le \sigma(k)$. Since $b^{\sigma(k)}$ is k's budget, and $\ell_1 < \ell_2$ implies $b^{\ell_1} < b^{\ell_2}$, this means k can only buy i if item i gets assigned a price b^ℓ less than or equal to its budget.

Through these substitutions, we arrive at a purely integer programming formulation, which we will denote by (BNL^v) :

$$\begin{aligned} (\text{BNL}^{\text{v}}) \quad \max_v \quad & \sum_{k\in K} \sum_{i\in S^k} (\sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell) x_i^k && (2.7) \\ \text{s.t.} \quad & \sum_{\ell=1}^{M} v_i^\ell \le 1, \quad \forall i \in I && (2.8) \\ & v_i^\ell \in \{0,1\}, \quad \forall i \in S^k, \ell \in \{1,\dots,M\} && (2.9) \end{aligned}$$

with x^k being solutions to the following problem:

$$\max_{x^k} \quad \sum_{i \in S^k} s_i^k x_i^k \tag{2.10}$$

$$\text{s.t.} \quad \sum_{i \in S^k} x_i^k \leq 1 \tag{2.11}$$

$$x_i^k \leq \sum_{\ell=1}^{\sigma(k)} v_i^\ell, \quad \forall k \in K, i \in S^k \tag{2.12}$$

$$x_i^k \in \{0,1\}, \quad \forall i \in S^k. \tag{2.13}$$

Other than the changes already mentioned and leading to constraints (2.8), (2.9) and (2.12), (BNL^v) distinguishes itself from (BNL^p) by its upper-level cost function (2.7). This is obtained from (2.1) by formally substituting p_i by the term $\sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell$. For each k, this sum equals the price given to the i'th item, if k can afford it, and 0 otherwise.

Changing from p to v variables leads to an interesting consequence. As noted in the paper [Calvete et al., 2019], the lower level of (BNL^v) can be incorporated to the upper level problem as a group of constraints, thus converting (BNL^v) in a single-level formulation. The specific way in which this may be archived will be explained in the next section.

2.1.3 Single level formulation

One important remark allowed the authors of [Calvete et al., 2019] to eliminate the lower level problem of (BNL^v) . This remark was the unimodularity of the matrix associated to these problems. As noted in the introductory chapter, if an integer formulation has a totally unimodular associated matrix, then the solution of its linear relaxation has integer coefficients, thus being possible to solve the integer problem as a linear one. The author has developed the following proof, which assures that this is indeed the case of our lower level problems:

Lemma 3. *The lower level problems of* (BNL^v) *have totally unimodular associated matrices.*

Proof. We note that the constraint matrix A takes in this case the form

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

We may then apply result 3 for concluding that A is totally unimodular. Indeed, for any set of columns J, we may choose a partition J_1, J_2 with cardinalities $|J_1|$ and $|J_2|$ with a difference of at most one unit. Then,

- For $i = 1$,

$$\left| \sum_{j\in J_1} a_{1j} - \sum_{j\in J_2} a_{1j} \right| = ||J_1| - |J_2|| \leq 1$$

- For $i \neq 1$, row $a_{i\cdot}$ contains exactly one non-zero entry, and thus

$$\left| \sum_{j\in J_1} a_{1j} - \sum_{j\in J_2} a_{1j} \right| = 1$$

□

This result assures us that constraints (2.12) may be relaxed to $0 \leq x_i^k \leq 1, \forall i \in S^k$. Furthermore, once the prices (i.e. the v_i^ℓ variables) are fixed, the set of products affordable by each customer k can be explicitly calculated without resorting to linear constraints as $I(k) = \{i \in S^k \mid \sum_{l=1}^{\sigma(k)} v_i^l = 1\}$. These two remarks allow us to write the lower level problem in the following manner:

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

Note that it is not necessary to explicitly add the restriction $x_i^k \leq 1$, for this is already implied by $\sum_{i\in I(k)} x_i^k \leq 1$. Now, we may follow the concepts from duality theory presented in Subsection 1.1.3 in order to give for each $k \in K$ the dual problem of the lower-level problem:

$$\begin{array}{ll} \min_{u^k} & u^k \\ \text{s.t.} & u^k \geq s_i^k, \quad i \in I(k) \\ & u^k \geq 0. \end{array}$$

Duality theory now allows us to characterise the primal and dual problems' solutions x^k and u^k through the following conditions:

$$\sum_{i\in I(k)} s_i^k x_i^k = u^k \tag{2.14}$$

$$u^k \geq s_i^k \quad \forall i \in I(k) \tag{2.15}$$

$$\sum_{i\in I(k)} x_i^k \leq 1$$

$$x_i^k, u^k \geq 0.$$

Note that equation (2.14) and inequalities (2.15) may be combined into a single inequality not containing u^k, so that the system of inequalities results in a set of conditions over x^k alone. These relations hold the key to removing the lower-level problem from (BNL^v) and for obtaining the following formulation, for their inclusion in the upper level problem will result in x_k having the same meaning as that given by being solution to the lower level problem. We thus arrive to the following single-level formulation for the RPP:

$$\text{(BNL)} \quad \max_{v,x} \quad \sum_{k\in K}\sum_{i\in S^k}(\sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell)x_i^k \tag{2.16}$$

$$\text{s.t.} \quad \sum_{\ell=1}^{M} v_i^\ell \le 1, \quad \forall i \in I \tag{2.17}$$

$$\sum_{i\in S^k} x_i^k \le 1, \quad \forall k \in K \tag{2.18}$$

$$x_i^k \le \sum_{\ell=1}^{\sigma(k)} v_i^\ell, \quad \forall k \in K, i \in I \tag{2.19}$$

$$\sum_{j\in S^k} s_j^k x_j^k \ge s_i^k \sum_{\ell=1}^{\sigma(k)} v_i^\ell, \quad \forall k \in K, i \in S^k \tag{2.20}$$

$$v_i^\ell, x_i^k \in \{0,1\}, \quad \forall k \in K, i \in S^k, \ell \in \{1,\dots,M\} \tag{2.21}$$

The problem is now over two sets of variables, v and x. Cost function (2.16) is the same as (2.7), and constraints (2.17) correspond to (2.8). Conditions (2.18) and (2.19) correspond to (2.11) and (2.12) above. Finally, constraints (2.20) have been obtained from the conditions above mentioned, by combining (2.14) and (2.15) into a single inequality, so that explicit reference to u^k can be removed.

Thus, if some vectors v, x satisfy conditions (2.17) to (2.21), v will be a valid price assignment, and all x^k will verify (2.14) and (2.15), thus being a solution to the lower level problem of (BNL^p) . Therefore, BNL is a valid formulation for the RPP.

2.2 Alternative single level formulation

We will now use the concepts and terminology of Section 2.1 to introduce a new single level formulation for the RPP which will serve as the basis for the further development of this work. Firstly, we will need to define two families of sets which will be used in the new formulation.

Definition 19. *For any customer $k \in K$ and products $i, j \in S^k$, we say that i is k-better than j if k prefers product i over product j. We denote the set of products k-better than i as $B(k,i) = \{j \in S^k \mid j \text{ is } k\text{-better than } i\}$.*

Similarly, i is defined to be k-worse than j if k prefers product j over product i. The set of products k-worse than i is denoted by $\overline{B(k,i)} = \{j \in S^k \mid j \text{ is } k\text{-worse than } i\}$.

We may use these sets in the following manner. A customer k buys product i iff i's price is within k's budget, and all products j that are k-better than i are above k's budget. Using the price-indicator variables v_j^ℓ, this is equivalent to asserting that $x_i^k = 1$ iff

$$\sum_{\ell=1}^{\sigma(k)} v_i^\ell = 1 \text{ and } \sum_{\ell=1}^{\sigma(k)} v_j^\ell = 0, \forall j \in B(k,i).$$

This property lends itself towards defining constraints (4d) and (4e) in the following single level non-linear formulation (SLNL) :

$$
\begin{aligned}
\text{(SLNL)} \quad \max_{v,x} \quad & \sum_{k\in K}\sum_{i\in S^k}(\sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell)x_i^k && (2.22)\\
\text{s.t.} \quad & \sum_{i\in S^k} x_i^k \le 1, \quad \forall k\in K && (2.23)\\
& \sum_{\ell=1}^{M} v_i^\ell \le 1, \quad \forall i\in I && (2.24)\\
& x_i^k + \sum_{\ell=1}^{\sigma(k)} v_j^\ell \le 1, \quad \forall k\in K, i\in S^k, j\in B(k,i) && (2.25)\\
& x_i^k + \sum_{\ell=\sigma(k)+1}^{M} v_i^\ell \le 1, \quad \forall k\in K, i\in S^k && (2.26)\\
& v_i^\ell, x_i^k \in \{0,1\}, \forall k\in K, i\in S^k, \ell\in\{1,\dots,M\}. && (2.27)
\end{aligned}
$$

The objective function (2.22) is the same as that of (BNL) , (2.16). Restriction families (2.23) and (2.24) correspond, respectively, to (2.18) and (2.17). Restriction family (2.26) results of combining (2.19) and (2.17), and for a customer k and a product i, it means that if k buys i, then i's price is not higher than k's budget. Finally, constraint family (2.25) is obtained from the remark above, and means that if k buys i, then it does not buy any product j that is k-better than i.

Finally, the authors of [Calvete et al., 2019] noted that constraints (2.25) can be strenghened by considering the following family of constraints:

Proposition 6. *The following inequalities are valid constraints for the problem* (SLNL) *and dominate constraints* (2.25)*:*

$$\sum_{j\in\overline{B(k,i)}} x_j^k + \sum_{\ell=1}^{\sigma(k)} v_i^\ell \le 1, \quad \forall k\in K, i\in S^k \textit{ such that } \overline{B(k,i)} \neq \emptyset. \tag{2.28}$$

Proof. First, we will prove that constraints (2.28) are valid. i.e. that every solution to (SLNL) verifies them. If product i is within customer k's budget, then $\sum_{\ell=1}^{\sigma(k)} v_i^\ell = 1$. Furthermore, none product k-worse than i will be bought, and so $\sum_{j\in\overline{B(k,i)}} x_j^k = 0$. Thus, (2.28) holds in this case. If product i is not within k's budget, then $\sum_{\ell=1}^{\sigma(k)} v_i^\ell = 0$. Furthermore, at most one product k-worse than i will be bought, and so $\sum_{j\in\overline{B(k,i)}} x_j^k \le 1$. Thus, (2.28) always hold.

Secondly, we will prove that (2.28) dominate (2.25). For that, let $k\in K$, $i\in S^k$ and $j\in B(k,i)$, so that $x_i^k + \sum_{\ell=1}^{\sigma(k)} v_j^\ell \le 1$. Since j is k-better than i, i is k-worse than j, and thus $x_i^k \le \sum_{i'\in\overline{B(k,j)}} x_{i'}^k$. Therefore,

$$\sum_{j\in\overline{B(k,i)}} x_j^k + \sum_{\ell=1}^{\sigma(k)} v_i^\ell \le 1.$$

□

Chapter 3

Linealisation and strenghtening

3.1 Motivation and chapter structure

We have already said in the introduction that, despite being a highly expressive paradigm that allows solving a wide variety of problems, integer programming is computationally expensive. Thus, two formulations for the very same problem may give rise to completely different solving times. This was the reason for abandoning formulation (BNL^v) , which had a conceptually simple structure explicitly reflecting the two agents (seller and client) involved in the RPP , in favour of (SLNL) , which was computationally simpler due to being a single-level formulation. This chapter's objective is to continue presenting results in this line, following [Calvete et al., 2019], deriving from (SLNL) two further formulations dealing with its main caveat: non-linearity.

Being all restrictions in (SLNL) linear inequalities, the only element introducing non-linearity is the objective function (2.22), in which x_i^k variables get multiplied with v_i^ℓ. This is thus the element to improve; improvement done by defining a new set of variables allowing a linear expression for the cost function. Two such sets of variables will be presented in subsections 3.2.1 and 3.3.1, each one leading to a different formulation.

The new linear formulations will still leave room for improvement, their restrictions being too weak. The next results will thus be two families of valid restrictions (3.2.2) and (3.3.2), which will be, however, too large to be directly added to the formulations. Thus, in Section 3.4, an algorithm will be presented for dynamically adding them to the formulation during the problem's resolution, following the branch-and-cut schema outlined in the introductory chapter.

3.2 First single level linear formulation

3.2.1 Formulation

The function to linearise is the sum of the prices of all sold products. One way of proceeding is defining a new set of variables over which (2.22) is a linear function. In this case, we will use, for each client k, z^k, the benefit obtained from client k. Thus, the objective may be expressed as the sum of the benefits obtained from each client. i.e.,

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

Several restrictions will be added to the formulation that will relate z^k with the old variables in order to codify z^k's meaning. These restrictions will be the following:

$$
\begin{aligned}
z^k &\leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell + b^{\sigma(k)}\left(1 - x_i^k\right), \quad \forall k \in K, i \in S^k && (3.1)\\
z^k &\leq b^{\sigma(k)} \sum_{i \in S^k} x_i^k \quad \forall k \in K && (3.2)\\
z^k &\geq 0 \quad \forall k \in K. && (3.3)
\end{aligned}
$$

These inequalities codify z^k's meaning in the following manner: If k cannot acquire any product in its preference list, then the benefit from him obtained will be zero. If k acquires the (unique) product i, the benefit from him obtained will be i's price, $\sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell$. More concretely,

1. If k acquires no product, then $\sum_{i \in S^k} x_i^k = 0$, and (3.2) implies then $z^k \leq 0$, the desired value of z^k in this case. Furthermore, for all $i \in S^k$,

$$z^k \leq 0 \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell + b^{\sigma(k)}$$

   and thus restrictions (3.1) do not change z^k's meaning.

2. If k acquires product $i_0 \in S^k$, then $b^{\sigma(k)}\left(1 - x_{i_0}^k\right) = 0$, and (3.1) leads to

$$z^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_{i_0}^\ell.$$

   Thus, z^k is less than or equal to the price assigned to i_0. For any other $i \in S^k, i \neq i_0$, the restriction (3.1) is redundant $z^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell + b^{\sigma(k)} = b^{\sigma(k)}$, as well as (3.2): $z^k \leq b^{\sigma(k)} \sum_{i \in S^k} x_i^k = b^{\sigma(k)}$.

These constraints would suffice for defining the new formulation. However, a further improvement must be made on them by noting that (3.1) can be reinforced as follows:

Proposition 7. *For any $k \in K$, the following is a reinforcement of* (3.2)*:*

$$z^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell + b^{\sigma(k)} \sum_{j \in S^k : j \neq i} x_j^k, \quad \forall k \in K, i \in S^k. \tag{3.4}$$

Proof. The proof is done by a case analysis:

- If k buys no $i' \in S^k$, then $\sum_{j \in S^k : j \neq i} x_j^k = 0$ and the restriction results $z^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell$, thus being redundant, for in this case, (3.2) takes the form $z^k \leq 0$.

- If k buys some $i_0 \neq i$, then $\sum_{j \in S^k : j \neq i} x_j^k = 1$ and the restriction results $z^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell + b^{\sigma(k)}$, as the corresponding one in (3.1).

- If k buys precisely i, the restriction results $z^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell$, as the corresponding one in (3.1).

$\square$

Finally, the new objective function, variables, and constraints can be added to (SLNL) in order to obtain the following formulation, named (SLL$_1$) :

$$\max_{v,x,z} \quad \sum_{k\in K} z^k \tag{3.5}$$

$$\text{s.t.} \quad \sum_{i\in S^k} x_i^k \le 1, \quad \forall k \in K \tag{3.6}$$

$$\sum_{\ell=1}^{M} v_i^\ell \le 1, \quad \forall i \in I \tag{3.7}$$

$$\sum_{j\in \overline{B(k,i)}} x_j^k + \sum_{\ell=1}^{\sigma(k)} v_i^\ell \le 1, \quad \forall k \in K, i \in S^k : \overline{B(k,i)} \ne \varnothing \tag{3.8}$$

$$x_i^k + \sum_{\ell=\sigma(k)+1}^{M} v_i^\ell \le 1, \quad \forall k \in K, i \in S^k \tag{3.9}$$

$$z^k \le \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell + b^{\sigma(k)} \sum_{j\in S^k: j\ne i} x_j^k, \quad \forall k \in K, i \in S^k \tag{3.10}$$

$$z^k \le b^{\sigma(k)} \sum_{i\in S^k} x_i^k, \quad \forall k \in K \tag{3.11}$$

$$v_i^\ell, x_i^k \in \{0,1\}, z^k \ge 0 \quad \forall k \in K, i \in S^k, \ell \in \{1,\ldots,M\} \tag{3.12}$$

3.2.2 Inequalities

As already noted, (SLL$_1$) is a valid formulation for the Rank Pricing Problem. However, the newly introduced families of inequations, (3.10) and (3.11), are weak restrictions on z^k that may give poor linear relaxations during the execution of the branch-and-bound procedure. To see this, note that both right hand sides of (3.10) and (3.11) contain a sum of x_i^k variables multiplied by the constant quantity $b^{\sigma(k)}$, which may be quite big. Therefore, if for some instance $z^{k_0} = 0$ for some client k_0, then the $x_i^{k_0}$ variables may take any value whatsoever.

In order to alleviate this issue, the authors of [Calvete et al., 2019] defined the following set of valid inequalities for formulation (SLL$_1$) , which impose further bounds on the variables:

Proposition 8. *For any customer $k \in K$, integers $r_i^k \in \{0,\ldots,\sigma(k)\}$ $\forall i \in S^k$ and subsets $Q_i^k \subseteq \{1,\ldots,r_i^k - 1\}$ $\forall i \in S^k$, the following is a valid inequality for problem* (SLL$_1$) *:*

$$z^k \le \sum_{i\in S^k} \left(b^{r_i^k} x_i^k + \sum_{\ell=r_i^k+1}^{\sigma(k)} \left(b^\ell - b^{r_i^k}\right) v_i^\ell + \sum_{\ell\in Q_i^k} \left(b^\ell - b^{r_i^k}\right)\left(x_i^k + v_i^\ell - 1\right) \right). \tag{3.13}$$

Proof. The proof proceeds by analysing two cases, proving for each of them that the right-hand side of (3.13) takes a value greater or equal to that of a valid restriction of z^k ((3.10) or (3.11)).

Let us suppose that $x_{i_0}^k = 1$ for some $i_0 \in S^k$. That is, that k buys product i, and focus on the summand of (3.13) corresponding to i_0:

$$b^{r_{i_0}^k} x_{i_0}^k + \sum_{\ell=r_{i_0}^k+1} \left(b^\ell - b^{r_{i_0}^k}\right) v_{i_0}^\ell + \sum_{\ell\in Q_{i_0}^k} \left(b^\ell - b^{r_{i_0}^k}\right)\left(x_{i_0}^k + v_{i_0}^\ell - 1\right). \tag{3.14}$$

Since i_0 must be within k's budget, there exists some $\ell_0 \leq \sigma(k)$ such that $v_{i_0}^{\ell_0} = 1$. It will be proved by a further case analysis on ℓ_0 that (3.14) is greater than or equal to the price assigned to i_0, b^{ℓ_0}.

1. If $\ell_0 > r_{i_0}^k$, then $v_{i_0}^{\ell} = 0$ for every $\ell \in Q_{i_0}^k \subseteq \{1, \ldots, r_{i_0}^k - 1\}$, so $\sum_{\ell \in Q_{i_0}^k} \left(b^\ell - b^{r_{i_0}^k}\right)\left(x_{i_0}^k + v_{i_0}^\ell - 1\right) = 0$ and, since a product has no more than one price, $\sum_{\ell = r_i^k + 1}^{\sigma(k)} \left(b^\ell - b^{r_{i_0}^k}\right) v_{i_0}^\ell = b^{\ell_0} - b^{r_{i_0}^k}$. Finally, (3.14) reduces thus to $b^{r_{i_0}^k} + b^{\ell_0} - b^{r_{i_0}^k} = b^{\ell_0}$.

2. If, on the contrary, $\ell_0 \leq r_{i_0}^k$, then $v_{i_0}^\ell = 0$ for any $\ell > r_{i_0}^k$, and the first sum is now the one which vanishes, $\sum_{\ell = r_{i_0}^k + 1}^{\sigma(k)} \left(b^\ell - b^{r_{i_0}^k}\right) v_{i_0}^\ell = 0$, so that (3.14) gets reduced to

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

   If $\ell_0 \in Q_{i_0}^k$, this quantity is $b^{r_{i_0}^k} + b^\ell - b^{r_{i_0}^k} = b^\ell$. If $\ell \not\in Q_{i_0}^k$, then (3.14) equals $b^{r_{i_0}^k}$, which is larger than b^{ℓ_0}, for we are supposing that $\ell_0 \leq r_{i_0}^k$.

In any case, if k buys some i_0, the corresponding summand of i_0 in the right-hand side of (3.13), and therefore all of the right-hand side, is greater than or equal to the maximum value of z^k allowed by restriction (3.10). Thus, (3.13) is a valid inequality in this case.

Let us now suppose that $x_i^k = 0$ for all $i \in S^k$; that is, k buys no product. Consider the summand of (3.13) associated to an arbitrary product $i \in S^k$:

$$\sum_{\ell = r_i^k + 1} \left(b^\ell - b^{r_i^k}\right) v_i^\ell + \sum_{\ell \in Q_i^k} \left(b^\ell - b^{r_i^k}\right)\left(v_i^\ell - 1\right).$$

Note that the term $b^{r_i^k} x_i^k$ has been removed, since $x_i^k = 0$. As in the first summand $\ell \geq r_i^k + 1$, $b^\ell - b^{r_i^k} > 0$, so $\sum_{\ell = r_i^k + 1}^{\sigma(k)} \left(b^\ell - b^{r_i^k}\right) v_i^\ell \geq 0$. As in the second summand $\ell < r_i^k$, $b^\ell - b^{r_i^k} < 0$, and $v_i^\ell - 1 \leq 0$, so $\sum_{\ell \in Q_i^k} \left(b^\ell - b^{r_i^k}\right)\left(v_i^\ell - 1\right) \geq 0$. Therefore, both summands of (3.13) are, in this case, non-negative, so (3.13) is a non-negative upper bound on z^k, and thus a valid inequality for z^k also in the case $x_i^k = 0, \forall i \in S^k$. □

It is a remarkable fact that the given family of inequalities is expressive enough to contain both sets of constraints over z^k present in (SLL_1) . Indeed, constraints (3.10)

$$z^k \leq \sum_{\ell=1}^{\sigma(k)} v^\ell v_i^\ell + \sum_{j \in S^k : j \neq i} x_j^k, \quad \forall k \in K, i \in S^k$$

can be obtained from (3.13) by setting $r_i^k = 0$, $r_j^k = \sigma(k), \forall j \in S^k - \{i\}$ and $Q_j^k = \emptyset, \forall j \in S^k$, as the following derivation implies:

$$
\begin{aligned}
z^k \quad & \leq \quad \sum_{j \in S^k} \left(b^{r_j^k} x_j^k + \sum_{\ell = r_j^k + 1}^{\sigma(k)} \left(b^\ell - b^{r_j^k} \right) v_j^\ell + \sum_{\ell \in Q_j^k} \left(b^\ell - b^{r_j^k} \right) \left(x_j^k + v_j^\ell - 1 \right) \right) \\
& \iff \quad z^k \leq b^{r_i^k} x_i^k + \sum_{\ell = r_i^k + 1}^{\sigma(k)} \left(b^\ell - b^{r_i^k} \right) v_i^\ell + \sum_{\ell \in Q_i^k} \left(b^\ell - b^{r_i^k} \right) \left(x_i^k + v_i^\ell - 1 \right) \\
& \qquad + \sum_{j \in S^k : j \neq i} \left(b^{r_j^k} x_j^k + \sum_{\ell = r_j^k + 1}^{\sigma(k)} \left(b^\ell - b^{r_j^k} \right) v_j^\ell + \sum_{\ell \in Q_j^k} \left(b^\ell - b^{r_j^k} \right) \left(x_i^k + v_i^\ell - 1 \right) \right) \\
& \overset{(r_i^k = 0)}{\iff} \quad z^k \leq \sum_{\ell = r_i^k + 1}^{\sigma(k)} b^\ell v_i^\ell + \sum_{\ell \in Q_i^k} b^\ell \left(x_i^k + v_i^\ell - 1 \right) \\
& \qquad + \sum_{j \in S^k : j \neq i} \left(\sum_{\ell = r_j^k + 1}^{\sigma(k)} \left(b^\ell - b^{r_i^k} \right) v_j^\ell + \sum_{\ell \in Q_j^k} \left(b^\ell - b^{r_i^k} \right) \left(x_i^k + v_i^\ell - 1 \right) \right) \\
& \overset{\left(r_j^k = \sigma(k)\right)}{\iff} \quad z^k \leq \sum_{\ell = r_i^k + 1}^{\sigma(k)} b^\ell v_i^\ell + \sum_{\ell \in Q_i^k} b^\ell \left(x_i^k + v_i^\ell - 1 \right) + \sum_{j \in S^k : j \neq i} \left(b^\sigma(k) x_j^k + \sum_{\ell \in Q_j^k} \left(b^\ell - b^{\sigma(k)} \right) \left(x_i^k + v_i^\ell - 1 \right) \right) \\
& \overset{\left(Q_j^k = \varnothing\right)}{\iff} \quad z^k \leq \sum_{\ell = 1}^{\sigma(k)} b^\ell v_i^\ell + b^{\sigma(k)} \sum_{j \in S^k : j \neq i} x_j^k .
\end{aligned}
$$

Similarly, inequalities (3.11) can be derived from (3.13) by setting $r_i^k = \sigma(k)$ and $Q_i^k = \varnothing, \forall i \in S^k$, as evidenced by the following derivation:

$$
\begin{aligned}
z^k \quad & \leq \textstyle\sum_{j \in S^k} \left(b^{r_j^k} x_j^k + \sum_{\ell = r_j^k + 1}^{\sigma(k)} \left(b^\ell - b^{r_j^k} \right) v_j^\ell + \sum_{\ell \in Q_j^k} \left(b^\ell - b^{r_j^k} \right) \left(x_j^k + v_j^\ell - 1 \right) \right) \\
& \qquad \overset{(r_i^k = \sigma(k))}{\iff} z^k \leq \textstyle\sum_{i \in S^k} \left(b^{\sigma(k)} x_i^k + \sum_{\ell \in Q_i^k} \left(b^\ell - b^{r_i^k} \right) (x_i^k + v_i^\ell - 1) \right) \\
& \qquad \qquad \overset{(Q_i^k = \varnothing)}{\iff} z^k \leq b^{\sigma(k)} \textstyle\sum_{i \in S^k} x_i^k
\end{aligned}
$$

3.3 Second single level linear formulation

3.3.1 Formulation

The set of variables presented in the last section can be modified in order to reach an alternative formulation. The idea of dividing the total benefit as the sum of partial benefits remains intact. However, the degree of granularity in the election of variables may be increased. Concretely, for each client $k \in K$ and product $i \in S^k$ variable z_i^k will be defined as the benefits obtained from the acquisition of i by k if this acquisition takes place, and as zero if k does not buy i. The objective function will thus be

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

Note that $z^k = \sum_{i \in S^k} z_i^k, \forall k \in K$. The restrictions to add will be

$$z_i^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell, \quad \forall k \in K, i \in S^k \tag{3.15}$$

$$z_i^k \leq b^{\sigma(k)} x_i^k, \quad \forall k \in K, i \in S^k \tag{3.16}$$

$$z_i^k \geq 0, \quad \forall k \in K, i \in S^k. \tag{3.17}$$

That for any $k \in K$ and $i \in S^k$ these restrictions conform to the definition of z_i^k given above is justified by the following case analysis:

- If k buys i, then (3.16) is redundant: $z_i^k \leq b^{\sigma(k)} x_i^k = b^{\sigma(k)}$ and (3.15) gives the price of the i'th product: $z_i^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell$.
- If k does not buy i, then (3.16) takes the form $z_i^k \leq b^{\sigma(k)} x_i^k = 0$ and (3.15) is redundant: $z_i^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell$.

The addition of this set of variables to (SLNL) gives then rise to the following formulation, which we will refer to as (SLL_2) ,

$$\max_{v,x,z} \quad \sum_{k \in K} \sum_{i \in S^k} z^k \tag{3.18}$$

$$\text{s.t.} \quad \sum_{i \in S^k} x_i^k \leq 1, \quad \forall k \in K \tag{3.19}$$

$$\sum_{\ell=1}^{M} v_i^\ell \leq 1, \quad \forall i \in I \tag{3.20}$$

$$\sum_{j \in \overline{B(k,i)}} x_j^k + \sum_{\ell=1}^{\sigma(k)} v_i^\ell \leq 1, \quad \forall k \in K, i \in S^k : \overline{B(k,i)} \neq \emptyset \tag{3.21}$$

$$x_i^k + \sum_{\ell=\sigma(k)+1}^{M} v_i^\ell \leq 1, \quad \forall k \in K, i \in S^k \tag{3.22}$$

$$z^k \leq \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell, \quad \forall k \in K, i \in S^k \tag{3.23}$$

$$z^k \leq b^{\sigma(k)} x_i^k, \quad \forall k \in K, i \in S^k \tag{3.24}$$

$$v_i^\ell, x_i^k \in \{0,1\}, z^k \geq 0 \quad \forall k \in K, i \in S^k, \ell \in \{1,\dots,M\} \tag{3.25}$$

3.3.2 Inequalities

In a similar manner to formulation (SLL_1) , the families (3.23) and (3.24) from (SLL_2) lead to poor linear relaxations. In order to alleviate this issue, the authors of [Calvete et al., 2019] defined the following family of valid inequalities:

Proposition 9. *For any customer $k \in K$, product $i \in S^k$, integer $r_i^k \in \{0,\dots,\sigma(k)\}$ and subset $Q_i^k \subseteq \{1,\dots,r_i^k - 1\}$, the following is a valid inequality for formulation* (SLL_2) *:*

$$z^k \leq b^{r_i^k} x_i^k + \sum_{\ell=r_i^k+1}^{\sigma(k)} \left(b^\ell - b^{r_i^k}\right) v_i^\ell + \sum_{\ell \in Q_i^k} \left(b^\ell - b^{r_i^k}\right)\left(x_i^k + v_i^\ell - 1\right). \tag{3.26}$$

Proof. As in the case of the family of inequations (3.13), the validity of (3.26) can be proved by a case analysis on x_i^k. If $x_i^k = 1$, then $v_i^{\ell_0} = 1$ for some $\ell_0 \le \sigma(k)$, i must have had an assigned price within k's budget. We will consider two cases for ℓ_0:

1. If $\ell_0 \le r_i^k$, then $v_i^\ell = 0$ for all $\ell > r_i^k$, so $\sum_{\ell=r_i^k+1}^{\sigma(k)} \left(b^\ell - b^{r_i^k}\right) v_i^\ell = 0$, and the right-hand side results

$$b^{r_i^k} + \sum_{\ell \in Q_i^k} \left(b^\ell - b^{r_i^k}\right) v_i^\ell. \tag{3.27}$$

   If $\ell_0 \in Q_i^k$, then (3.27) is equal to $b^{r_i^k} + b^{\ell_0} - r^{r_i^k} = b^{\ell_0}$, which leads to the trivially true condition $z^k \le b^{\ell_0}$. If $\ell_0 \not\in Q_i^k$, then (3.27) equals $b^{r_i^k}$, which is greater than ℓ_0 due to the relation $\ell_0 \le r_i^k$, and the inequality is again true.

2. If $\ell_0 > r_i^k$, then $v_i^\ell = 0$ for all $\ell \in Q_i^k \subseteq \{1, \ldots, r_i^k - 1\}$ and the right-hand side results

$$b^{r_i^k} + b^{\ell_0} - b^{r_i^k} = b^{\ell_0}.$$

In any case, if $x_i = 0$, the right-hand side of (3.26) is greater than or equal to the upper bound of z_i^k, b^{ℓ_0}. If, on the other side, $x_i^k = 0$, the last argument used in the proof of the validity of 3.14 is usable, and implies that the right-hand side of (3.26) is non-negative. Since in this case $z_i^k \le 0$ is a valid upper bound, the non-negativity of (3.26)'s right-hand side implies again that (3.26) is a valid inequality. □

As in Section 3.2.2, the inequation family defined in the last result is expressive enough as to contain both restriction sets for z_i^k on (SLL$_2$) . Indeed, if $k \in K$ and $i \in S^k$, then taking $r_i^k = 0$ and $Q_i^k = \varnothing$, (3.26) takes the form of (3.23):

$$\begin{aligned}
z^k \quad & \le \quad b^{r_i^k} x_i^k + \sum_{\ell=r_i^k+}^{\sigma(k)} \left(b^\ell - b^{r_i^k}\right) v_i^\ell + \sum_{\ell \in Q_i^k} \left(b^\ell - b^{r_i^k}\right)\left(x_i^k + v_i^\ell - 1\right) \\
& \overset{(Q_i^k=\varnothing)}{\iff} \quad z_i^k \le b^{r_i^k} x_i^k + \sum_{\ell=r_i^k+1}^{\sigma(k)} \left(b^\ell - b^{r_i^k}\right) v_i^\ell \\
& \overset{(r_i^k=0)}{\iff} \quad z_i^k \le \sum_{\ell=1}^{\sigma(k)} b^\ell v_i^\ell
\end{aligned}$$

In a similar manner, if $k \in K$ and $i \in S^k$, we may take $r_i^k = \sigma(k)$ and $Q_i^k = \varnothing$ in order to transform (3.26) into (3.24):

$$\begin{aligned}
z^k \quad & \le \quad b^{r_i^k} x_i^k + \sum_{\ell=r_i^k+}^{\sigma(k)} \left(b^\ell - b^{r_i^k}\right) v_i^\ell + \sum_{\ell \in Q_i^k} \left(b^\ell - b^{r_i^k}\right)\left(x_i^k + v_i^\ell - 1\right) \\
& \overset{(Q_i^k=\varnothing)}{\iff} \quad z_i^k \le b^{r_i^k} x_i^k + \sum_{\ell=r_i^k+1}^{\sigma(k)} \left(b^\ell - b^{r_i^k}\right) v_i^\ell \\
& \overset{(r_i^k=\sigma(k))}{\iff} \quad z_i^k \le b^{\sigma(k)} x_i^k
\end{aligned}$$

3.4 Separation of inequalities

3.4.1 Introduction

The section will start with formulation (SLL_1) in mind, but its similarity with (SLL_2) will cause the conclusions for (SLL_1) to also be applicable to (SLL_2) . Let us thus consider a fractional solution of (SLL_1) , $(\overline{x}_i^k, \overline{v}_i^\ell, \overline{z}^k)$. The strategy will be to find integers r_i^k and sets Q_i^k minimising the value of the right-hand side of (3.13). he resulting inequality will be added to (SLL_1) iff it separates the fractional solution from the feasible set. That is, iff it is violated by $(\overline{x}_i^k, \overline{v}_i^\ell, \overline{z}^k)$.

Note that the sum of the right-hand side of (3.13) may be minimised by choosing each $r_i^k, Q_i^k, \forall i \in S^k$ so as to minimise the summand corresponding to product i. Therefore, for each $i \in S^k$ the following problem must be solved

$$\min_{r\in\{0,\dots,\sigma(k)\} Q\subseteq\{1,\dots,r-1\}} \left(b^r \overline{x}_i^k + \sum_{\ell=r+1}^{\sigma(k)} \left(b^\ell - b^r\right) \overline{v}_i^\ell + \sum_{\ell \in Q} \left(b^\ell - b^r\right)\left(\overline{x}_i^k + \overline{v}_i^k - 1\right)\right). \tag{3.28}$$

We have defined $r = r_i^k$ and $Q = Q_i^k$ in order to simplify the notation. Note that the result of this minimisation problem also minimises the right-hand side of (3.26) for $x = \overline{x}_i^k$ and $v = \overline{v}_i^\ell$. This will permit in (SLL_2) the same separation schema that presented for (SLL_1) . Studying (3.28), the following conclusion is reached:

Lemma 4. *For a fixed r, a possible value of $Q^r \subseteq \{1,\dots,r-1\}$ minimising* (3.28) *is*

$$Q^r = \{\ell \in \{1,\dots,r-1\} \ : \ \overline{x}_i^k + \overline{v}_i^\ell > 1\}.$$

Proof. It suffices to note that the only summand in (3.28) depending on Q is

$$\sum_{\ell \in Q} \left(b^\ell - b^r\right)\left(\overline{x}_i^k + \overline{v}_i^\ell - 1\right).$$

Since $Q \subseteq \{1,\dots,r-1\}$, the term $b^\ell - b^r$ is always negative in the sum above. Therefore, an optimal Q^r contains all ℓ such that $\overline{x}_i^k + \overline{v}_i^\ell - 1 > 0$, and no ℓ for which $\overline{x}_i^k + \overline{v}_i^\ell - 1 < 0$. □

This way, defining $S(r)$ to be the objective value of (3.28) when $Q = Q^r$, we may devote ourselves to studying the following problem, which is equivalent to (3.28):

$$\min_{r\in\{0,\dots,\sigma(k)\}} S(r) \tag{3.29}$$

3.4.2 Minimisation of S

In this section, S's minimum will be found. The objective is to prove that S has a very concrete shape: it is first decreasing, then increasing. We will start presenting the following auxiliary result characterising the first difference of S:

Lemma 5. *If $r < \sigma(k)$, then*

$$S(r+1) - S(r) = \left(b^{r+1} - b^r\right)\left(\overline{x}_i^k + \sum_{\ell=r+1}^{\sigma(k)} \overline{v}_i^\ell + \sum_{\ell \in Q^{r+1}} (1 - \overline{x}_i^k - \overline{v}_i^\ell).\right)$$

Proof. The result can be obtained by a direct derivation:

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

In (3.30), it has been used that $Q^{r+1} = Q^r \cup \{r\}$ if $\overline{x}_i^k + \overline{v}_i^r > 1$ and $Q^{r+1} = Q^r$ otherwise. □

This expression may now be used to prove that S is of the form described at the start of the section (decreasing, then increasing). This is done by proving the following result:

Proposition 10. *If $0 < r < \sigma(k)$, then $S(r) - S(r-1) \geq 0$ implies that $S(r+1) - S(r) \geq 0$.*

Proof. Since $b^{r+1} - b^r > 0$, from

$$S(r+1) - S(r) = \left(b^{r+1} - b^r\right)\left(\overline{x}_i^k + \sum_{\ell=r+1}^{\sigma(k)} \overline{v}_i^\ell + \sum_{\ell\in Q^{r+1}} \left(1 - \overline{x}_i^k - \overline{v}_i^\ell\right)\right)$$

we infer that $S(r) - S(r-1) \geq 0$ iff

$$\overline{x}_i^k + \sum_{\ell=r}^{\sigma(k)} \overline{v}_i^\ell + \sum_{\ell\in Q^{r}} \left(1 - \overline{x}_i^k - \overline{v}_i^\ell\right) \geq 0.$$

Therefore, in order to prove $S(r+1) - S(r) \geq 0$, it suffices to show that

$$\overline{x}_i^k + \sum_{\ell=r+1}^{\sigma(k)} \overline{v}_i^\ell + \sum_{\ell\in Q^{r+1}} \left(1 - \overline{x}_i^k - \overline{v}_i^\ell\right) \geq \overline{x}_i^k + \sum_{\ell=r}^{\sigma(k)} \overline{v}_i^\ell + \sum_{\ell\in Q^{r}} \left(1 - \overline{x}_i^k - \overline{v}_i^\ell\right).$$

However,

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

Which finishes the proof. □

Therefore, $S(r)$ reaches its minimum value for the least $0 < r < \sigma(k)$ such that $S(r) - S(r-1) \leq 0$ and $S(r+1) - S(r) > 0$, if it exists; $r = 0$ if S is strictly increasing, or $r = \sigma(k)$ if it is strictly decreasing or constant. This characterisation lends itself to a simple algorithm for finding this r: simply iterate from $r = 1$ to $\sigma(k) - 1$ until the minimality condition is satisfied.

However, continuing this study of the difference $S(r+1) - S(r)$, a simple rule for easily concluding $S(r) - S(r-1) \leq 0$ may be obtained without having to compute $\sum_{\ell \in Q^{r+1}} \left(1 - \overline{x}_i^k - \overline{v}_i^\ell\right)$:

Lemma 6. *If* $\overline{x}_i^k \leq \sum_{\ell=1}^{\sigma(k)} \overline{v}_i^\ell$, *then* $S(r) - S(r-1) \leq 0$.

Proof. It is enough to note that $b^{r+1} - b^r$ is always positive and $\sum_{\ell \in Q^{r+1}} \left(1 - \overline{x}_i^k - \overline{v}_i^\ell\right)$ always less than or equal to zero. □

3.4.3 Separation algorithm

Last section presented a method for solving minimisation problem (3.28). As indicated in Subsection 3.4.1, it may now be checked whether the restriction resulting from this minimisation procedure is violated by $(\overline{x}_i^k, \overline{v}_i^\ell, \overline{z}^k)$. If that were the case, it would be added to (SLL$_1$) . This procedure is resumed in the following algorithm:

Algorithm 1: Separation of inequalities for (SLL$_1$) .

Data: $(\overline{x}_i^k, \overline{v}_i^\ell, \overline{z}^k)$ an optimal fractional solution of the linear relaxation of (SLL$_1$) .
Result: An inequation from family 3.13 separating $(\overline{x}_i^k, \overline{v}_i^\ell, \overline{z}^k)$ if it can be found, nothing otherwise.

begin
- **for** $k \in K$ **do**
  - **for** $i \in S^k$ **do**
    - $r_i^k = 0$
    - **while** $r_i^k < \sigma(k)$ and $\overline{x}_i^k \leq \sum_{\ell=r_i^k}^{k} \overline{v}_i^\ell$ **do**
      - $r_i^k = r_i^k + 1$
    - **while** $r_i^k < \sigma(k)$ and $S(r_i^k + 1) - S(r_i^k) \leq 0$ **do**
      - $r_i^k = r_i^k + 1$
    - $Q_i^k = \{\ell \in \{1, \ldots, r_i^k - 1\} \mid \overline{x}_i^k + \overline{v}_i^\ell > 1\}$
  - Return constraint

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

  iff it is violated by $(\overline{x}_i^k, \overline{v}_i^\ell, \overline{z}^k)$.

Algorithm 2: Separation of inequalities for (SLL_2) .

Data: $(\overline{x}_i^k, \overline{v}_i^\ell, \overline{z}_i^k)$ an optimal fractional solution of the linear relaxation of (SLL_2) .

Result: An inequation from family 3.26 separating $(\overline{x}_i^k, \overline{v}_i^\ell, \overline{z}^k)$ if it can be found, nothing otherwise.

begin
- **for** $k \in K$ **do**
  - **for** $i \in S^k$ **do**
    - $r_i^k = 0$
    - **while** $r_i^k < \sigma(k)$ and $\overline{x}_i^k \le \sum_{\ell=r_i^k}^k \overline{v}_i^\ell$ **do**
      - $r_i^k = r_i^k + 1$
    - **while** $r_i^k < \sigma(k)$ and $S(r_i^k+1) - S(r_i^k) \le 0$ **do**
      - $r_i^k = r_i^k + 1$
    - $Q_i^k = \{\ell \in \{1, \ldots, r_i^k - 1\} \mid \overline{x}_i^k + \overline{v}_i^\ell > 1\}$
  - Return constraint

$$z_i^k \le b^{r_i^k} x_i^k + \sum_{\ell=r_i^k+1}^{\sigma(k)} \left(b^\ell - b^{r_i^k}\right) v_i^\ell + \sum_{\ell \in Q_i^k} \left(b^\ell - b^{r_i^k}\right)\left(x_i^k + v_i^\ell - 1\right)$$

  iff it is violated by $(\overline{x}_i^k, \overline{v}_i^\ell, \overline{z}_i^k)$.

Chapter 4

Polyhedral structure of the set packing subproblem

4.1 Introduction

All of the work presented up to this point has been concerned with the development of an adequate formulation for the RPP. However, despite all of this work, no mention has been made to precisely how good the obtained formulations are. However, some precise statements can be made regarding the tightness of some restrictions in formulations (SLL_1) and (SLL_2) , their presentation being this chapter's objective. This will be done noting that a particular set of restrictions of these formulations corresponds to a Set Packing Problem (see Definition 9), thus being able to apply to it Padberg's characterisation of SPP facets through intersection graphs. For a review of these topics, please refer to Section 1.5 in the Introduction.

Consider the families of restrictions (3.6) - (3.9) of problem (SLL_1) , which we will refer to as *set packing subproblem* (SPSP):

$$\text{(SPSP)} \qquad \sum_{i \in S^k} x_i^k \leq 1, \quad \forall k \in K \tag{4.1}$$

$$\sum_{\ell=1}^{M} v_i^\ell \leq 1, \quad \forall i \in I \tag{4.2}$$

$$\sum_{j \in \overline{B(k,i)}} x_j^k + \sum_{\ell=1}^{\sigma(k)} v_i^\ell \leq 1, \quad \forall k \in K, i \in S^k : \overline{B(k,i)} \neq \varnothing \tag{4.3}$$

$$x_i^k + \sum_{\ell=\sigma(k)+1}^{M} v_i^\ell \leq 1, \quad \forall k \in K, i \in S^k. \tag{4.4}$$

This terminology is justified by noting that the coefficient matrix of all these restrictions has binary entries, and that the right-hand side vector's components are all ones, thus corresponding this set of restrictions to the feasible region of a Set Packing Problem. Please note that these very restrictions can also be found in Formulation (SLL_2) , in constraints (3.19) - (3.22). Thus, by studying (SPSP) we will be able to derive conclusions equally applicable to both of these formulations.

However, due to technical details, we will have to add some restrictions to (SPSP). These additions were implicitly assumed in [Calvete et al., 2019], but here, we will state them explicitly. We will namely add all valid restrictions of the form $x_i^k + x_j^{k'} \leq 1$. An explicit enumeration of these will be obtained in Point 3 of Proposition 11.

Let us now represent by $G = (V, E)$ the intersection graph corresponding to (SPSP), as described in Definition 15, and by $P(G)$ the convex hull of all feasible solutions of (SPSP). Since tightness of restrictions has a direct influence in the efficiency of resolution, quality of formulations (SLL_1) and (SLL_2) may be checked by ensuring that many of the inequalities (4.1) - (4.4) are facets of $P(G)$. This will be obtained as conclusion to the main theorems presented in Section 4.3. Introduction of the intersection graph G does not serve merely explanatory purposes: it stands at the hearth of theorems 2, 3 and 4. The point in the proofs of these theorems in which G will allow us to make statements about facets is a notable theorem due to Padberg [Padberg, 1973]. It is therefore necessary to give a precise description of G, which will be done in next section.

4.2 Description of the intersection graph

In order to handle the intersection graph G, a characterisation of it through variables x and v must be obtained, since phrasing any argument in terms of (SPSP)'s constraint matrix and its rows would be overly cumbersome. However, as will be seen in the proof, these characterisations are almost direct translations of the constraint matrix's structure.

Proposition 11. *If G is the intersection graph associated to* (SPSP), then the following statements on G's nodes hold:

1. For any client $k \in K$ and distinct products $i \neq j$ in S^k, nodes x_i^k and x_j^k are adjacent.
2. For any pair of distinct clients $k, k' \in K$, $k \neq k'$ and product $i \in I$, nodes x_i^k and $x_j^{k'}$ are *not* adjacent.
3. For any pairs $k \neq k'$ and $i \neq j$ of distinct clients and products, adjacency of nodes x_i^k and $x_j^{k'}$ is equivalent to conditions $\sigma(k) \geq \sigma(k')$ and $j \in B(k,i)$.
4. For any client $k \in K$, product $i \in I$, and price index $\ell \in \{1, \ldots, M\}$, adjacency of nodes x_i^k and v_i^ℓ is equivalent to the condition $\ell > \sigma(k)$.
5. For any client $k \in K$, distinct products $i \neq j$ and price index $\ell \in \{1, \ldots, M\}$, adjacency of nodes x_i^k and v_j^ℓ is equivalent to conditions $\ell \leq \sigma(k)$ and $j \in B(k,i)$.
6. For any product $i \in I$, and distinct price indices $\ell \neq \ell'$, nodes v_i^ℓ and $v_i^{\ell'}$ are adjacent.
7. For any distinct products $i \neq j$ and pair of price indices $\ell, \ell' \in \{1, \ldots, M\}$, nodes v_i^ℓ and $v_j^{\ell'}$ are *not* adjacent.

Proof. This result will be proved by relating each of the assertions with their translation into statements about (SPSP)'s constraint matrix, which we will henceforth refer to as A. Please recall that, from the definition of intersection graph (Definition 15), two variables are adjacent iff a row of A exists which simultaneously assigns them a coefficient of 1. We will separately prove each statement:

1. For any client $k \in K$ and distinct products $i, j \in S^k$, variables x_i^k and x_j^k appear in the left-hand side of the (4.1) restriction associated to k. Therefore, x_i^k and x_j^k are adjacent in this case.
2. In this case, variables x_i^k and $x_i^{k'}$ do not simultaneously appear on the left-hand side of any of (SPSP)'s restrictions. This corresponds to the fact that knowing that k purchases a product i does not give any clue on whether k' will buy i.

3. If $x_i^k = 1$, that is, if k purchases product i, then k cannot afford any product k-better than i, since otherwise he would have bought it instead of i. Similarly, no client k' with a tighter budget, $\sigma(k') \leq \sigma(k)$, will be able to afford it either. Therefore, $x_j^{k'} = 1$, as stated on Point 3. Conversely, knowing that k does not purchase i provides us with no information on whether clients k' richer than i will be able to afford products $\overline{B(k,i)} \cup \{i\}$.

4. The only way x_i^k and v_i^ℓ can appear in the same left-hand side is in inequality family (4.4). This happens precisely when $\ell \geq \sigma(k)+1$.

5. The only way x_i^k and v_j^ℓ can appear in the same constraint is in constraint family 4.3, and in this family this only occurs for $l \leq \sigma(k)$ and $j \in B(k,i) \iff i \in \overline{B(k,j)}$.

6. It suffices to note that, for any $i \in I$, all the v_i^ℓ are added together in restriction family (4.2).

7. In none of the (SPSP) restrictions two v variables with different i indices get added together. Thus, v_i^ℓ and $v_j^{\ell'}$ are never adjacent if $i \neq j$.

□

4.3 Description of SPP facets

Having introduced last section's result on the structure of the intersection graph's structure we may now head on to present this chapter's main results. In order to do so, however, we will need to introduce some further nomenclature extending the definition of the $B(k,i)$ sets:

Definition 20. *If $k \in K$ is a customer and $P \subseteq S^k$, then we define $B(k,P) = \{i \in S^k \mid i >_k j \ \forall j \in P\}$. That is, $B(k,P)$ is the set of products which k prefers over all products in P. Similarly, we may define $\overline{B(k,P)}$ to be the set of products which are less preferred by k than any product in P. More precisely, $\overline{B(k,P)} = \{i \in S^k \mid i <_k j \ \forall j \in P\}$. Please note that $\overline{B(k,P)}$ need not be $B(k,P)$'s complement. For instance, products in P do not belong to any of these sets.*

Furthermore, an auxiliary result must be presented. It is a natural result stating that cliques containing some pair $v_i^{\ell_1}$ and $v_i^{\ell_2}$ of v-variables also contains all v-variables with ℓ between ℓ_1 and ℓ_2:

Lemma 7. *Any clique in G containing nodes $v_i^{\ell_1}$ and $v_i^{\ell_2}$ for some product i and price indices $\ell_1 < \ell_2$ also contains all nodes v_i^ℓ with $\ell_1 < \ell < \ell_2$.*

Proof. Given a clique G' in G containing $v_i^{\ell_1}$ and $v_i^{\ell_2}$, we will prove that if $\ell_1 < \ell < \ell_2$, then v_i^ℓ is adjacent to every node in G'. Since G contains x-nodes and v-nodes, we will prove that v_i^ℓ is adjacent to nodes in each of these families.

Firstly, if $v_j^{\ell'} \in G'$, then it is adjacent to $v_i^{\ell_1}$, since G' is a complete subgraph. By parts (6) and (7) of Proposition 11 this means that $i = j$. By these very results, we then get that $v_j^{\ell'}$ is adjacent to v_i^ℓ.

Secondly, if $x_i^k \in G'$, then it is adjacent to $v_i^{\ell_1}$, since G' is a complete subgraph. Thus, by Proposition 11 part (4), it must be the case that $\sigma(k) < \ell_1$. This very result asserts then that $\sigma(k) < \ell_1 < \ell$ implies x_i^k is adjacent to v_i^ℓ.

Lastly, if $x_j^k \in G'$ for some $j \neq i$, then it is adjacent to $v_i^{\ell_2}$, since G' is a complete subgraph. Thus, by Proposition 11 part 5, $j \in B(k,i) \iff i \in \overline{B(k,j)}$ and $\sigma(k) \geq \ell_2$. Therefore, since $\ell < \ell_2 \leq \sigma(k)$, that same result implies that x_j^k is adjacent to v_i^ℓ, thus ending the proof. □

It is now possible for us to introduce the main results in this section. These are 2 and 3, relating to x-variables and 4 relating to v-variables. For each set of variables, they present valid families of restrictions and conditions over these families for their restrictions to be facet-defining. The importance of these four results will become clear at the end of the section, when we discuss their implications for formulations (SLL$_1$) and (SLL$_2$) .

Theorem 2. *For any set of customers* $\{k_2,\dots,k_n\}$ *with* $n \geq 2$ *and ordered non-decreasingly by purchase power,* $\sigma(k_2) \leq \cdots \leq \sigma(k_n)$*, and for any non empty and pairwise disjoint sets* $P^{k_q} \subseteq S^{k_q}$*,* $q \in \{2,\dots,n\}$ *such that*

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

for all $q \in \{3,\dots,n\}$*, the following are valid restrictions for formulations* (SLL$_1$) *and* (SLL$_2$) *:*

$$\sum_{q=2}^{n} \sum_{j \in P^{k_q}} x_j^{k_q} \leq 1. \tag{4.5}$$

Proof. Let us denote by $G = (V,E)$ the intersection graph of the set packing subproblem, and let $Q = (V',E')$ be a clique of G containing only x variables. Choose k_2, a client with minimum budget in the clique, and $P^{k_2} \subseteq S^{k_2}$ such that $x_j^{k_2} \in V'$ for all $j \in P^{k_2}$.

Let us consider other customers k_q, $q = 3,\dots,n$ ordered such that $\sigma(k_2) \leq \cdots \leq \sigma(k_n)$, and alongside nonempty sets of products $P^{k_q} \subseteq S^{k_q}$ such that $x_j^{k_q} \in V'$, $\forall j \in P^{k_q}, q = 3,\dots,n$. Then, by the second point of Proposition 11, the $P^{k_2},\dots,P^{k_n}$ are pairwise disjoint. Furthermore, they verify the following two inclusions:

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

   If this relation didn't hold, there would exist some k_r with $\sigma(k_r) < \sigma(k_q)$ and products $i \in P^{k_r}$ and $j \in P^{k_q}$ with $x_i^{k_r}, x_j^{k_q} \in V'$ and $j \notin \overline{B(k_q,i)}$. However, this is absurd, because the third point of Proposition 11 would then imply that $x_i^{k_r}$ and $x_j^{k_q}$ would not be neighbours in G, contradicting that V' is a complete subgraph of G.

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

   If this were not the case, there would exist some k_r with $\sigma(k_r) = \sigma(k_q)$ and products $i \in P^{k_r}$ and $j \in P^{k_q}$ such that $x_i^{k_r}, x_j^{k_q} \in V'$. However, the third point of Proposition 11 implies that $x_i^{k_r}$ and $x_j^{k_j}$ are not neighbours in G. As before, this is absurd, since it contradicts that V is a complete graph.

In summary, these restrictions ensure that the x variables appearing in a sum of the type 4.5 give rise to a complete subgraph V', which implies that inequalities of family 4.5 are valid constraints for the set packing subproblem. □

Theorem 3. *Inequalities of the family* (4.5) *are facets of* (SPSP) iff

1. There does not exist any client $k_0 \in K$ and client $i_0 \in S^k$ satisfying the following conditions:

(a) i_0 belongs to $B(k_0,P^{k_0})$ for all $q \in \{2,\dots,n\}$ such that $\sigma(k_q) \geq \sigma(k_0)$.

(b) i_0 belongs to $\overline{B(k_0,P^{k_0})}$ for all $q \in \{2,\dots,n\}$ such that $\sigma(k_q) \leq \sigma(k_0)$.

2. $\left|\bigcap_{q=2:\sigma(k_q)=\sigma(k_2)}^{n} P_q^k\right| \geq 2$

Finally, all facets of (SPSP) whose associated clique in G consist of only x-variables are contained in family (4.5).

Proof. Let us suppose that conditions above are not fulfilled, and define (V,E), (V',E') as in the proof above. We have two options:

1. There exists some $(k_0,i_0) \in K \times S^{k_0}$ fulfilling the above conditions over (k_0,i_0). Then, Point 3 of Proposition 11 we have that the node $x_{i_0}^{k_0}$ is adjacent to every node in V', contradicting the fact that V' is a clique.

2. Last condition, $\left|\bigcap_{q=2:\sigma(k_q)=\sigma(k_2)}^{n} P_q^k\right| \geq 2$, holds. Then, there cannot exist any v-variable that is adjacent to all other nodes within subgraph V'. If this were not the case, we would have $P^{k_2} = \{i\}$, and one of $\sigma(k_2) < \sigma(k_3)$ or $n = 2$ would hold. However, this would imply that $v_i^{\sigma(k_2)+1}$ be adjacent to every node in V', contradicting the fact that the subgraph be maximal.

Otherwise, the construction in last proof above ensures that the graph be maximal, and thus, that the corresponding inequality is a facet. □

Theorem 4. *For any nonempty set of price indices $L = \{\ell_1,\dots,\ell_p\} \subseteq \{1,\dots,M\}$, product $i \in I$ and*

- *if $\ell_1 > 1$, customer k_1 such that $\sigma(k_1) = \ell_1 - 1$, $i \in S^{k_1}$, and set $P^{k_1} = \{i\}$; otherwise $P^{k_1} = \varnothing$;*
- *if $\ell_p < M$, customers $k_2,\dots,k_n$, $n \geq 2$, such that $\ell_p = \sigma(k_2) \leq \dots \leq \sigma(k_n)$ ($n = 1$ otherwise) and non empty pairwise disjoint sets of products $P^{k_q} \subseteq S^{k_q} - \{i\}$, $q \in \{2,\dots,n\}$ such that $P^{k_2} \subseteq \overline{B(k_2,i)}$ and*

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

for all $q \in \{3,\dots,n\}$, the following are valid restrictions for formulations (SLL$_1$) *and* (SLL$_2$) *:*

$$\sum_{\ell \in L} v_i^\ell + \sum_{q=1}^{n} \sum_{j \in P^{k_q}} x_j^{k_q} \leq 1 \tag{4.6}$$

In addition, inequalities of family (4.6) *are facets of* (SPSP) iff there does not exist any client $k_0 \in K$ and $i_0 \in S^{k_0} - \{i\}$ satisfying the following conditions:

- $\sigma(k_0) \geq \ell_p$
- i_0 belongs to $B(k_q,P^{k_q})$ for all $q \in \{1,\dots,n\}$ such that $\sigma(k_q) \geq \sigma(k_0)$.
- i_0 belongs to $\overline{B(k_0,P^{k_q})}$ for all $q \in \{1,\dots,n\}$ such that $\sigma(k_q) \leq \sigma(k_0)$.

Finally, all facets of (SPSP) whose associated clique in G consists of only x-variables are contained in family (4.6).

Proof. The proof will be similar to that of Theorem 2. However, the special structure for v variables suggested by Lemma 7 will induce a case analysis on the superindices of v_i^ℓ.

Let us denote by $G = (V,E)$ the intersection graph of the set packing subproblem (SPSP), and let $Q = (V',E')$ be a clique of G containing some v variables. Since all variables within the clique are adjacent, Point 7 of Proposition 11, all variables within clique V' must have the same subindex, which we may represent as i. Similarly, Lemma 7 asserts that all variables have consecutive superindices, their set being represented as $L = \{\ell_1,\dots,\ell_p\}$. We will now prove that the conditions in the statement are all met by following a case analysis on the extent of L.

- $L = \{1,\dots,M\}$.

  If this is the case, we may employ Proposition 11 in order to conclude that any x node x_j^k in the neighbourhood of L's associated v variables, $v_i^1,\dots,v_i^M$ must satisfy conditions $\sigma(k) = M$ and $j \in \overline{B(k,i)}$. Since $\sigma(k) = M$ means k is a richest customer, and the richest customers can purchase the product they prefer the most, we can infer that $\varnothing = P^{k_2} = \cdots = P^{k_n}$. Furthermore, Point 4 of Proposition 11 does not provide us with any node which is adjacent to all v-nodes v_i^ℓ, for all ℓ. We may therefore conclude that $\{v_i^\ell \mid \ell \in L\}$ is a maximal complete subgraph of the intersection graph G.

- There is some $\ell_1 > 1$ such that $L = \{\ell_1,\dots,M\}$.

  In this case, there exists a node in the clique adjacent to all v_i^ℓ with $\ell \geq \ell_1$, but not to $v_i^{\ell_1-1}$. Furthermore, Proposition 11 and Lemma 7 allow us to more explicitly classify this node as an x-variable. Let us consider this node is x_j^k for some client k and product j. By applying Point 5 of Proposition 11 we cannot obtain any other node adjacent to v_i^M, so it follows that $\varnothing = P^{k_2} = \cdots = P^{k_n}$. Then, Point 4 of the same proposition asserts that x_j^k has to be adjacent to all v_i^ℓ with $\ell \geq \ell_1$.. Therefore, $j = i$, and there exists some customer k_1 with $\sigma(k_1) < \ell_1$ and $P^{k_1} = \{i\}$ such that $k = k_1$. Since, by construction, $x_i^{k_1}$ is not adjacent to v-node $v_i^{\ell_1-1}$, Point 4 allows us to conclude that $\sigma(k_1) \geq \ell_1 - 1$, and therefore $\sigma(k_1) = \ell_1 - 1$. Thus, the first point in the statement is verified.

  Let us now suppose that there exists some other node $x_j^k \in V'$. In this case, Point 4 of Proposition 11 asserts that x_j^k is adjacent to all v_i^ℓ with $\ell \geq \ell_1$. Consequently, j must equal i. However, in this case Point 2 of the same proposition asserts that x_i^k and $x_i^{k_1}$ are not adjacent for any $k \neq k_1$, and therefore $\{v_i^\ell \mid \ell \geq \ell_1\} \cup \{x_i^{k_1}\}$ is in this case a clique in the intersection graph V'.

- There is some $\ell_p < M$ such that $L = \{1,\dots,\ell_q\}$.

  In a similar manner to that of last case, relations $v_i^\ell \notin V'$ for $\ell > \ell_p$, Lemma 7 and Proposition 11 imply the existence of an x-variable $x_{i_0}^k$ in V' adjacent to $v_i^{\ell_p}$ but not to $v_i^{\ell_p+1}$. We cannot obtain any node adjacent to v_i^1 by means of Point 4 of Proposition 11. Thus, $P^{k_1} = \varnothing$, and, by Point 5, $x_{i_0}^k$ and v_i^ℓ are adjacent for all $\ell \leq \ell_p$. We may thus infer the existence of a customer k_2 with budget $\sigma(k_2) \geq \ell_p$ and a subset of products $P^{k_2} \subseteq \overline{B(k_2,i)}$ such that $i_0 \in P^{k_2}$and, by Point 5, $x_j^{k_2} \in V'$ for all $j \in P^{k_2}$. Furthermore, that $x_{i_0}^{k_2}$ is not adjacent to v-node $v_i^{\ell_p+1}$ allows us to conclude relation $\sigma(k_2) = \ell_p$ in the theorem statement. If there also exist customers k_q for $3 \leq q \leq n$ with $\sigma(k_2) \leq \sigma(k_3) \leq \cdots \leq \sigma(k_n)$, and, associated to them, nonempty subsets $P^{k_q} \subseteq S^{k_q}$ such that $x_j^{k_q} \in F'$ for all $j \in P^{k_q}$, then an application of Point 2 allows us to conclude that the P^{k_i}, $1 \leq i \leq n$ are pairwise disjoint. Furthermore, none of the P^{k_q}, $3 \leq q \leq n$, contains product

i. If this were not the case, we would have $x_i^{k_q} \in V'$ for some k_q with $\sigma(k_q) \geq \ell_p$ not adjacent to $v_i^{\ell_p}$, contradicting the assumption of V' being a complete graph.

- There exist $\ell_1 > 1$ and $\ell_p < M$ such that $L = \{\ell_1, \ldots, \ell_p\}$.

  This case is handled by combining the arguments of the two previous cases.

□

Provided with these results, we are now able to assess the quality of (SPSP)'s inequalities. Indeed, cases 1 and 2 from the proof of Theorem 4 assert that restrictions from families (4.2) and (4.4) are always facet-defining. Furthermore, theorem 3 allows us to assert that all inequalities from (4.1) for which $|S^k| \geq 2$ and there does not exist any pair (k_0, i_0) in $K \times S^{k_0}$ satisfying $\sigma(k_0) \geq \sigma(k)$ and $i_0 \in \overline{B(k_0, S^k)}$. Certainly, this does not allow us to conclude that all inequalities in (4.1) are facet-defining, but it indicates that most of them are. In a similar manner, conditions from Theorem 4 allow us to conclude that most restrictions of family (4.3) are facet-defining.

It should be noted that being facet-defining in (SPSP) does not imply being facet-defining in either (SLL_1) or (SLL_2) , for these formulations also include z variables and further restrictions. However, it does indicate that they are very tight, thus leading to fast solving times.

Chapter 5

The envy-free Capacitated Rank Pricing Problem

Most of this work has been devoted to exposing the results obtained for the RPP in the paper [Calvete et al., 2019]. This chapter, however, will present some independent work carried out by the author in extending the RPP. Namely, we will be working on the extension known as CRPP, which, as will be explained later, has two variants: with envy and envy-free. The variant with envy, whose treatment is notoriously more difficult, was extensively studied in [Domínguez et al., 2022]. Here, we will present and independently develop a formulation for the envy-free CRPP. Furthermore, a computational study will be carried out for it.

5.1 Problem description

As currently stated, the RPP does not impose any restriction on the number of times an item might be bought. For instance, in a particular RPP instance, the optimal solution may involve all clients purchasing a particular product, not considering that this product might run out of stock. This is a reasonable simplification in a wealth of scenarios. For instance, a washing-machine manufacturer may price its various washing-machine models without fear of running out of stock, for it may adjust the production to meet the demand. In some other situations, however, this simplification may be too unrealistic. Think, for instance, of a company selling items which cannot be produced. This is the case of theatres, which offer a fixed number of seats and cannot change how many there are available. It is thus of interest to include this restriction to the RPP.

Another direction in which the RPP may be expanded is to allow clients to have a different budget for each product. As an example of when this added freedom may be necessary, let us return to the washing-machine application case. A manufacturer may offer models A and B, being A a usual washing machine, and offering B some added capabilities, like a low power consumption mode. A client may have a budget of €300 for A, but may be willing to pay up to €350 in order to enjoy B's added capabilities.

Including both of these capabilities into the RPP, stock limitation and selective budgets, we obtain the *Capacitated Rank Pricing Problem*, which was first presented in [Domínguez et al., 2022]. However, the inclusion of stock limitation can effectively lead to some customer not being able to purchase his preferred product because it has run out of stock, even though it may be within his budget. In this case, we may say this customer is envious of the customers who were able to acquire his preferred product. Whether we allow or not for this scenario to arise gives rise to two different versions of the CRPP: CRPP *with envy* and *envy-free* CRPP. The distinction is not merely theoretical, for the resolution of the CRPP with envy requires much more complicated formulations. The interested reader may consult these in [Domínguez et al., 2022]. Here, we will study the simpler envy-free variant.

The choice of whether or not allowing envy in the formulation should be studied on a case by case basis. Solutions allowing for envy allow for higher immediate gains, but may cause discontent amongst the clients, thus eroding the company's trustworthiness, which may lead to loss of profit over long periods of time.

5.2 Formulation

We will have to modify the notation presented for the RPP in order to accommodate the two additions presented for the CRPP: selective budgets and stock limitation. For the former, we will need to introduce, for each product $i \in I$, the number of units available, $c_i > 0$. For the latter, we must change the way budgets are represented. Now, the budget assignment function $\sigma : K \mapsto B$ must be divided into a family of functions, $\sigma_i : K \mapsto B$, with one such function for each σ_i. This reflects the fact that any client k may have different budgets for different products. We may apply these modifications to formulations (SLL_1) and (SLL_2) in order to obtain corresponding formulation for the envy-free version of CRPP:

$$
\begin{aligned}
(\text{EF}-\text{SLL}_1) \quad \max_{v,x,z} \quad & \sum_{k\in K} z^k && (5.1)\\
\text{s.t.} \quad & \sum_{i\in S^k} x_i^k \le 1, \quad \forall k \in K && (5.2)\\
& \sum_{\ell=1}^{M} v_i^\ell \le 1, \quad \forall i \in I && (5.3)\\
& \sum_{j\in \overline{B(k,i)}} x_j^k + \sum_{\ell=1}^{\sigma_i(k)} v_i^\ell \le 1, \quad \forall k \in K, i \in S^k : \overline{B(k,i)} \neq \emptyset && (5.4)\\
& x_i^k + \sum_{\ell=\sigma_i(k)+1}^{M} v_i^\ell \le 1, \quad \forall k \in K, i \in S^k && (5.5)\\
& z^k \le \sum_{\ell=1}^{\sigma_i(k)} b^\ell v_i^\ell + \sum_{j\in S^k : j\neq i} b^{\sigma_j(k)} x_j^k, \quad \forall k \in K, i \in S^k && (5.6)\\
& z^k \le \sum_{i\in S^k} b^{\sigma_i(k)} x_i^k, \quad \forall k \in K && (5.7)\\
& \sum_{k\in K} x_i^k \le c_i, \quad \forall i \in I && (5.8)\\
& v_i^\ell, x_i^k \in \{0,1\}, z^k \ge 0 \quad \forall k \in K, i \in S^k, \ell \in \{1,\dots,M\}. && (5.9)
\end{aligned}
$$

Here, we have made minor changes in the restrictions of (SLL_1) in order to use σ_i instead of σ. In restrictions (5.4) and (5.5) these changes have been merely formal, syntactically substituting one function for the other. However, in restrictions (5.6) and (5.7) we have had to move the $b^\sigma(k)$ terms inside their respective summations. Other than these changes, restrictions (5.8) have been included. These restrictions assert that, for each product $i \in I$, no more than c_i units have been sold in total. Therefore, these constraints codify the stock limitations. In a similar manner, we can present an extension of formulation (SLL_2) :

$$
\begin{aligned}
(\mathrm{EF}-\mathrm{SLL}_2) \quad \max_{v,x,z} \quad & \sum_{k\in K}\sum_{i\in S^k} z_i^k && (5.10)\\
\text{s.t.} \quad & \sum_{i\in S^k} x_i^k \le 1, \quad \forall k\in K && (5.11)\\
& \sum_{\ell=1}^{M} v_i^\ell \le 1, \quad \forall i\in I && (5.12)\\
& \sum_{j\in \overline{B(k,i)}} x_j^k + \sum_{\ell=1}^{\sigma_i(k)} v_i^\ell \le 1, \quad \forall k\in K, i\in S^k : \overline{B(k,i)} \ne \varnothing && (5.13)\\
& x_i^k + \sum_{\ell=\sigma_i(k)+1}^{M} v_i^\ell \le 1, \quad \forall k\in K, i\in S^k && (5.14)\\
& z_i^k \le \sum_{\ell=1}^{\sigma_i(k)} b^\ell v_i^\ell, \quad \forall k\in K, i\in S^k && (5.15)\\
& z_i^k \le b^{\sigma_i(k)} x_i^k, \quad \forall k\in K, i\in S^k && (5.16)\\
& \sum_{k\in K} x_i^k \le c_i, \quad \forall i\in I && (5.17)\\
& v_i^\ell, x_i^k \in \{0,1\}, z^k \ge 0 \quad \forall k\in K, i\in S^k, \ell\in\{1,\dots,M\}. && (5.18)
\end{aligned}
$$

As in the case of $(\mathrm{EF}-\mathrm{SLL}_1)$, most differences introduced by $(\mathrm{EF}-\mathrm{SLL}_2)$ with respect to (SLL_2) are formal substitutions from σ to σ_i. In this case, the changes have been introduced in constraints (5.13), (5.14), (5.15), and (5.16). Restriction family (5.17) is new, representing the limitation of product stocks.

5.3 Computational study

In order to assess the performance of formulations $(\mathrm{EF}-\mathrm{SLL}_1)$ and $(\mathrm{EF}-\mathrm{SLL}_2)$, a computational study has been carried out. This has been done over AMPL Version 20230124, and over the 20.1.0.0 version of the CPLEX commercial solver. The device hosting the experiments was an M1 MacBook personal computer with 8GB of memory.

In order to conduct the computational study, we have randomly generated several instances of the CRPP. For this, the number of clients has been set to 50. Then, three instances have been independently generated for all possible combinations of the following parameters:

- The number of items in an instance, $|I| \in \{5, 15, 25, 35, 45\}$.
- The cardinalities of the sets of preferences $|S^k|$, which are defined to be 10, 20 or 50% the size of $|I|$.

Thus, there are a total of $5 \times 3 = 15$ configurations. Since three instances were independently generated for each configuration, there were a total of $15 \times 3 = 45$ instances. Finally, in each instance, capacities c_i have been set to an uniformly random number between 5% and 10% of the number of clients. Table 5.1 summarises these results. Columns $|I|$ and $|S^k|$ identify instance parameter configurations. Then, the number of nodes and the time required to solve each instance by each formulation is displayed in the next four columns. These results are averaged over the three independent

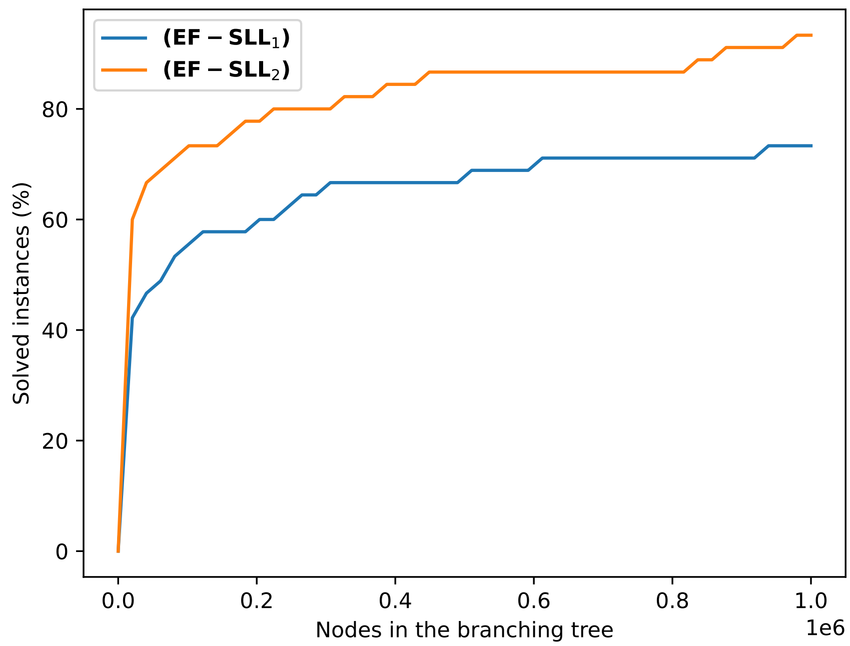

Figure 5.1: Plot for nodes

instances corresponding to each parameter configuration. These results were further summarised in figures 5.1 and 5.2, which we will now comment.

Figure 5.1 represents the percentage of instances that a particular formulation was able to solve using less than a certain number of nodes in its branch and bound tree. The x-axis represents the number of nodes in the branching tree, and the y-axis the percentage of instances that could be solved by using fewer than the corresponding number of nodes. The number of nodes needed to solve an instance may be considered as an informal measure of the formulation's complexity. Indeed, having less nodes in the branch-and-bound tree implies that the solver needs solving less linear problems for obtaining the best integer solution. We can see that $(\mathrm{EF}-\mathrm{SLL}_2)$ consistently obtained better results than $(\mathrm{EF}-\mathrm{SLL}_1)$, being able to solve instances using fewer nodes.

Figure 5.2 represents the percentage of instances that the solver was able to solve to optimality within a given time. The x-axis represents time in seconds, and the y-axis represents the percentage of instances that were solved in less than a given amount of time. We see that, again, formulation $(\mathrm{EF}-\mathrm{SLL}_2)$ was consistently better, being able to solve most instances within 200 seconds. Formulation $(\mathrm{EF}-\mathrm{SLL}_1)$, on the other hand, needed more than 600 seconds to solve most of its instances.

Overall, results were good for both instances, thus providing empirical proof for the quality of formulations (SLL_1) and (SLL_2), on which $(\mathrm{EF}-\mathrm{SLL}_1)$ and $(\mathrm{EF}-\mathrm{SLL}_2)$ are based. Therefore, this practical argument adds up to the theoretical ones provided in Chapter 4.

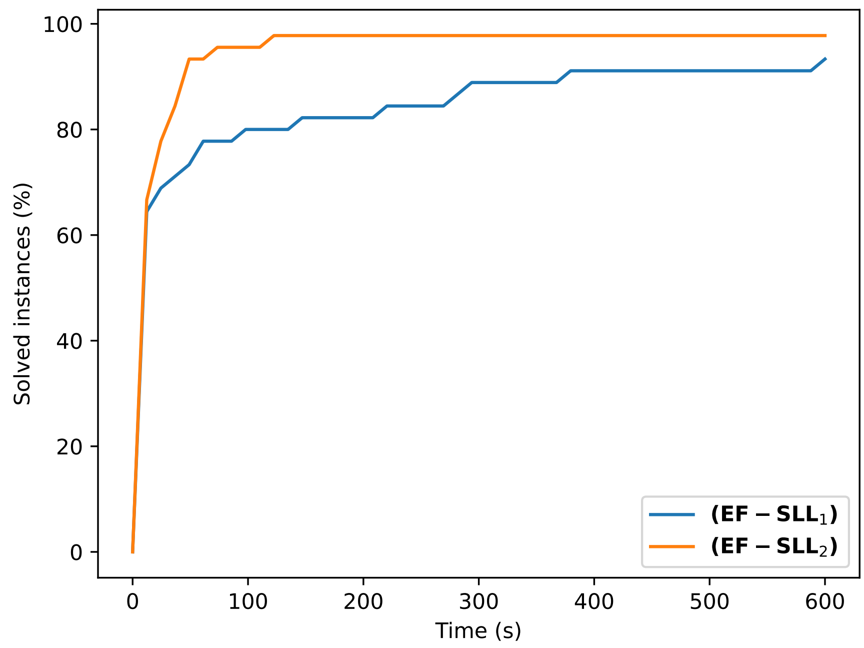

Figure 5.2: Plot for time

$\lvert I\rvert$	$\lvert S^k\rvert$	$(\mathbf{EF}-\mathbf{SLL}_1)$		$(\mathbf{EF}-\mathbf{SLL}_2)$	
		Nodes	Time (s)	Nodes	Time (s)
5	1	13	0.1	13	0.1
5	2	9202	0.2	1685	0.2
5	3	9149	0.3	2856	0.2
15	2	756	0.2	450	0.1
15	4	1985090	25.3	50972	1.0
15	8	42150631	623.9	524614	29.6
25	3	48444	0.8	766	0.2
25	7	5290711	99.2	150064	7.2
25	13	35859122	685.3	893052	47.8
35	4	99160	2.2	851	0.2
35	9	253636	6.9	7383	17.9
35	18	821013	112.3	727472	268.6
45	5	23659	0.7	934	0.2
45	12	104889	8.8	145355	20.0
45	23	422506	717.6	708252	44.4

Table 5.1: Results of the computational study

Chapter 6

Conclusions

In this work, we have studied the Rank Pricing Problem, an NP-hard integer optimisation problem. In Chapter 2, a bilevel formulation was first presented building on a natural interpretation of the program as an interaction between seller and customer. Then, single-level formulations were obtained from this bilevel one. In Chapter 3, single-level linear formulations were introduced, building on intuition from the formulations in Chapter 2. Furthermore, families of valid inequalities of exponential size were introduced, alongside a separation procedure for dynamically introducing them into the formulation. Chapter 4 presented a theoretical study characterising the facet-defining restrictions of (SPSP), a set packing subproblem found in formulations (SLL_1) and (SLL_2) . The results there introduced permitted assessing the tightness of (SLL_1) and (SLL_2) 's restrictions.

Finally, Chapter 5 contained the author's original contributions. It presented the Capacitated Rank Pricing Problem, an extension for the RPP allowing the seller to restrict the quantity of items available from each product. Two formulations were developed for the envy free version of the problem based on (SLL_1) and (SLL_2). This complements the study carried out in [Domínguez et al., 2022], which gave formulation for the more complex case of CRPP with envy. Finally, Chapter 5 also presented a computational study on the proposed formulations for the CRPP, thus assesssing their performance.

Bibliography

[Bazaraa et al., 2009] Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D. (2009). *Linear Programming and Network Flows, 4th Edition*. John Wiley & Sons, Inc.

[Calvete et al., 2019] Calvete, H. I., Domínguez, C., Galé, C., Labbé, M., and Marín, A. (2019). The rank pricing problem: Models and branch-and-cut algorithms. *Computers & Operations Research*, 105:12–31.

[Dantzig, 1981] Dantzig, G. B. (1981). Reminiscences about the origins of linear programming. Technical report, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

[Dempe, 2002] Dempe, S. (2002). *Foundations of Bilevel Programming*. Kluwer Academic Publishers.

[Domínguez et al., 2022] Domínguez, C., Labbé, M., and Marín, A. (2022). Mixed-integer formulations for the capacitated rank pricing problem with envy. *Computers & Operations Research*, 140:105664.

[Nemhauser and Wolsey, 1988] Nemhauser, G. and Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc.

[Padberg, 1973] Padberg, M. W. (1973). On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215.

[Rusmevichientong et al., 2006] Rusmevichientong, P., Van Roy, B., and W. Glynn, P. (2006). A non-parametric approach to multiproduct pricing. *Operations Research*, 54:82–98.

[Sierksma and Zwols, 2015] Sierksma, G. and Zwols, Y. (2015). *Linear and Integer Optimization: Theory and Practice, Third Edition (3rd ed.)*. Chapman and Hall/CRC.

[von Stackelberg, 1934] von Stackelberg, H. F. (1934). *Marktform und Gleichgewicht*. Springer-Verlag, Berlin.

[Wolsey, 1988] Wolsey, L. A. (1988). *Integer Programming*. John Wiley & Sons, Inc.