



---

**Universidad de Valladolid**

**FACULTAD DE CIENCIAS**

**TRABAJO FIN DE GRADO**

**Grado en Matemáticas**

**TEORÍA POLIÉDRICA EN PROGRAMACIÓN LINEAL ENTERA**

**Autora:**

**Rocío Alonso González**

**Tutores:**

**Pedro César Álvarez Esteban**

**Jesús Sáez Aguado**

**2023/2024**



*A Pedro César y Jesús Sáez, por aceptar ser mis tutores en este trabajo. En particular a Jesús, por su paciencia, dedicación y esfuerzo para sacar adelante el trabajo de la mejor manera posible.*

*A mi familia, en especial a mis padres, por recordarme que con esfuerzo y constancia todo llega, por apoyarme en cada una de mis decisiones y confiar siempre en mí.*

*A mis amigas, por acompañarme en este camino, por apoyarme en los momentos difíciles y celebrar mis logros tanto o más que yo. En especial a Dani, por estar a mi lado desde el primer momento y recordarme que soy capaz de conseguir lo que me proponga.*



# Índice general

<b>Resumen</b>	<b>7</b>
<b>Abstract</b>	<b>9</b>
<b>Introducción</b>	<b>11</b>
<b>1. Teoría poliédrica</b>	<b>13</b>
1.1. Poliedros y resultados básicos . . . . .	13
1.2. Envolverte convexa en programación entera . . . . .	19
<b>2. Problemas bien resueltos</b>	<b>25</b>
2.1. Poliedros enteros . . . . .	25
2.2. Matrices unimodulares . . . . .	28
2.3. Problemas de flujo de redes y de transporte . . . . .	33
2.3.1. Problema de flujo de redes . . . . .	33
2.3.2. Problema de transporte . . . . .	35
<b>3. Descripción parcial de <math>\text{conv}(S)</math> mediante desigualdades</b>	<b>37</b>
3.1. Relaxaciones y desigualdades válidas . . . . .	37
3.2. Algoritmo genérico de planos de corte . . . . .	41
3.3. Algunas familias de desigualdades . . . . .	42
3.3.1. Desigualdades de cubrimiento . . . . .	43
3.3.2. Aritmética modular . . . . .	47
3.3.3. Redondeo de Chvátal-Gomory . . . . .	48
3.4. Formulación fuerte y débil. Problema de la p-mediana . . . . .	50
<b>4. Cortes de Gomory</b>	<b>59</b>
4.1. Cortes Fundamentales . . . . .	60
4.2. Problema entero puro . . . . .	60
4.3. Convergencia finita del algoritmo . . . . .	65
4.4. Problema entero mixto . . . . .	67
4.5. Dificultades computacionales . . . . .	69
<b>Lista de figuras y tablas</b>	<b>71</b>
<b>Bibliografía</b>	<b>72</b>



# Resumen

La programación lineal entera es una rama importante de la programación lineal en la que las variables toman valores enteros, lo que dificulta la búsqueda de la solución óptima.

Sin embargo, se emplea en diferentes ámbitos con el fin de optimizar una función objetivo sujeta a una serie de restricciones.

Este trabajo se centra en dar una descripción ideal de la envolvente convexa de la región factible y acercarnos lo más posible a la solución óptima del problema mediante la utilización de algoritmos de corte.

En el primer capítulo se presentan definiciones y resultados fundamentales de la teoría poliédrica, además de la formulación de los diferentes tipos de problemas de programación entera (pura, mixta y binaria) y el Teorema de Meyer, resultado principal de este capítulo.

El Capítulo 2 se centra en la resolución efectiva de problemas de programación entera, debido a que se conoce la descripción completa de la envolvente convexa. Se estudian los conceptos de poliedro entero y matrices unimodulares y se analizan los problemas de transporte y flujo de redes.

En el tercer capítulo se profundiza en la descripción parcial de la envolvente convexa mediante familias de desigualdades, se introducen los conceptos de desigualdades válidas, relajaciones y cortes. Se detalla el algoritmo de planos de corte, como herramienta para aproximarse a la solución óptima de un problema de programación entera. Y finalmente se da una lista con una breve explicación de algunas familias de desigualdades.

Por último en el Capítulo 4 se tratan los cortes de Gomory, el algoritmo de cortes fraccionarios de Gomory tanto para el problema de programación entera puro como para el mixto, y se estudia la efectividad de dicho algoritmo para llegar a una solución óptima en un número finito de iteraciones.

**Palabras clave:** Programación lineal entera, planos de corte, desigualdades válidas, relajaciones, envolvente convexa y cortes de Gomory.



# Abstract

Integer linear programming is an important branch of linear programming in which variables take on integer values, making it difficult to find the optimal solution. However, it is used in different areas in order to optimize an objective function subject to a series of constraints.

This work focuses on giving an ideal description of the convex envelope and getting as close as possible to the optimal solution of the problem by using cutting algorithms.

In the first chapter, definitions and fundamental results of polyhedral theory are presented, as well as the formulation of the different types of integer programming problems (pure, mixed and binary) and Meyer's Theorem, the main result of this chapter, is presented.

Chapter 2 focuses on the effective resolution of integer programming problems, as the complete description of the convex envelope is known. The concepts of integer polyhedron and unimodular matrices are studied and the problems of transport and network flow are analyzed.

In the third chapter, the partial description of the convex envelope is deepened by means of families of inequalities, the concepts of valid inequalities, relaxations and cuts are introduced. The cutting plane algorithm is detailed as a tool to approximate the optimal solution of an entire programming problem. And finally a list is given with a brief explanation of some families of inequalities.

Finally, Chapter 4 deals with Gomory cuts, Gomory's fractional cut algorithm for both the pure and mixed integer programming problem, and studies the effectiveness of this algorithm to arrive at an optimal solution in a finite number of iterations.

**Keywords:** Integer linear programming, cutting planes, valid inequalities, relaxations, convex hull and Gomory cuts.



# Introducción

La programación lineal es una disciplina dentro de la optimización matemática que se desarrolló a mediados del siglo XX.[5] En 1947, el estadounidense George Dantzig, desarrolló el método simplex, lo que permitió resolver problemas de optimización lineal de manera más efectiva.

Este método se convirtió rápidamente en una herramienta importante en numerosas áreas gracias a su capacidad de resolver grandes sistemas de ecuaciones con múltiples restricciones.

En los problemas de programación lineal [7], se busca maximizar o minimizar una función, conocida como función objetivo, para llegar a la solución óptima dentro de la región factible definida por las restricciones. La formulación general de un problema de programación lineal es

$$\text{mín}\{cx : Ax \leq b, x \geq 0\}$$

donde  $A$  es una matriz real de dimensiones  $m \times n$ ,  $x \in \mathbb{R}^n$  es el vector columna de variables de decisión,  $b \in \mathbb{R}^m$  es el vector columna de términos independientes y  $c \in \mathbb{R}^n$  es el vector fila de los coeficientes de la función objetivo.

A lo largo del trabajo supondremos que la mayoría de los problemas de optimización considerados son de la forma minimizar la función objetivo.

La programación lineal entera extiende este concepto al considerar variables con valores enteros, lo que dificulta la búsqueda de la solución óptima debido a la naturaleza discreta de la región factible.

En este caso la formulación del problema es de la siguiente manera

$$\text{mín}\{cx + hy : Ax + Gy \leq b, x \geq 0, y \geq 0, x \in \mathbb{R}^n, y \in \mathbb{Z}^p\}$$

Donde  $A$  es una matriz  $m \times n$ ,  $G$  una matriz  $m \times p$ ,  $x$  un vector de dimensión  $n$  e  $y$  de dimensión  $p$  y  $c \in \mathbb{R}^n$  y  $h \in \mathbb{R}^p$  son los vectores fila de los coeficientes de la función objetivo.

La región factible de estos problemas deja de ser convexa, por lo que, inicialmente, dificulta la aplicación de la teoría de poliedros.

Sin embargo, gracias a resultados, como el teorema de Meyer, que estudiaremos a continuación, podemos ver que la envolvente convexa de la región factible es un

poliedro.

A lo largo de las décadas de 1950 y 1960 se realizaron grandes aportaciones a la programación entera de matemáticos como Ralph Gomory, que introdujo los cortes de Gomory.

Sin embargo, los avances computacionales más recientes han permitido abordar problemas de programación entera más complejos y de mayor escala.

El trabajo de Crowder, Harlan and Johnson, Ellis L and Padberg, Manfred, “Solving large-scale zero-one linear programming problems” [2] se considera un punto de inflexión en la mejora de la resolución de problemas de programación entera, debido precisamente a la obtención de relajaciones fuertes mediante algoritmos de planos de corte.

# Capítulo 1

## Teoría poliédrica

### 1.1. Poliedros y resultados básicos

La teoría poliédrica es un marco fundamental para la investigación operativa que proporciona definiciones y resultados básicos para la comprensión de problemas de programación entera.

En este capítulo vamos a estudiar aquellas definiciones y resultados sobre la teoría de poliedros que nos serán de gran utilidad en los siguientes capítulos.

A partir de ahora consideramos  $A$  una matriz de dimensiones  $m \times n$  y  $b$  un vector columna de dimensión  $m$ , ambos de coeficientes racionales.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & \cdots & a_{mn} \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

**Definición 1.1.** Llamamos poliedro racional,  $P \subseteq \mathbb{R}^n$ , al conjunto de puntos que satisface un número finito de inecuaciones lineales de la forma  $P = \{x \in \mathbb{R}^n : a^i x \leq b_i, \forall i \in \{1, \dots, m\}\}$ . [11]

**Definición 1.2.** Un poliedro racional  $P \subseteq \mathbb{R}^n$  se dice que es acotado si existe  $w \in \mathbb{R}$  tal que  $P \subseteq \{x \in \mathbb{R}^n : -w \leq x_j \leq w, \forall j \in \{1, \dots, n\}\}$ .

**Definición 1.3.** Sea  $P = \{x \in \mathbb{R}^n : a^i x \leq b_i, \forall i \in \{1, \dots, m\}\}$  un poliedro racional. La dimensión de  $P$  es  $k$ ,  $\dim(P) = k$ , si el número máximo de puntos linealmente independientes de  $P$  es  $k + 1$ .

**Definición 1.4.** Decimos que  $x$  es un punto interior de  $P$  si para todo  $i \in \{1, 2, \dots, m\}$  se satisface  $a^i x < b_i$ .

Vamos a considerar ahora  $(A^{\bar{=}}, b^{\bar{=}})$  las filas correspondientes de  $(A, b)$  que satisfacen las ecuaciones  $a^i x = b_i, \forall x \in P$ .

**Proposición 1.5.** *Dado un poliedro  $P$ , si  $P \subset \mathbb{R}^n$  entonces  $\dim(P) + \text{rang}(A^{\bar{=}}, b^{\bar{=}}) = n$ . [11]*

*Demostración.* Suponemos que  $\text{rang}(A^{\bar{=}}) = \text{rang}(A^{\bar{=}}, b^{\bar{=}}) = n - k$  con  $0 \leq k \leq n$ . Sabemos que hay  $k + 1$  soluciones linealmente independientes del sistema  $A^{\bar{=}} x = 0$ , y las denotamos como  $y^1, y^2, \dots, y^{k+1}$ .

Sea  $\hat{x}$  un punto interior de  $P$  entonces para un  $\varepsilon > 0$  lo suficientemente pequeño tenemos  $\hat{x} + \varepsilon y^i$  con  $i \in \{1, \dots, k + 1\}$  son puntos linealmente independientes de  $P$ , luego  $\dim(P) \geq k$  y por tanto  $\dim(P) + \text{rang}(A^{\bar{=}}, b^{\bar{=}}) \geq n$ .

Si consideramos el caso  $\dim(P) = k$ , con  $x^1, \dots, x^{k+1}$  puntos linealmente independientes de  $P$  tal que  $A^{\bar{=}} x^i = b^{\bar{=}}$  con  $i \in \{1, \dots, k + 1\}$ . Tendremos entonces  $\text{rang}(A^{\bar{=}}, b^{\bar{=}}) \leq n - k$  y  $\dim(P) + \text{rang}(A^{\bar{=}}, b^{\bar{=}}) \leq n$ .

Es decir,  $\dim(P) + \text{rang}(A^{\bar{=}}, b^{\bar{=}}) = n$ .

□

**Ejemplo 1.6.** Consideramos el poliedro acotado  $P \in \mathbb{R}^3$  dado por las inecuaciones siguientes:

$$x_1 + x_2 + x_3 \leq 1 \tag{1.1}$$

$$-x_1 - x_2 - x_3 \leq -1 \tag{1.2}$$

$$x_1 + x_3 \leq 1 \tag{1.3}$$

$$-x_1 \leq 0 \tag{1.4}$$

$$-x_2 \leq 0 \tag{1.5}$$

$$x_3 \leq 3 \tag{1.6}$$

$$x_1 + x_2 + 2x_3 \leq 2 \tag{1.7}$$

Representado en la Figura 1.1

Los puntos  $(1, 0, 0)$ ,  $(0, 1, 0)$ ,  $(0, 0, 1)$  son linealmente independientes y están en  $P$ , y como los tres satisfacen la igualdad de (1.1)  $x_1 + x_2 + x_3 = 1$  se tiene que  $\text{rang}(A^{\bar{=}}, b^{\bar{=}}) \geq 1$  y por la proposición anterior  $\dim(P) \leq 2$ . Luego  $\dim(P) = 2$ . [11]

A continuación veremos como se puede representar un poliedro en términos de puntos y direcciones extremas. Comenzaremos con las definiciones y teoremas previos antes de ver el resultado fundamental.[11]

**Definición 1.7.** Dado un  $x \in P$ , decimos que es un punto extremo de  $P$  si no puede escribirse como combinación estrictamente convexa de dos puntos distintos de  $P$ . Es decir, que si tenemos  $x = \lambda x_1 + (1 - \lambda)x_2$  con  $\lambda \in \mathbb{R}$ , entonces  $x$  no es punto extremo.

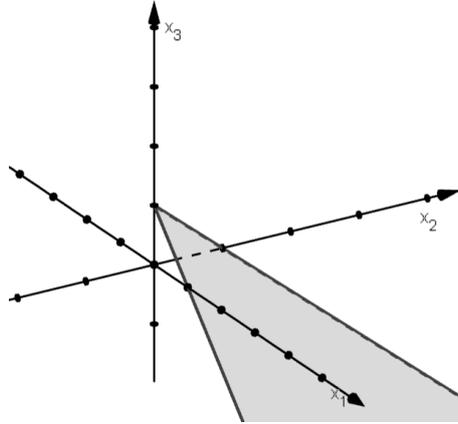


Figura 1.1: Poliedro

**Definición 1.8.** Dado un vector  $d \neq 0$ , decimos que  $d$  es dirección de  $P$  si satisface  $Ad \leq 0$ .

También podemos decir que  $d$  es dirección de  $P$  si y solo si para todo  $x \in P$  y  $\lambda \geq 0$  se tiene que  $x + \lambda d \in P$ .

**Definición 1.9.** Una dirección  $d$  de  $P$  decimos que es dirección extrema si no se puede escribir como una combinación lineal positiva de dos direcciones distintas de  $P$ , es decir, si no existen  $d^1, d^2$ ,  $d^1 \neq \lambda d^2$  para cualquier  $\lambda \geq 0$ , tal que para  $\mu_1, \mu_2 \geq 0$ , se tiene  $d = \mu_1 d^1 + \mu_2 d^2$ .

**Definición 1.10.** Sea  $P$  un poliedro racional y sea  $cx \leq c_0, \forall x \in P$  una desigualdad válida de  $P$ , decimos que el conjunto  $F = \{x \in P : cx = c_0\}$  es una cara de  $P$ .

**Proposición 1.11.** Sea  $P \neq \emptyset$  un poliedro racional y  $\text{rang}(A) = n - k$ , entonces  $P$  tiene una cara  $F$  de dimensión  $n - \text{rang}(A) = k$  y además no tiene una cara  $F' \neq \emptyset$  de dimensión menor.

*Demostración.* Sea  $F \neq \emptyset$  una cara de  $P$  y  $\text{rang}(A_{\overline{F}}, b_{\overline{F}}) \leq n - k$ , por la proposición 1.5  $\dim(F) \geq k$ .

Si estamos en el caso  $\dim(F) = k = 0$  entonces no hay ninguna cara de menor dimensión y no hay nada que probar.

Suponemos  $\dim(F) > 0$ , sea  $\hat{x}$  un punto interior de  $F$ , como  $\dim(F) > 0$  existe un punto  $y \in F$ . Consideramos la recta que los une  $\hat{x} + \lambda(y - \hat{x}) = z(\lambda)$ , donde  $\lambda \in \mathbb{R}$ , y suponemos que interseca con la recta  $a^i x = b_i$  para  $i \in \{1, \dots, m\}$ .

Consideramos  $\lambda^*$  el menor de los  $\lambda$  que hace que la recta  $z(\lambda)$  se corte con la recta  $a^i x = b_i$ . Como  $\hat{x}$  es un punto interior  $\lambda^* \neq 0$ . Y tenemos  $F_i = \{x \in P : A_{\overline{F}} x = b_{\overline{F}}, a^i x = b_i\}$  una cara de  $P$  de dimensión menor lo que es una contradicción.

Luego no puede haber caras de dimensión menor que  $F$ .

Entonces  $\hat{x} + \lambda(y - \hat{x})$  no interseca con la recta  $a^i x = b_i$  para ningún  $i \in \{1, \dots, m\}$  lo que quiere decir que  $A\hat{x} + A\lambda(y - \hat{x}) \leq b$  para todo  $\lambda \in \mathbb{R}$ .

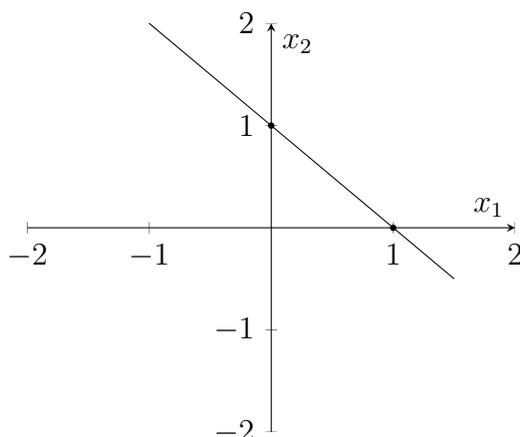


Figura 1.2: Cara de dimensión 1 del poliedro

Como  $A\hat{x} < b$  por ser punto interior, entonces  $A\lambda(y - \hat{x}) = 0$  para todo  $y \in F$ , y por lo tanto  $F = \{y : A\hat{x} = Ay\}$  y como  $\text{rang}(A) = n - k$  por la proposición 1.5  $\dim(F) = k$ .

□

**Ejemplo 1.12.** Dado un poliedro  $P = \{x \in \mathbb{R}^2 : x_1 + x_2 \leq 1\}$ . (Figura 1.2). Tenemos que  $\text{rang}(A) = 1$ , y una cara de  $P$  de una dimensión es  $F = \{x \in \mathbb{R}^2 : x_1 + x_2 = 1\}$

**Proposición 1.13.** Sea  $P$  un poliedro racional,  $x$  es un punto extremo de  $P$  si, y solo si es una cara de dimensión cero de  $P$ .

*Demostración.* Vamos a demostrar ambas implicaciones, empezamos con  $(\Leftarrow)$ .

Sea  $x$  una cara de  $P$  de dimensión 0, por la proposición 1.5 sabemos que  $\text{rang}(A_x^-) = n$ , tomamos  $(\tilde{A}, \tilde{b})$  una submatriz de  $(A, b)$ , con  $\tilde{A}$  de dimensión  $n \times n$ .

De esta manera podemos escribir  $x = \tilde{A}^{-1}\tilde{b}$ .

Si  $x = \frac{1}{2}x_1 + \frac{1}{2}x_2$  con  $x_1, x_2 \in P$ , y  $\tilde{A}x_i \leq \tilde{b}$  para  $i = 1, 2$ , entonces  $\tilde{A}x_1 = \tilde{A}x_2 = \tilde{b}$ , es decir que  $x_1 = x_2 = x$  y por lo tanto  $x$  es un punto extremo de  $P$ .

Para probar la otra implicación vamos a probar: no b  $\Rightarrow$  no a, y así tendremos la implicación que buscamos (a  $\Rightarrow$  b).

Suponemos que  $x$  no es una cara de  $P$  de dimensión cero. Por la proposición 1.5  $\text{rang}(A_x^-) < n$ , pero ahora existe un  $y \neq 0$  que satisface  $A_x^-y = 0$ , y para un  $\epsilon > 0$  lo suficientemente pequeño tenemos que  $x^1 = x + \epsilon y \in P$  y  $x^2 = x - \epsilon y \in P$ . Ahora tenemos  $x = \frac{1}{2}x^1 + \frac{1}{2}x^2$  y por lo tanto no es punto extremo.

□

**Proposición 1.14.** Sea  $P$  es un poliedro racional no vacío, entonces  $d$  es dirección extrema de  $P$  si, y solo si el conjunto  $\{\lambda d : \lambda \in \mathbb{R}\}$  es una cara de dimensión uno de  $P$ .

*Demostración.* Vamos a probar las dos implicaciones, comenzamos por ( $\Leftarrow$ )  
 Sea  $A_d^- = \{a^i : i \in \{1, \dots, m\}, a^i d = 0\}$ . Si  $\{\lambda d : \lambda \geq 0\}$  es una cara de dimensión uno de  $P$ ,  $\text{rang}(A_d^-) = n - 1$ . Por lo tanto todas las soluciones de  $A_d^- y = 0$  son de la forma  $y = \lambda d, \lambda \in \mathbb{R}$ . Si  $d = \frac{1}{2}d^1 + \frac{1}{2}d^2$  llegamos a una contradicción como la proposición 1.12 y por lo tanto  $d$  es dirección extrema.

Probamos la otra implicación al igual que en la proposición anterior, no  $b \Rightarrow$  no  $a$ , y así tendremos la implicación que buscamos ( $a \Rightarrow b$ ).

Sea  $d$  un vector que satisface  $Ad \leq 0$ , y  $\text{rang}(A_d^-) < n - 1$ , entonces existe  $\hat{d} \neq \lambda d, \lambda \in \mathbb{R}$  tal que  $A_d^- \hat{d} = 0$ . Las direcciones  $d^1 = d + \epsilon \hat{d}, d^2 = d - \epsilon \hat{d}$  nos muestran que  $d = \frac{1}{2}d^1 + \frac{1}{2}d^2$  y por lo tanto  $d$  no es dirección extrema.

□

De las dos proposiciones anteriores se deduce lo siguiente.

**Corolario 1.15.** *Todo poliedro racional tiene un número finito de puntos extremos y direcciones extremas.*

**Observación 1.** El número de puntos extremos de un poliedro racional, aunque sea finito, puede llegar a ser muy grande.

**Proposición 1.16.** *Sea  $P \neq \emptyset$  un poliedro racional y acotado, y consideramos el problema de programación lineal  $\min\{cx : x \in P\}$ , entonces existe una solución óptima para este problema que es un punto extremo.*

*Demostración.* El conjunto de soluciones óptimas de  $P$  es una cara  $F = \{x \in P : cx = c_0\}$ . Por la proposición 1.11  $F$  contiene una cara de dimensión  $n - \text{rang}(A)$ , y como  $n - \text{rang}(A) = 0$ ,  $F$  contiene un punto extremo por la proposición 1.12.

□

**Proposición 1.17.** *Si  $P \neq \emptyset$  es un poliedro racional no acotado y consideramos el problema de programación lineal  $\min\{cx : x \in P\}$ , entonces  $P$  tiene una dirección extrema  $d^*$  que satisface  $cd^* > 0$ .*

*Demostración.* Dado el problema de programación lineal  $\min\{cx : x \in P\}$  existe un vector  $d \in \mathbb{R}^n$  tal que  $Ad \leq 0$  y  $cd > 0$ . Considerando el nuevo problema de programación lineal  $\min\{cd : Ad \leq 0, cd \leq 1\} = 1$ . Por la proposición 1.16, tiene una solución óptima que es un punto extremo  $\hat{d}$ .

Por lo tanto  $\hat{d}$  satisface  $c\hat{d} > 0$  y  $\text{rang}(A_{\hat{d}}^-) = n - 1$ , y por la proposición 1.14,  $\hat{d}$  es una dirección extrema de  $P$ .

□

A continuación vamos a dar el primer resultado fundamental de este capítulo.  
[11]

**Teorema 1.18.** *Sea  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  un poliedro racional. Entonces tenemos el conjunto de puntos extremos de  $P$ ,  $S_p = \{x^1, \dots, x^k\} \neq \emptyset$  y el conjunto de direcciones extremas de  $P$ ,  $S_d = \{d^1, \dots, d^l\} \neq \emptyset$  tal que todo  $x \in P$  se puede representar como combinación convexa de puntos extremos y combinación lineal no negativa de direcciones extremas,*

$$x = \sum_{j=1}^k \lambda_j x^j + \sum_{j=1}^l \nu_j d^j$$

$$\lambda_j \geq 0, \sum_{j=1}^k \lambda_j = 1, \nu_j \geq 0$$

*Si  $P$  es un poliedro acotado entonces no tiene direcciones extremas y se representa como combinación convexa de puntos extremos.*

$$x = \sum_{j=1}^k \lambda_j x^j$$

$$\lambda_j \geq 0, \sum_{j=1}^k \lambda_j = 1$$

*Demostración.* Sea  $Q = \{x : x = \sum_{j=1}^k \lambda_j x^j + \sum_{j=1}^l \nu_j d^j, \lambda_j \geq 0, \sum_{j=1}^k \lambda_j = 1, \nu_j \geq 0\}$ .

Como  $x^j \in P$  con  $j = 1, \dots, k$  y  $P$  es convexo tenemos que  $\hat{x} = \sum_{j=1}^k \lambda_j x^j$  para todo  $\lambda_j$  que satisface  $\lambda_j \geq 0, \sum_{j=1}^k \lambda_j = 1$ .

Además, como para cada  $j = 1, \dots, l$  se tiene que  $d^j$  son direcciones extremas de  $P$ , es decir  $\hat{x} + \sum_{j=1}^l \nu_j d^j \in P$  para todo  $\nu_j \geq 0$  con  $j = 1, \dots, l$ .  
Por lo tanto tenemos que  $Q \subset P$ .

Queremos ver que  $P = Q$ , para ello vamos a razonar por reducción al absurdo.

Suponemos que  $Q \subsetneq P$ , entonces existe un  $y \in P \setminus Q$ , es decir no existen  $\lambda_k$  ni  $\nu_l$  que cumplan las propiedades pedidas:

$$y = \sum_{j=1}^k \lambda_j x^j + \sum_{j=1}^l \nu_j d^j$$

$$\lambda_j \geq 0, \sum_{j=1}^k \lambda_j = 1, \nu_j \geq 0$$

Esto quiere decir que existen  $c$  y  $c_0$  tal que  $cx^k \leq c_0$ ,  $cd^l \leq 0$  y  $cy \leq c_0$ .

Si consideramos el problema de programación lineal  $\min\{cx : x \in P\}$  tenemos dos casos:

- Caso 1: El poliedro racional  $P$  es acotado, y por la proposición 1.16 existe una solución óptima que es un punto extremo, por lo que  $y \in P$ ,  $cy > cx^k$  para todo  $x^k \in P$  que son puntos extremos de  $P$ . Por lo que llegamos a una contradicción.
- Caso 2:  $P$  es no acotado y aplicando la proposición 1.17 existe una dirección extrema  $d^*$  tal que  $cd^* > 0$  y de nuevo llegamos a una contradicción.

En ambos casos se llega a una contradicción y por lo tanto podemos concluir que  $P = Q$ .

□

Gracias a este resultado concluimos que se puede dar una representación finita de todo poliedro, ya sea acotado o no.

**Observación 2.** Existe una equivalencia entre el número finito de puntos extremos y soluciones básicas que se aplica en el método simplex [1] para la resolución de problemas de programación lineal.

## 1.2. Envolverte convexa en programación entera

En esta sección daremos una clasificación de los diferentes problemas de programación entera y definiciones básicas que utilizaremos a lo largo del resto de capítulos.

Dado un problema de programación lineal  $\{cx : Ax \leq b, x \geq 0\}$  que se quiere maximizar o minimizar, con  $A$  una matriz de dimensiones  $m \times n$  y  $b$  un vector columna de dimensión  $m$  de coeficientes racionales, podemos añadir algunas restricciones sobre las variables, como por ejemplo  $x \in \mathbb{Z}^n$ .

De esta forma obtenemos un problema de programación entera y en función de las condiciones que demos sobre las variables tendremos distintos tipos de problemas. [10]

- Programación entera mixta:  
Cuando no todas las variables  $x \geq 0$  son enteras hablamos de problemas de programación entera mixta y se puede formular de la siguiente manera.

$$\min\{cx + hy : Ax + Gy \leq b, x \geq 0, y \geq 0, y \in \mathbb{Z}^p\}$$

Donde  $A$  es una matriz  $m \times n$ ,  $G$  una matriz  $m \times p$ ,  $x$  un vector de dimensión  $n$  e  $y$  de dimensión  $p$ .

- Programación entera pura:

En este caso todas las variables son enteras y estamos en el problema de programación entera pura

$$\text{mín}\{cx : Ax \leq b, x \geq 0, x \in \mathbb{Z}^n\}$$

- Programación entera binaria:

Si las variables son binarias, es decir, toman valores 0-1 entonces hablamos de problemas de programación entera binaria.

$$\text{mín}\{cx : Ax \leq b, x \in \{0, 1\}^n\}$$

Los problemas de programación lineal se resuelven de manera eficiente pues existen métodos simples para resolver problemas de grandes dimensiones, fundamentalmente el método simplex [1] y se tienen teoremas y resultados que funcionan muy bien, debido sobre todo a los teoremas de dualidad.

Sin embargo esto no ocurre con la programación entera ya que la región factible no es convexa, si no discreta.

Conocemos el problema de programación entera  $\text{mín}\{cx : Ax \leq b, x \geq 0, x \in \mathbb{Z}^n\}$ , que denotaremos por  $P_E$  y consideramos el conjunto  $S \subseteq \mathbb{Z}^n$  de puntos factibles  $S = \{x \in \mathbb{Z}^n, x \geq 0 : Ax \leq b\}$ .

**Definición 1.19.** Dado el conjunto factible de un problema de programación entera  $S \subset \mathbb{R}^n$ , la envolvente convexa de  $S$ ,  $\text{conv}(S)$  se define como el conjunto de todos los puntos que son combinación convexa de un número finito de puntos de  $S$ ,  $\text{conv}(S) = \{x : x = \sum_{i=1}^t x^i \lambda_i, \sum_{i=1}^t \lambda_i = 1, \lambda_i \geq 0, i = 1, \dots, t, \{x^1, \dots, x^t\} \in S\}$ . [10]

**Proposición 1.20.** La envolvente convexa de  $S$  es el conjunto convexo más pequeño que contiene a  $S$ .

*Demostración.* Supongamos que  $C$  es un conjunto convexo que contiene a  $S$ . Queremos probar que  $\text{conv}(S) \subset C$ . Por definición de  $C$  todo punto  $x_i$  de  $S$  pertenece a  $C$ . Dado que  $C$  es convexo, cualquier combinación convexa de puntos de  $S$  también debe pertenecer a  $C$ , es decir cualquier punto de la forma  $\sum_{i=1}^t x^i \lambda_i$  con  $x_i \in S$  y  $\lambda_i \geq 0$  está en  $C$ . Por lo tanto  $\text{conv}(S) \subset C$ .

Como  $\text{conv}(S)$  es convexo esta contenido en cualquier conjunto convexo que contenga a  $S$ . Luego es el conjunto convexo más pequeño que contiene a  $S$ .

□

Un problema central en programación entera es encontrar un conjunto de desigualdades que describan la envolvente convexa, es decir  $\text{conv}(S) = \{x : Hx < h\}$  con  $H$  matriz de dimensiones muy grandes, y esto se resuelve de manera teórica en el siguiente teorema.[11]

**Teorema 1.21.** (de Meyer) Sea  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  un poliedro racional y  $S = P \cap \mathbb{Z}^n$  conjunto factible para el problema de programación lineal entera  $\min\{cx : Ax \leq b, x \geq 0, x \in \mathbb{Z}^n\}$ , entonces  $\text{conv}(S)$  es un poliedro racional.

*Demostración.*

*Nota.* Antes de dar la demostración de este teorema vamos a ver un resultado previo.

**Proposición 1.22.** Sea  $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\} \neq \emptyset$  y  $S = P \cap \mathbb{Z}^n$ , entonces existe un conjunto finito de puntos  $\{q^l\}_{l \in L}$  de  $S$  y un conjunto finito de direcciones  $\{d^j\}_{j \in J}$  de  $P$  tal que

$$S = \{x \in \mathbb{R}^n : x = \sum_{l \in L} \alpha_l q^l + \sum_{j \in J} \beta_j d^j, \sum_{l \in L} \alpha_l = 1, \alpha_l \in \mathbb{Z}^{|L|}, \beta_j \in \mathbb{Z}^{|J|}, \alpha, \beta > 0\}$$

*Demostración.* Sea  $\{x^k \in \mathbb{R}^n : k \in K, x^k \geq 0\}$  un conjunto finito de puntos extremos de  $P$  y sea  $\{d^j \in \mathbb{R}^n : j \in J, d^j \geq 0\}$  conjunto finito de direcciones extremas de  $P$ . Como  $P$  es un poliedro racional, todos estos vectores extremos tienen coordenadas racionales. Tenemos

$$P = \{x \in \mathbb{R}^n : x = \sum_{k \in K} \lambda_k x^k + \sum_{j \in J} \mu_j d^j, \sum_{k \in K} \lambda_k = 1, \lambda_k, \mu_j \geq 0\}$$

Sin pérdida de generalidad podemos suponer que  $\{d^j\}$  con  $j \in J$  son vectores enteros. Sea

$$Q = \{x \in \mathbb{Z}^n : x = \sum_{k \in K} \lambda_k x^k + \sum_{j \in J} \mu_j d^j, \sum_{k \in K} \lambda_k = 1, \lambda_k \geq 0, 0 \leq \mu_j < 1, x > 0\}$$

Entonces  $Q$  es un conjunto finito y lo denotamos por  $Q = \{q^l \in \mathbb{Z}^n : l \in L\}$ , y  $Q \subseteq S$ .

Ahora observamos que  $x^i \in S$  si y solo si  $x^i \in \mathbb{Z}^n$  y

$$x^i = \left\{ \sum_{k \in K} \lambda_k^i x^k + \sum_{j \in J} (\mu_j^i - |\mu_j^i|) d^j \right\} + \left\{ \sum_{j \in J} |\mu_j^i| d^j \right\}, \sum_{k \in K} \lambda_k = 1, \lambda_k, \mu_j \geq 0$$

para  $j \in J, k \in K$ .

El primer término de la suma anterior es un punto de  $Q$ , luego existe  $l(i) \in L$  tal que  $x^i = q^{l(i)} + \sum_{j \in J} \beta_j^i d^j, \beta_j^i = |\mu_j^i|$  para todo  $j \in J$ . Y concluimos.

□

Ahora ayudándonos del resultado anterior procedemos a demostrar el Teorema de Meyer.

Como todo punto  $x^i \in S$  se puede escribir de la forma  $x^i = q^{l(i)} + \sum_{j \in J} \beta_j^i d^j$  por el resultado anterior, entonces toda combinación convexa de puntos  $\{x^i \in S : i \in I\}$  se puede escribir como

$$\begin{aligned} x &= \sum_{i \in I} \gamma_i x^i = \sum_{i \in I} \gamma_i (q^{l(i)} + \sum_{j \in J} \beta_j^i d^j) \\ &= \sum_{l \in L} \left( \sum_{i \in I: l(i)=l} \gamma_i \right) q^l + \sum_{j \in J} \left( \sum_{i \in I} \gamma_i \beta_j^i \right) d^j \\ &= \sum_{l \in L} \alpha_l q^l + \sum_{j \in J} \beta_j d^j \end{aligned}$$

donde  $\alpha_j = \sum_{i \in I: l(i)=l} \gamma_i \geq 0$  con  $l \in L$  y  $\sum_{l \in L} \alpha_l = \sum_{i \in I} \gamma_i = 1$  y  $\beta_j = \sum_{i \in I} \gamma_i \beta_j^i \geq 0$  para  $j \in J$ .

Por lo tanto  $\text{conv}(S) = \{x \in \mathbb{R}^n : x = \sum_{l \in L} \alpha_l q^l + \sum_{j \in J} \beta_j d^j \text{ con } x > 0, \beta_j, \alpha_l \geq 0 \text{ para } l \in L, j \in J \text{ y } \sum_{l \in L} \alpha_l = 1\}$ . Con  $q^l, r^j \in \mathbb{Z}^n$  para  $l \in L, j \in J$ .

Luego  $\text{conv}(S)$  es un poliedro racional.

□

**Proposición 1.23.** *Los puntos extremos de  $\text{conv}(S)$  pertenecen a  $S$ .*

Gracias a estos resultados podemos dar la descripción de la envolvente convexa,  $P_I = \text{conv}(S)$ , de manera que  $P_I \subset P$ , y como consecuencia del Teorema de Meyer podemos escribir  $P_I = \{x : Hx \leq h\}$  con  $H$  matriz de coeficientes enteros y dimensiones muy grandes.

Cuando  $\text{conv}(S)$  sea un poliedro acotado lo denotaremos como politopo.

En general  $\text{conv}(S)$  tiene una estructura muy complicada, y de ahí la dificultad de los problemas de programación entera.

El Teorema de Meyer no es más que un resultado teórico, que nos garantiza un conjunto de desigualdades para describir la envolvente convexa, en la práctica este conjunto puede llegar a ser muy grande y por lo tanto no es tan fácil dar dicha descripción.

Se puede tener una idea pero normalmente será una cantidad enorme (exponencial) de desigualdades y no es simple obtenerlas.

Aun así se conoce una familia de ejemplos con una sola restricción y dos variables y con un número ilimitado de puntos extremos de  $\text{conv}(S)$ .

Cuando  $S$  es el conjunto de soluciones de un problema combinatorial, normalmente las variables son  $x_j = 0, 1$  y  $\text{conv}(S)$  es un politopo. Se le suele dar el nombre del

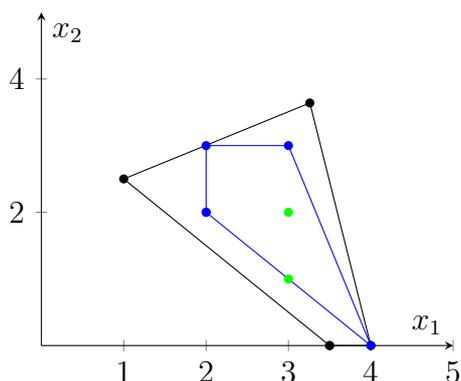


Figura 1.3: Envoltente convexa de S

politopo del problema, por ejemplo; politopo de la mochila, politopo de acoplamiento o politopo del viajante.

**Ejemplo 1.24.** Sea  $S = \{(x_1, x_2) \in \mathbf{Z}^2 :$

$$\begin{aligned} -x_1 + 2x_2 &\leq 4, \\ 5x_1 + x_2 &\leq 20, \\ -2x_1 - 2x_2 &\leq -7 \\ x_1 &\geq 0, x_2 \geq 0 \end{aligned}$$

Por lo tanto  $S = \{(2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (4, 0)\}$  y de estos seis puntos solo cuatro son puntos extremos de  $\text{conv}(S)$ . (ver Figura 1.3)

En este caso es fácil dar una descripción de  $\text{conv}(S)$  por desigualdades lineales.

$\text{conv}(S) = \{(x_1, x_2) \in \mathbf{Z}^2 :$

$$\begin{aligned} -x_1 &\leq -2, \\ x_2 &\leq 3, \\ -x_1 - x_2 &\leq -4 \\ 3x_1 + x_2 &\leq 12 \\ x_1 &\geq 0, x_2 \geq 0 \end{aligned}$$



# Capítulo 2

## Problemas bien resueltos

### 2.1. Poliedros enteros

En este capítulo continuamos con el estudio del conjunto de puntos factibles para un problema de programación lineal entera.

Dado el conjunto de puntos factibles  $S = \{x \in \mathbb{Z}^n : Ax \leq b\}$  donde  $A$  es una matriz de dimensiones  $m \times n$  y  $b$  un vector columna de dimensión  $m$ , ambos de coeficientes enteros, consideramos el problema de programación entera  $P_E$

$$\text{mín}\{cx : x \in S\}$$

A lo largo de este capítulo vamos a estudiar los casos en los que se conoce la descripción de la envolvente convexa de  $S$ . Conociendo  $\text{conv}(S)$  podemos encontrar un algoritmo eficiente para resolver el problema de programación lineal, y por tanto el problema  $P_E$ .

**Definición 2.1.** Sea  $P \subset \mathbb{R}^n$  un poliedro no vacío, decimos que es un poliedro entero si cada una de sus caras no vacías contiene un punto entero. [11]

**Proposición 2.2.** Sea  $P \neq \emptyset$  un poliedro  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  con  $\text{rang}(A) = n$ ,  $P$  es entero si, y solo si todos sus puntos extremos son enteros.

*Demostración.* Sea  $P$  un poliedro entero, con  $\text{rang}(A) = n$ , por la proposición 1.11, deducimos que  $P$  tiene una cara de dimensión  $n - \text{rang}(A) = 0$ , luego nos fijamos en sus caras mínimas no vacías, cada una de ellas es un punto extremo (proposición 1.13), y aplicando la definición de poliedro entero son puntos enteros.

Sea  $x \in P$  un punto extremo entero, cada punto extremo es una cara de dimensión cero de  $P$  y por lo tanto contienen un punto entero, luego  $P$  es un poliedro entero.

□

Del resultado anterior se deduce lo siguiente.

**Corolario 2.3.** *Sea  $P \neq \emptyset$  un poliedro  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ,  $P$  es entero si, y solo si todos sus puntos extremos son enteros.*

Consideramos el problema de programación lineal para el poliedro  $P$ ,  $z_{P_L} = \min\{cx : x \in P\}$ . Los poliedros enteros pueden caracterizarse a partir de la solución del problema  $P_L$ . [11]

**Proposición 2.4.** *Las siguientes afirmaciones son equivalentes:*

1.  $P$  es un poliedro entero.
2. El problema  $P_L$  tiene una solución óptima entera para todo  $c \in \mathbb{R}^n$  para el cual tiene solución óptima.
3. El problema  $P_L$  tiene una solución óptima entera para todo  $c \in \mathbb{Z}^n$  para el cual tiene solución óptima.
4.  $z_{P_L}$  es entero para todo  $c \in \mathbb{Z}^n$  para el cual  $P_L$  tiene solución óptima.

*Demostración.* 1  $\Rightarrow$  2: Si el problema  $P_L$  tiene una solución óptima, entonces dicha solución es un punto extremo de  $P$ , por el teorema 1.17 del capítulo anterior.

2  $\Rightarrow$  3  $\Rightarrow$  4 son evidentes.

4  $\Rightarrow$  1: Dado  $\min\{cx : x \in P\}$  con  $P$  entero y el problema  $(P_L)$  con solución óptima, entonces  $x \in P$  es punto extremo, y tenemos  $c \in \mathbb{Z}^n$  tal que  $x$  es la única solución óptima de  $P_L$  por la proposición 1.16 del capítulo anterior.

Vamos a probar no 1  $\Rightarrow$  no 4

Si  $P$  no es entero  $\Rightarrow$  existe  $\tilde{c} \in \mathbb{Z}^n$  tal que  $x'$  es la única solución óptima de  $P_L$  y alguna de las componentes de  $x'$ , por ejemplo  $(x_j)'$  no es entera.

Entonces podemos tener un  $q \in \mathbb{Z}^n$  tal que  $x'$  es también solución óptima para  $c' = \tilde{c} + \frac{1}{q}c_j$  y para  $qc' = q\tilde{c} + c_j$ . Pero  $qc'x' - q\tilde{c}x' = (x_j)'$  lo que quiere decir que  $\min\{cx : x \in P\}$  no es entero y por lo tanto no se cumple 4.

Hemos probado 1  $\Rightarrow$  2  $\Rightarrow$  3  $\Rightarrow$  4  $\Rightarrow$  1

□

**Definición 2.5.** Un sistema de desigualdades lineales  $Ax \leq b$  decimos que es entero totalmente dual, TDI, si para todo  $c \in \mathbb{Z}^n$  tal que  $z_{P_L} = \min\{cx : Ax \leq b\}$  es finito, entonces su dual  $\max\{yb : yA = c, y \in \mathbb{R}^n, y \geq 0\}$  tiene solución óptima entera.

**Corolario 2.6.** Si  $Ax \leq b$  es TDI y  $b \in \mathbb{Z}^n$  entonces  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  es entero.

*Demostración.* Si  $Ax \leq b$  es TDI quiere decir que su dual tiene solución óptima, y como  $b \in \mathbb{Z}^n$  tenemos que  $z_{P_L}$  del dual es también entero.

Por tanto, para todo  $c$  para el cual  $z_{P_L}$  es finito se tiene que  $z_{P_L}$  es entero y por la proposición anterior  $P$  es entero.

□

**Ejemplo 2.7.** Dada una partición de  $V = \{1, \dots, n\}$  el conjunto de nodos,  $V_1 = \{1, \dots, m\}$  y  $V_2 = \{m+1, \dots, n\}$  tal que  $V = V_1 \cup V_2$ , los pesos de cada nodo los denotamos por  $c_j$  para  $j \in V$  y los pesos de los ejes por  $b_{ij}$  para  $i \in V_1$  y  $j \in V_2$ .

Tenemos el problema  $\min\{\sum_{j=1}^n c_j x_j : x_i + x_j \leq b_{ij} \forall i \in V_1, j \in V_2\}$ .

Su dual es

$$\begin{aligned} & \text{máx}\left\{\sum_{i=1}^m \sum_{j=m+1}^n b_{ij} y_{ij} : \right. \\ & \left. \sum_{j=m+1}^n y_{ij} = c_i, \forall i \in V_1 \right. \\ & \left. \sum_{i=1}^m y_{ij} = c_j, \forall j \in V_2, y_{ij} \geq 0, \forall i, j \in V\right\} \end{aligned}$$

y tiene solución óptima entera si es factible y si  $c_j$  es entero  $\forall j \in V$ . Por lo tanto el sistema  $x_i + x_j \leq b_{ij}$  es TDI y  $b_{ij}$  son enteros, luego  $P = \{x \in \mathbb{R}^n : x_i + x_j \leq b_{ij} \text{ con } i \in V_1, j \in V_2\}$  es un poliedro entero.

Este ejemplo es un tipo de problema de transporte que estudiaremos más adelante.

**Proposición 2.8.** Si  $Ax \leq b$  es un sistema lineal de coeficientes racionales, entonces existe un entero positivo,  $q \in \mathbb{Z}$  tal que  $\frac{1}{q}Ax \leq \frac{1}{q}b$  es TDI.

*Demostración.* Tenemos el problema dual  $yA = c$ ,  $y \in \mathbb{R}^n$ ,  $c \in \mathbb{Z}^n$ .

Podemos escribir todo punto extremo  $y$  de la forma  $y = \frac{1}{q}(p_1, \dots, p_n)$  donde  $p_i$  son enteros.

Ahora sea  $y' = qy$  por lo tanto todo punto extremo de  $\frac{1}{q}y'A = c$ ,  $y' \in \mathbb{R}^n$  es entero y el sistema dual  $\frac{1}{q}Ax \leq \frac{1}{q}b$  es TDI.

□

Del resultado anterior se deduce el siguiente corolario.

**Corolario 2.9.** *Todo poliedro  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  se puede representar por un sistema de desigualdades lineales TDI.*

Sin embargo no siempre la descripción con un sistema de desigualdades lineales de un poliedro entero es TDI.

**Ejemplo 2.10.** El problema

$$\text{mín}\{x_{12} + x_{13} + x_{14} + x_{23} + x_{24} + x_{34} :$$

$$x_{12} + x_{13} + x_{14} \geq 1$$

$$x_{13} + x_{23} + x_{24} \geq 1$$

$$x_{13} + x_{23} + x_{34} \geq 1$$

$$x_{14} + x_{24} + x_{34} \geq 1\}$$

Tiene tres soluciones óptimas :

- $x_{12} = x_{34} = 1$  y  $x_{ij} = 0$  para el resto de variables
- $x_{13} = x_{24} = 1$  y  $x_{ij} = 0$  para el resto de variables
- $x_{14} = x_{23} = 1$  y  $x_{ij} = 0$  para el resto de variables

Su dual es el problema máx $\{\sum_{i=1}^4 y_i : y_i + y_j \leq 1$  para todo  $i, j = \{1, \dots, 4\} : j > i, y \in \mathbb{R}^4\}$ .

Su única solución es  $y_i = \frac{1}{2}$  para todo  $i \in \{1, \dots, 4\}$ , por lo tanto el sistema lineal  $Ax \geq 1, x \geq 0$  no es TDI, pero el poliedro  $P = \{x \in \mathbb{R}^4 : Ax \geq 1, x \geq 0\}$  si es entero.

## 2.2. Matrices unimodulares

Dado un problema de programación lineal entera  $P_E$  de la forma mín $\{cx : Ax \leq b, x \in \mathbb{Z}^n, x \geq 0\}$  con  $A$  y  $b$  de coeficientes enteros, la idea principal para resolver dicho problema es dar un problema relajado  $P_R$  y estudiar si tiene una solución óptima entera.

*Nota.* En el siguiente capítulo estudiamos con más detalle cómo obtener un problema relajado para encontrar la solución óptima del problema de programación lineal entera  $P_E$ .

**Observación 3.** Sabemos de programación lineal que las soluciones factibles básicas toman la forma  $x = (x_B, x_N) = (B^{-1}b, 0)$  donde  $B$  es una matriz  $m \times n$  no singular, submatriz de  $(A, I)$ .

**Observación 4.** Si  $B$ , base óptima, satisface que  $\det(B) = \pm 1$  entonces el problema de programación lineal relajado  $P_R$  resuelve el problema  $P_E$ .

*Demostración.* Por la regla de Cramer podemos escribir  $B^{-1} = B^* \frac{1}{\det(B)}$  donde  $B^*$  es la matriz adjunta. Los términos de  $B^*$  son productos de términos de  $B$ , por lo tanto  $B^*$  es entera y como  $\det(B) = \pm 1$ , entonces  $B^{-1}$  también es entera. Por lo tanto  $B^{-1}b$  es entera para  $b$  entero.

□

Ahora nos preguntamos cuándo las bases óptimas satisfacen que  $\det(B) = \pm 1$ .

**Definición 2.11.** Una matriz  $A$  es totalmente unimodular, TU, si toda submatriz cuadrada de  $A$  tiene determinante 0, 1 o -1.[11]

El siguiente resultado recoge como reconocer las matrices TU.[9] [11]

**Proposición 2.12.** *Las siguientes afirmaciones son equivalentes:*

1.  $A$  es TU.
2. La traspuesta de  $A$ ,  $A^T$ , es TU.
3.  $(A, I)$  es TU.
4. Una matriz obtenida eliminando una columna (o fila) de  $A$  es TU.
5. Una matriz obtenida multiplicando una columna (o fila) de  $A$  por -1 es TU.
6. Una matriz obtenida intercambiando dos columnas (o filas) de  $A$  es TU.
7. Una matriz obtenida duplicando una columna (o fila) de  $A$  es TU.
8. Una matriz obtenida por una operación pivote en  $A$  es TU.

*Demostración.*  $1 \Leftrightarrow 2$ ,  $1 \Leftrightarrow 3$ ,  $1 \Leftrightarrow 4$ ,  $1 \Leftrightarrow 5$ ,  $1 \Leftrightarrow 6$  y  $1 \Leftrightarrow 7$  son evidentes.

Veamos  $1 \Leftrightarrow 8$

Pivotar en una matriz  $(0,1,-1)$  incluye los siguientes pasos.

1. Si  $a_{ij} = -1$  entonces la fila  $i$  la multiplicamos por -1 y tenemos la nueva fila  $\bar{a}^i$ .

2. Para  $k = i$  tenemos

$$\bar{a}^k = \begin{cases} a^k & \text{si } a_{kj} = 0 \\ a^k - \bar{a}^i & \text{si } a_{kj} = 1 \\ a^k + \bar{a}^i & \text{si } a_{kj} = -1 \end{cases}$$

Sea  $B$  submatriz de  $A$  y  $\bar{B}$  la matriz que obtenemos tras realizar una operación pivote sobre  $B$ . Veamos que  $\bar{B}$  es una matriz  $(0,1,-1)$ .

Para ellos diferenciamos distintos casos.

- Si la fila  $i$  de  $A$  está en  $B \rightarrow |\det(\bar{B})| = |\det(B)|$  y ya hemos terminado.

- Si la columna  $j$  de  $A$  está en  $B$  pero la fila  $i$  no  $\rightarrow \det(\bar{B}) = 0$  y hemos terminado.
- ni la fila  $i$  ni la columna  $j$  de  $A$  están en  $B$ . Entonces definimos

$$C = \begin{pmatrix} a_{ij} & a_{i2} & \cdots & a_{1p} \\ a_{2j} & & & \\ \vdots & & B & \\ a_{kj} & & & \end{pmatrix}, \bar{C} = \begin{pmatrix} 1 & \bar{a}_{i2} & \cdots & \bar{a}_{ip} \\ 0 & & & \\ \vdots & & \bar{B} & \\ 0 & & & \end{pmatrix}$$

Por lo tanto tenemos  $|\det(\bar{B})| = |\det(\bar{C})| = |\det(C)|$ , y hemos terminado.  $\square$

**Observación 5.** Si  $A$  es una matriz totalmente unimodular, entonces  $a_{ij} \in \{0, 1, -1\}$  para todo  $i, j$ .

**Proposición 2.13.**  $A$  es TU si y solo si  $P(b) = \{x \in \mathbb{R}^n : Ax \leq b\}$  es entero para todo  $b \in \mathbb{Z}^m$  tal que  $P(b) \neq \emptyset$ .

*Demostración.* ( $\Rightarrow$ ): Consideramos el problema de programación lineal con restricciones  $Ax + Iy = b$  con  $x \in \mathbb{R}^n, y \in \mathbb{R}^m, x, y \geq 0$  donde  $A$  es TU y  $b$  es entero. Sea  $(A, I) = (A_B, A_N)$  donde  $A_B$  es matriz base para dicho problema. Por el apartado (8) de la proposición anterior,  $A_B^{-1}$  es entera. Por lo tanto  $A_B^{-1}b$  es entera y los puntos extremos se corresponden con las soluciones factibles básicas, luego  $P(b)$  es entero.

( $\Leftarrow$ ): Sea  $A_1$  una matriz arbitraria no singular de  $A$  de dimensión  $k \times k$  y sea

$$\tilde{A} = \begin{pmatrix} A_1 & 0 \\ A_2 & I_{m-k} \end{pmatrix}$$

submatriz de  $(A, I)$  no singular generada por  $A_1$  tomando los vectores  $m - k$  apropiados de  $I$ . Sea  $b = \tilde{A}z + e_i$  donde  $z \in \mathbb{Z}^m$  y

$$e_i = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{pmatrix}$$

el vector de  $I$  con un 1 en la fila  $i$ . Entonces se tiene que  $\tilde{A}^{-1}b = z + \tilde{a}_i^{-1}$ , donde  $\tilde{a}_i^{-1}$  es la columna  $i$ -ésima de la matriz  $\tilde{A}^{-1}$ .

Tomamos  $z$  tal que  $z + \tilde{a}_i^{-1} \geq 0$ . Por lo tanto  $z + \tilde{a}_i^{-1}$  es un vector de variables básicas y puntos extremos de  $P(b)$ . Por hipótesis,  $z + \tilde{a}_i^{-1} \in \mathbb{Z}^m$  y  $z \in \mathbb{Z}^m$ , es decir  $\tilde{a}_i^{-1} \in \mathbb{Z}^m$  y  $\tilde{A}^{-1}$  es una matriz entera, luego  $A_1^{-1}$  es una matriz entera.

Finalmente  $\det(A_1)$  y  $\det(A_1^{-1})$  son enteros tales que

$$|\det(A_1)| \cdot |\det(A_1^{-1})| = |\det(A_1 A_1^{-1})| = 1$$

Es decir  $|\det(A_1)| = 1$ .

□

**Observación 6.** Como la traspuesta de  $A$  es TU, el poliedro dual es también entero. Es decir,

$$Q(c) = \{u \in \mathbb{R}^m : uA \geq c, c \in \mathbb{Z}^n\}$$

es poliedro entero.

En el siguiente teorema se recoge la caracterización de las matrices totalmente unimodulares, junto con algunas condiciones suficientes como corolarios. [10][9]

**Teorema 2.14.** *La matriz  $A$  es totalmente unimodular si y solo si para todo  $J \subset N = \{1, \dots, n\}$  existe una partición  $J_1, J_2$  que satisface  $|\sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij}| \leq 1$  para cada  $i \in \{1, \dots, m\}$ .*

*Demostración.* ( $\Rightarrow$ ): Sea  $J$  un subconjunto de  $N$  arbitrario. Definimos  $z$  tal que

$$z_j = \begin{cases} 1 & \text{si } j \in J \\ 0 & \text{si } j \notin J \end{cases}$$

Y sean  $d' = 0, d = z, g = Az$  y  $b'_i = b_i = \frac{1}{2}g_i$  si  $g_i$  es par y  $b'_i = \frac{1}{2}(g_i - 1), b_i = b'_i + 1$  si  $g_i$  es impar.

Consideramos  $P(b, b', d, d') = \{x \in \mathbb{R}^n : b' \leq Ax \leq b, d' \leq x \leq d, x \geq 0\}$ . Observamos que  $x = \frac{z}{2} \in P(b, b', d, d')$ .

Como  $A$  es TU, tenemos que  $b', b \in \mathbb{Z}^m$  y  $d, d' \in \mathbb{Z}^n$ , luego  $P(b, b', d, d')$  es un poliedro entero por la proposición 2.13.

Por lo tanto existe  $x^0 \in P \cap B$  con  $x_j^0 = 0$  para  $j \in N \setminus J$  y  $x_j^0 \in \{0, 1\}$  si  $j \in J$ .

Observamos que  $z_j - 2x_j^0 = \pm 1$  para  $j \in J$ . Sea  $J_1 = \{j \in J : z_j - 2x_j^0 = 1\}$  y  $J_2 = \{j \in J : z_j - 2x_j^0 = -1\}$ . Tenemos

$$\sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij} = \sum_{j \in J} a_{ij}(z_j - 2x_j^0) =$$

$$\begin{cases} g_i - g_i = 0 & \text{si } g_i \text{ es par} \\ g_i - (g_i \pm 1) = \pm 1 & \text{si } g_i \text{ es impar} \end{cases}$$

Y por lo tanto tenemos

$$|\sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij}| \leq 1, i = 1, \dots, m$$

( $\Leftarrow$ ): Si  $|J| = 1$  tenemos que  $a_{ij} \in \{0, \pm 1\}$  para todo  $i, j$ . Por lo que razonamos por inducción sobre la dimensión de las submatrices de  $A$  no singulares utilizando la hipótesis de que toda submatriz  $(k-1) \times (k-1)$  de  $A$  tiene determinante 0, 1 o -1.

Sea  $B$  una submatriz no singular de  $A$  de dimensiones  $k \times k$  y sea  $r = |\det(B)|$ . El objetivo es ver que  $r = 1$ .

Aplicando la hipótesis de inducción y la regla de Cramer tenemos que  $B^{-1} = \frac{B^*}{r}$  donde  $b_{ij}^* \in \{0, \pm 1\}$ . Por definición de  $B^*$  tenemos que  $Bb_1^* = re_1$  donde  $b_1^*$  es el primer vector columna de  $B^*$ .

Sea  $J = \{i : b_{i1}^* \neq 0\}$  y  $J'_1 = \{i \in J : b_{i1}^* = 1\}$ . Entonces para  $i = 2, \dots, k$ , tenemos

$$(Bb_1^*)_i = \sum_{j \in J'_1} b_{ij} - \sum_{j \in J \setminus J'_1} b_{ij} = 0$$

.

Por lo tanto  $|\{i : b_{ij}^* \neq 0\}|$  es par, es decir, para cualquier partición  $(\tilde{J}_1, \tilde{J}_2)$  de  $J$ , luego  $\sum_{j \in \tilde{J}_1} b_{ij} - \sum_{j \in \tilde{J}_2} b_{ij}$  es par para  $i = 2, \dots, k$ .

Ahora por hipótesis, existe una partición  $(J_1, J_2)$  de  $J$  tal que  $|\sum_{j \in \tilde{J}_1} b_{ij} - \sum_{j \in \tilde{J}_2} b_{ij}| \leq 1$ . Es decir que

$$\sum_{j \in \tilde{J}_1} b_{ij} - \sum_{j \in \tilde{J}_2} b_{ij} = 0$$

para  $i = 2, \dots, k$ .

Consideramos  $\alpha_1 = |\sum_{j \in \tilde{J}_1} b_{1j} - \sum_{j \in \tilde{J}_2} b_{1j}|$ . Si  $\alpha_1 = 0$ , definimos  $y \in \mathbb{R}^k$  como  $y_i = 1$  para  $i \in J_1$ ,  $y_i = -1$  para  $i \in J_2$  y  $y_i = 0$  si  $i \notin J_1, J_2$ . Como  $By = 0$  y  $B$  es no singular,  $y = 0$ , lo que contradice que  $J \neq \emptyset$ .

Luego por hipótesis tenemos  $\alpha_1 = 1$  y  $By = \pm e_1$ . Por lo tanto,  $Bb_1^* = re_1$ . Como  $y$  y  $b_1^*$  son vectores con elementos  $(0, 1, -1)$ , entonces  $b_1^* = \pm y$  y  $|r| = 1$ .

□

Ahora volviendo al problema de programación lineal entera  $\min\{cx : x \in \mathbb{Z}^n, Ax \leq b, x \geq 0\}$ , si  $A$  es una matriz TU entonces el problema relajado  $P_R$  resuelve el problema original.

**Proposición 2.15.** *El problema de programación lineal  $P_L$ ,  $\min\{cx : x \in \mathbb{R}^n, Ax \leq b, x \geq 0\}$  tiene solución óptima entera para todo vector entero  $b$  si, y solo si  $A$  es TU.*

Con este resultado podemos concluir que para un problema de programación lineal entera  $P_E$  con matriz  $A$ , TU, se conoce una descripción de la envolvente convexa del conjunto de soluciones factibles,  $\text{conv}(S) = P$ .

## 2.3. Problemas de flujo de redes y de transporte

En esta sección del capítulo trataremos dos tipos de problemas que se pueden resolver mediante programación lineal entera.

### 2.3.1. Problema de flujo de redes

Estudiamos el problema del flujo de redes. [1]

Dado un grafo  $D = (V, A)$  donde  $V = \{1, \dots, m\}$  es el conjunto finito de nodos y  $A = \{(i, j), (k, l), \dots, (s, t)\}$  el conjunto finito de arcos que unen los nodos, (el arco  $(i, j)$  une el nodo  $i$  con el nodo  $j$ ).

Cada nodo  $i$  tiene asociado un valor  $b_i$ , que llamamos demanda si  $b_i < 0$  o suministro disponible si  $b_i > 0$ . Podemos clasificar los nodos en función del valor de  $b_i$ :

- Si  $b_i > 0$  el nodo  $i$  lo llamamos fuente.
- Si  $b_i < 0$  el nodo  $i$  lo llamamos sumidero.
- Si  $b_i = 0$  no hay ni suministros disponibles ni demanda y el nodo  $i$  lo llamamos nodo de transferencia o intermedio.

Asociado a cada arco  $(i, j)$  tenemos  $x_{ij}$  la cantidad de flujo que hay en dicho arco, ( $x_{ij} \geq 0$ ) y  $c_{ij}$  el coste de flujo a lo largo de dicho arco.

Suponemos que la oferta total es igual a la demanda total dentro de la red, es decir  $\sum_{i=1}^m b_i = 0$ .

El problema del coste mínimo del flujo de redes consiste en encontrar un flujo factible que satisfaga todas las demandas en un coste mínimo, y lo formulamos de la siguiente forma

$$\begin{aligned} \text{mín} \{ & \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} : \sum_{j=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = b_i \\ & i = 1, \dots, m, x_{ij} \geq 0, i, j = 1, \dots, m \}. \end{aligned} \quad (2.1)$$

Donde la ecuación (2.1) la denominamos ecuación de “conservación de flujo”, e indica que el flujo ni se crea ni se destruye, si no que se conserva.

En la ecuación de conservación, el sumando  $\sum_{j=1}^m x_{ij}$  representa el flujo total que sale del nodo  $i$ , y el sumando  $\sum_{k=1}^m x_{ki}$  el flujo total que entra en el nodo  $i$ . Esto sugiere que el flujo neto que sale del nodo  $i$ ,  $\sum_{j=1}^m x_{ij} - \sum_{k=1}^m x_{ki}$ , debe ser igual a  $b_i$ .

Luego, en particular, si  $b_i < 0$  entonces debe de haber más flujo que entre en  $i$  del que sale. Este problema lo denominamos “incapacitado”, mientras que si el problema incluye cotas superiores en los arcos lo denominamos “capacitado”.

El problema del coste mínimo del flujo de redes puede sugerir una red logística en la que personas y materiales se desplazan entre distintos puntos del mundo. Puede estar asociado al movimiento de locomotoras entre puntos de una red ferroviaria para satisfacer la energía de los trenes y minimizar los costes del viaje.

Los problemas del coste mínimo del flujo de redes aparecen en el diseño y análisis de sistemas de comunicación, problemas de distribución y suministro de cadenas, y en una gran variedad de áreas.

Claramente el problema de coste mínimo del flujo de redes es un problema de programación lineal que puede resolverse de diferentes maneras, una de ellas es aplicando el algoritmo simplex.[1]

**Ejemplo 2.16.** Consideramos el problema siguiente

Minimizar

$$2x_{12} - 5x_{13} - x_{23} + 4x_{24} + 6x_{32} + 3x_{34} + 7x_{41}$$

sujeto a

$$\begin{aligned} x_{12} + x_{13} - x_{41} &\leq 4, \\ -x_{12} + x_{23} + x_{24} - x_{32} &\leq 2, \\ -x_{13} - x_{23} + x_{32} + x_{34} &\leq -1 \\ -x_{24} - x_{34} + x_{41} &\leq -5 \\ x_{ij} &\geq 0 \end{aligned}$$

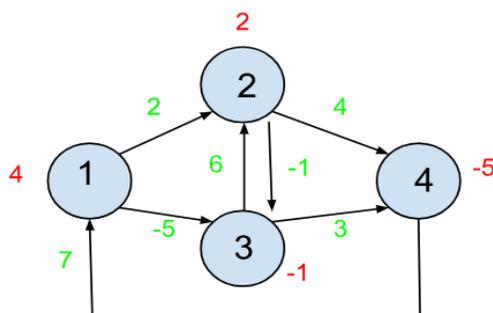


Figura 2.1: Problema de flujo de redes

En la figura 2.1 aparecen los 4 nodos, con los valores  $b_i$  en rojo y el coste del flujo,  $c_{ij}$ , en verde.

Además en este problema podemos ver que no hay más de 4 y 2 unidades de suministro disponible en los nodos 1 y 2 respectivamente, y hay al menos 1 y 5 unidades de demanda en los nodos 3 y 4.

Con esto obtenemos el problema de la figura 2.2

$x_{12}$	$x_{13}$	$x_{23}$	$x_{24}$	$x_{32}$	$x_{34}$	$x_{41}$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	b
1	1	0	0	0	0	-1	1	0	0	0	0	4
-1	0	1	1	-1	0	0	0	0	1	0	0	2
0	-1	-1	0	1	1	0	0	0	-1	0	0	-1
0	0	0	-1	0	-1	1	0	0	0	-1	0	-5
0	0	0	0	0	0	0	-1	-1	1	1	1	0

Tabla 2.1: Tabla método simplex

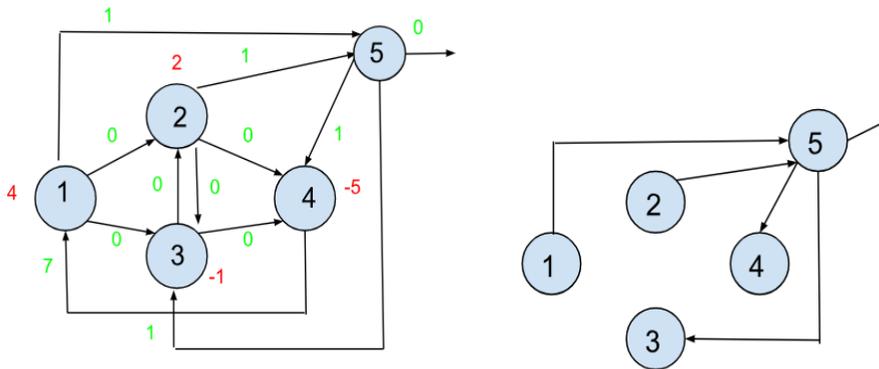


Figura 2.2: Primera fase del método simplex y base factible

### 2.3.2. Problema de transporte

A continuación estudiamos el problema de transporte.[11].

Dado un conjunto de nodos  $V = \{1, 2, \dots, n\}$  y un conjunto de arcos  $A = \{(i, j), \dots, (k, l)\}$ . Los nodos representan las ciudades y los arcos determinan el par de ciudades que están conectadas por un viaje directo.

Para cada arco  $(i, j) \in A$ ,  $c_{ij}$  es la duración (en tiempo) del viaje entre la ciudad  $i$  y la ciudad  $j$ .

En este tipo de problemas buscamos un circuito, que empiece en la ciudad 1, que pase una sola vez por cada ciudad y que regrese a la ciudad 1 en el menor tiempo posible.

Para formular el problema introducimos las variables  $x_{ij} = 1$  si las ciudades  $j$  e  $i$  van seguidas en el circuito y  $x_{ij} = 0$  en caso contrario.

De manera que el problema de transporte es de la forma

Minimizar

$$\sum_{(i,j) \in A} c_{ij} x_{ij}$$

sujeto a

$$\sum_{i:(i,j) \in A} x_{ij} = 1, j \in V, \quad \sum_{j:(i,j) \in A} x_{ij} = 1, i \in V$$

$$\sum_{(i,j) \in A: i \in U, j \in V \setminus U} x_{ij} \leq 1$$

con  $U \subset V$  y  $2 \leq |U| \leq |V| - 2$

Este problema también se puede formular como un problema de flujo de redes, tomando el grafo bipartito  $D = (V_1 \cup V_2, A)$  donde  $V_1 = \{1, \dots, m_1\}$  es el conjunto de fuentes,  $V_2 = \{m_1 + 1, \dots, m\}$  el conjunto de sumideros y  $A = \{(i, j) : i \in V_1, j \in V_2\}$ . Sin pérdida de generalidad podemos suponer que hay un arco que va de cada nodo “demanda” a cada nodo “suministro”.

El coste de transporte por unidad del nodo  $i \in V_1$  al nodo  $j \in V_2$  lo denotamos por  $w_{ij}$  y si no hay un arco que una dichos nodos tomamos un valor lo suficientemente grande para  $w_{ij}$ .

Tendremos que el nodo  $i$  tendrá un valor de suministro  $a_i > 0$  y el nodo  $j$  un valor de demanda  $b_j > 0$ .

Por lo tanto tenemos una condición necesaria para el problema,  $\sum_{i \in V_1} a_i = \sum_{j \in V_2} b_j$ .

Una vez descrito el problema de transporte como uno de flujo de redes:

Minimizar

$$\sum_{i \in V_1} \sum_{j \in V_2} w_{ij} x_{ij}$$

sujeto a

$$\sum_{j \in V_2} x_{ij} = a_i, i \in V_1, \sum_{i \in V_1} x_{ij} = b_j, j \in V_2, x \in \mathbb{R}^n, x > 0$$

el objetivo es encontrar un flujo que satisfaga las ecuaciones de suministro y demanda con un coste mínimo.

## Capítulo 3

# Descripción parcial de $\text{conv}(S)$ mediante desigualdades

### 3.1. Relajaciones y desigualdades válidas

A continuación consideramos el problema de programación lineal entera  $P_E$ ,  $z_{P_E} = \min\{cx : x \in S\}$  con  $S = \{x \in \mathbb{Z}^n : Ax \leq b, x \geq 0\}$ , que podemos escribir de forma equivalente como  $z_{P_E} = \min\{cx : x \in \text{conv}(S)\}$ , pues según el Teorema de Meyer visto en el capítulo 1, la envolvente convexa es un poliedro racional descrito de la forma  $\text{conv}(S) = \{x : Hx \leq h\}$ .

Por lo tanto, en teoría se podrá obtener una solución óptima de todo problema de programación lineal entera. Sin embargo hemos visto que en la práctica es imposible obtener una descripción total de  $\text{conv}(S)$  por lo que vamos a ver familias de desigualdades que nos dan una descripción parcial.

El objetivo es encontrar una buena cota inferior para  $z_{P_E}$  sin necesidad de resolver  $P_E$ , que denominaremos “relajación lineal fuerte”.

**Ejemplo 3.1.** Dado un conjunto  $S = \{(1, 1), (2, 2), (3, 1), (1, 2), (2, 2), (3, 2), (2, 3)\}$ . En la Figura 3.4 observamos dos formulaciones diferentes de  $S$ ,  $P_1$  y  $P_2$ , pero ninguna de ellas es la descripción ideal de  $\text{conv}(S)$ .

$$P_1 = \{x \in \mathbb{R}^4 : 0 \leq x \leq 1, 83x_1 + 61x_2 + 49x_3 + 20x_4 \leq 100\}$$

$$P_2 = \{x \in \mathbb{R}^4 : 0 \leq x \leq 1, 4x_1 + 3x_2 + 2x_3 + x_4 \leq 4\}$$

$$\text{conv}(S) = \{x \in \mathbb{R}^4 : 0 \leq x \leq 1, 4x_1 + 3x_2 + 2x_3 + x_4 \leq 4, x_1 + x_2 + x_3 \leq 1, x_1 + x_4 \leq 1\}$$

Todas estas formulaciones contienen a la formulación ideal de  $S$ ,  $\text{conv}(S)$ . Habrá veces que no consigamos dar la descripción de  $\text{conv}(S)$  y que por lo tanto nos quedemos con una descripción como  $P_1$  o  $P_2$ .

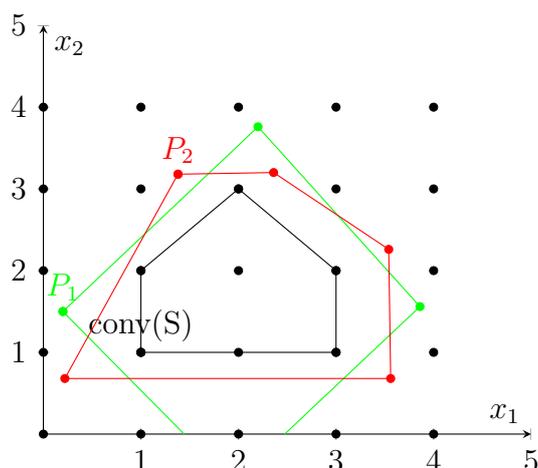


Figura 3.1:  $P_1$  es mejor formulación de  $S$  que  $P_2$

En primer lugar vamos a estudiar lo que es una relajación de restricción. Veremos que es un tipo habitual de relajación o problema relajado que consiste en omitir alguna restricción de las que definen el problema original  $P_E$ , obteniendo así un conjunto  $S_R \supset S$  factible de  $P_R$ .

**Definición 3.2.** Dado un problema  $P_E$ ,  $z_{P_E} = \min\{cx : x \in S\}$ , una relajación de restricciones para este problema es otro problema de minimización  $P_R$  relajado,  $z_{P_R} = \min\{cx : x \in S_R\}$  con  $S \subseteq S_R$ . [11]

Dentro de las relajaciones de restricción tenemos diferentes tipos, como la relajación lineal:

- Simple: consiste en omitir la restricción de que las variables sean enteras. Es decir, pasamos a un problema  $P_L$  que ya sabemos resolver.
- Fuerte: consiste en añadir a la relajación lineal simple una serie de restricciones adicionales que llamamos "desigualdades válidas o planos de corte".

También existen otro tipo de relajaciones que no son de restricción:

- Relajación Lagrangiana: muy utilizada en  $P_E$
- Relajación de grupo
- Relajación subrogada

En cualquier caso se pasa del problema original  $P_E$  que es difícil de resolver, al problema relajado  $P_R$  más fácil de resolver.

A continuación vamos a enunciar una serie de propiedades, pues una vez resuelto el problema relajado se puede extraer información sobre el problema original. [10][11]

**Proposición 3.3.** *Si  $P_R$  es no factible entonces el problema original  $P_E$  es no factible.*

*Demostración.* Como  $S \subseteq S_R$  por definición de problema relajado, si  $S_R = \emptyset$  entonces  $S = \emptyset$ .

□

**Proposición 3.4.** *Si  $P_E$  es factible, entonces  $z_{P_R} \leq z_{P_E}$ , es decir, el valor óptimo de un problema relajado de minimización es una cota inferior del valor óptimo.*

*Demostración.* Tenemos  $z_{P_E}$  finito y  $\hat{x}$  solución óptima de  $P_E$ . Entonces  $z_{P_E} = c\hat{x} \leq z_{P_R}(\hat{x}) \leq z_{P_R}$ .

Ambas desigualdades se obtienen de la definición de  $P_R$  y  $\hat{x} \in S \subset S_R$ .

Por lo tanto si  $z_{P_E} = \infty$ , entonces  $z_{P_R} = \infty$ .

□

**Proposición 3.5.** *Sea  $\hat{x} \in S_R$  la solución óptima de  $P_R$ . Si  $\hat{x} \in S$  entonces  $\hat{x}$  es solución óptima de  $P_E$ , es decir, si la solución óptima de  $P_R$  es factible para  $P_E$ , entonces es solución óptima de  $P_E$ .*

*Demostración.* Sea  $\hat{x} \in S$ , solución óptima de  $P_R$  entonces  $z_{P_E} \geq c\hat{x} = z_{P_R}$ , como  $z_{P_R} \leq z_{P_E}$  entonces  $z_{P_R} = z_{P_E}$  y por lo tanto  $\hat{x}$  es solución óptima de  $P_E$ .

□

**Observación 7.** En el caso de los problemas de maximización la cota que se obtiene es superior  $z_{P_R} \geq z_{P_E}$  y se calcula con métodos heurísticos. Nos vamos a centrar en la obtención de la cota inferior.

**Definición 3.6.** Una desigualdad  $\pi x \leq \pi_0$  decimos que es válida para el conjunto para  $S$  si todo  $x \in S$  la satisface.

**Proposición 3.7.** *Sea  $S = P \cap \mathbb{Z}^n$  el conjunto factible del problema  $P_E$ , y sea  $\text{conv}(S)$  la envolvente convexa. Si  $\pi x \leq \pi_0$  es una desigualdad válida para  $S$  entonces es también válida para  $\text{conv}(S)$ .*

**Definición 3.8.** Dadas dos desigualdades válidas para el conjunto  $S$ ,  $\pi x \leq \pi_0$  y  $cx \leq c_0$ , decimos que son equivalentes si  $(c, c_0) = \lambda(\pi, \pi_0)$  con  $\lambda > 0$ .

Si no son equivalentes y existe un  $\mu > 0$  tal que  $c \geq \mu\pi$  y  $c_0 \leq \mu\pi_0$ , entonces  $\{x \in \mathbb{R}^n : cx \leq c_0\} \subset \{x \in \mathbb{R}^n : \pi x \leq \pi_0\}$  y decimos que  $cx \leq c_0$  domina o es más fuerte que  $\pi x \leq \pi_0$ .

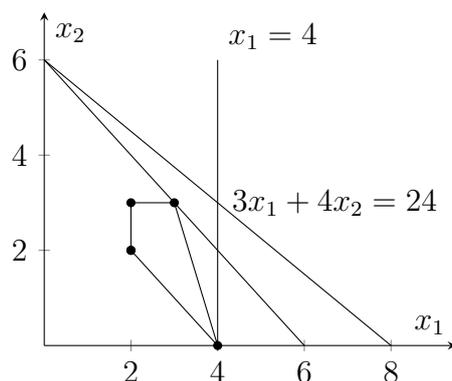


Figura 3.2: desigualdades válidas fuertes

**Ejemplo 3.9.** Si retomamos el ejemplo 1.22 del capítulo 1, en el que concluíamos que  $\text{conv}(S) = \{(x_1, x_2) \in \mathbb{Z}^2 :$

$$\begin{aligned} -x_1 &\leq -2, \\ x_2 &\leq 3, \\ -x_1 - x_2 &\leq -4 \\ 3x_1 + x_2 &\leq 12 \\ x_1 \geq 0, x_2 &\geq 0 \end{aligned}$$

Podemos observar en la figura 3.2 que la desigualdad válida  $3x_1 + 4x_2 \leq 24$  está dominada por la desigualdad válida  $x_1 + x_2 \leq 6$ .

La desigualdad válida  $x_1 \leq 4$  es una cara de dimensión 0 que también está dominada por una desigualdad válida, en este caso  $3x_1 + x_2 \leq 12$ .

Como hemos explicado anteriormente, el objetivo de introducir desigualdades válidas en un problema de optimización  $P_E$  es reducir la región factible del problema  $P_R$ , y simplificar así su resolución. Por ello, de entre todas las posibles desigualdades que podemos dar, solo nos interesan aquellas que eliminen la solución actual del problema  $P_R$  sin eliminar soluciones enteras.

Con esta idea introducimos el concepto de corte.

**Definición 3.10.** Sea  $\hat{x}$  solución óptima de  $P_R$ , entonces decimos que una desigualdad válida  $\pi x \leq \pi_0$  es un corte, o plano de corte, (Figura 3.3) si elimina a la solución  $\hat{x}$ , es decir, si  $\pi \hat{x} > \pi_0$ . [11]

**Definición 3.11.** Dado  $S \subset \mathbb{R}^n$  el conjunto factible y dos formulaciones  $P_1 = \{x \in \mathbb{R}^n : A^1 x \leq b^1\}$  y  $P_2 = \{x \in \mathbb{R}^n : A^2 x \leq b^2\}$  de  $S$ , con  $S \subset P_1$  y  $S \subset P_2$ , entonces decimos que  $P_1$  es mejor formulación que  $P_2$  si  $z_{P_2} \leq z_{P_1} \leq z_{P_E}$ .

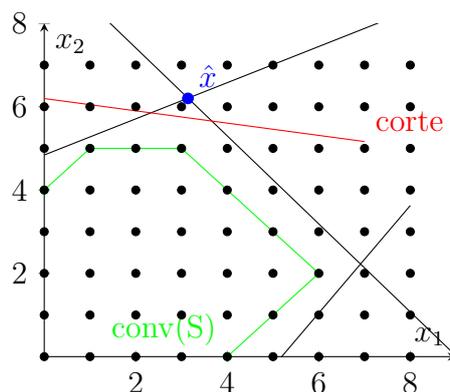


Figura 3.3: Representación de corte

### 3.2. Algoritmo genérico de planos de corte

Dado que no podemos dar una descripción completa de  $\text{conv}(S)$ , aplicando el algoritmo de planos de corte [8] vamos a ir consiguiendo una descripción parcial de  $\text{conv}(S)$  mediante lo que hemos definido como “cortes”.

Cada iteración de corte comienza resolviendo un problema relajado y si la solución óptima es factible en  $P_E$ , es decir, es entera, entonces termina el algoritmo por la proposición 3.5.

La idea es generar nuevas relajaciones del problema original  $P_E$  hasta obtener la solución óptima o llegar a una relajación lineal fuerte.

*Notación.* Sea  $z_P$  el valor óptimo de un problema  $P$ :

- $z_P = +\infty$  si el problema  $P$  es un problema de minimizar no factible.
- $z_P = -\infty$  si el problema  $P$  es un problema de minimizar no acotado.

A continuación definimos las diferentes etapas del algoritmo.

Sean  $z_{P_E}$  el valor óptimo del problema  $P_E$  con conjunto factible  $S$ . En cualquier iteración se resuelve un problema relajado  $P_R$  con  $S_R$  el conjunto factible asociado y se obtiene una sucesión de la forma

$$S_{R^1} \supset S_{R^2} \supset \dots \supset S_{R^t} \supset \dots$$

**Inicialización:** Tomamos  $t = 1$ , y consideramos un problema relajado lineal simple  $P_R$  con  $S_{R^1} = \{x : Ax \leq b, 0 \leq x_j \leq 8\}$ ,  $S \subseteq S_{R^1}$ .

Pasamos a la iteración genérica,  $t$ :

- **Etapa 1:** Resolvemos el problema relajado actual  $P_{R^t} = \min\{cx : x \in S_{R^t}\}$  y obtenemos una solución óptima para este problema  $\hat{x}^t$  tal que  $z^t = c\hat{x}^t$ .
- **Etapa 2:** Esta etapa la llamamos “Test de optimalidad”, pues lo que hacemos es comprobar si la solución óptima de  $P_{R^t}$  obtenida en la etapa anterior  $\hat{x}^t$  es solución óptima para el problema original  $P_E$ .
  - si  $\hat{x}^t$  es entera, entonces también es solución óptima para  $P_E$  por la proposición 3.5 y hemos terminado.
  - si  $\hat{x}^t$  no es entera pasamos a la siguiente etapa.
- **Etapa 3:** Esta etapa tiene el nombre de “problema separador”. Buscamos una nueva desigualdad válida  $\pi x \leq \pi_o$  para  $S$  tal que  $\pi\hat{x}^t > \pi_o$ , y definimos  $S_{R^{t+1}} = S_{R^t} \cap \{x : \pi x \leq \pi_o\}$  de manera que  $S_{R^{t+1}} \subset S_{R^t}$  y pasamos de nuevo a la etapa 1.

Si no encontramos una desigualdad válida que satisfaga  $\pi\hat{x}^t > \pi_o$  entonces en este caso se termina el algoritmo sin llegar al óptimo, pero obtenemos una relajación lineal fuerte con valor óptimo  $z_{P_{R^t}}$ .

A lo largo del algoritmo vamos obteniendo una sucesión de valores óptimos de la forma

$$z_{P_{R^1}} \leq z_{P_{R^2}} \leq \dots \leq z_{P_{R^t}} \leq z_{P_E}, \forall t \in \mathbb{N}$$

Donde  $z_{P_{R^t}}$  es la cota inferior del valor óptimo del problema original  $z_{P_E}$ .

Este algoritmo, dependiendo de las familias de desigualdades que se tengan, acabará en un valor óptimo entero o en una relajación lineal fuerte.

Idealmente un algoritmo de planos de corte debería terminar en la solución óptima del problema  $P_E$ , es decir terminaría en la etapa 2.1.

Pero para la mayoría de familias de desigualdades no se llega al óptimo, es decir, se llega hasta la etapa 3.2.

### 3.3. Algunas familias de desigualdades

Como ya hemos dicho, uno de los problemas principales es que no podemos dar una descripción completa mediante familias de desigualdades de  $\text{conv}(S)$ .

Si embargo, como hemos visto en la sección anterior, 3.2, para una familia de desigualdades dada se obtiene una familia de relajaciones cada vez más fuertes, que en algunos casos acaba en el óptimo y en otros, simplemente, en una relajación lineal fuerte.

Hay una gran variedad de problemas interesantes para los cuales se han obtenido familias de desigualdades fuertes. En esta sección vamos a dar una breve descripción de algunas de ellas.

A continuación enumeramos algunas de las familias de desigualdades más conocidas.

- Desigualdades de cubrimiento
- Desigualdades de cubrimiento de flujo
- Desigualdades disyuntivas
- Desigualdades de aritmética modular
- Desigualdades de Chvatal-Gomory de redondeo
- Desigualdades para problemas específicos.  
Para cualquier problema de combinatoria está muy estudiado cómo encontrar desigualdades válidas para dicho problema. Como por ejemplo
  - Problema de localización
  - Problema del agente viajero
  - Problema de tipo mochila

### 3.3.1. Desigualdades de cubrimiento

Las desigualdades de cubrimiento son, en particular, las que se dan para el problema de programación lineal de tipo mochila.

Aprovechando la estructura del problema se puede desarrollar una familia de desigualdades fuertes y eficientes.[11]

Suponemos que tenemos un problema con una de las restricciones del problema de tipo mochila de la forma

$$\text{máx} \sum_{j=1}^n c_j x_j$$

sujeto a

$$\sum_{j=1}^n a_j x_j \leq b, x_j \in \{0, 1\}, j = 1, 2, \dots, n$$

Donde  $b > 0$  es la capacidad de la mochila, y disponemos de  $n$  objetos que podemos introducir en ella.

Para cada objeto  $j$  tenemos  $a_j > 0$  el espacio que ocupa y  $c_j > 0$  el beneficio que proporciona.

El objetivo del problema es elegir un conjunto de objetos que no superen la capacidad máxima y que aporten los máximos beneficios.

De manera que la región factible de este problema es

$$S = \{x \in \{0, 1\}^n : \sum_{j=1}^n a_j x_j \leq b\}$$

A continuación estudiamos la definición de cubrimiento.[10]

**Definición 3.12.** Un conjunto  $C \subset N$  es un cubrimiento para  $S$  si  $\sum_{j \in C} a_j > b$ . Diremos que un cubrimiento es minimal si  $C \setminus \{j\}$  no es cubrimiento para ningún  $j \in C$ .

**Proposición 3.13.** Si  $C \subseteq N$  es un cubrimiento para  $S$ , la desigualdad de cubrimiento

$$\sum_{j \in C} x_j \leq |C| - 1$$

es válida para  $S$ . [10]

*Demostración.* Dado un subconjunto  $R \subset N = \{1, 2, \dots, n\}$ , definimos el vector indicador  $x^R \in \{0, 1\}^n$  tal que  $x_j^R = 1$  si  $j \in R$  y  $x_j^R = 0$  si  $j \notin R$ . Para todo  $x \in S$  se tiene que  $x_j = 1$  si  $j \in R$  un subconjunto de  $N$ .

Entonces, sea  $x \in S$ ,  $x = x^R$ , vamos a probar que si  $x^R$  no satisface la desigualdad, entonces  $x^R \notin S$ .

Si  $\sum_{j \in C} x_j^R > |C| - 1$ , entonces  $|R \cap C| = |C|$  y por lo tanto  $C \subseteq R$ . Luego  $\sum_{j=1}^n a_j x_j^R = \sum_{j \in R} a_j \geq \sum_{j \in C} a_j > b$  y por tanto  $x^R \notin S$ .

□

**Ejemplo 3.14.** Dada la región factible de un problema

$$S = \{x \in \{0, 1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$$

Algunos de los cubrimientos minimales para  $S$  y sus restricciones correspondientes son,

$$\begin{aligned} C = \{1, 2, 3\} &: x_1 + x_2 + x_3 \leq 2, \\ C = \{1, 2, 6\} &: x_1 + x_2 + x_6 \leq 2, \\ C = \{1, 5, 6\} &: x_1 + x_5 + x_6 \leq 2 \\ C = \{3, 4, 5, 6\} &: x_3 + x_4 + x_5 + x_6 \leq 3 \end{aligned}$$

Además observamos que hay una forma simple de hacer más fuertes estas desigualdades de cubrimiento. Existen diferentes procedimientos para hacer estas desigualdades de cubrimiento más fuertes, como la extensión, que estudiamos a continuación.

**Definición 3.15.** La extensión  $E(C)$  de un cubrimiento minimal  $C$  es el conjunto  $C \cup \{j \in N \setminus C : a_j \geq a_i \text{ para todo } i \in C\}$ . [11]

**Proposición 3.16.** Si  $C$  es un cubrimiento de  $S$ , la desigualdad de cubrimiento extendida

$$\sum_{j \in E(C)} x_j \leq |C| - 1$$

es válida para  $S$ . [10]

*Demostración.* La demostración es similar a la de la proposición 3.14.

**Ejemplo 3.17.** Siguiendo con el ejemplo anterior, la desigualdad de cubrimiento extendida para  $C = \{3, 4, 5, 6\}$  es  $x_1 + x_2 + x_3 + x_4 + x_4 + x_6 \leq 3$ . Por lo que podemos decir que la desigualdad de cubrimiento  $x_3 + x_4 + x_5 + x_6 \leq 3$  está dominada por la desigualdad  $x_1 + x_2 + x_3 + x_4 + x_4 + x_6 \leq 3$ .

### Lifting (levantamiento) de desigualdades de cubrimiento

En general, dado un cubrimiento, buscamos los mejores valores posibles de  $\alpha_j$  con  $j \in N \setminus C$  tal que la desigualdad

$$\sum_{j \in C} x_j + \sum_{j \in N \setminus C} \alpha_j x_j \leq |C| - 1$$

sea válida para  $S$ .

**Ejemplo 3.18.** En el ejemplo anterior podemos ver como la desigualdad  $x_1 + x_2 + x_3 + x_4 + x_4 + x_6 \leq 3$  está dominada por  $2x_1 + x_2 + x_3 + x_4 + x_4 + x_6 \leq 3$ . Veamos como lo hemos obtenido.

Consideramos el cubrimiento  $C = \{3, 4, 5, 6\}$  y claramente observamos que cuando  $x_1 = x_2 = x_7 = 0$ , la desigualdad  $x_3 + x_4 + x_4 + x_6 \leq 3$  es válida para  $S = \{x \in \{0, 1\}^4 : 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19\}$

Ahora tomando  $x_2 = x_7 = 0$ , buscamos valores de  $\alpha_1$  que cumplan que la desigualdad

$$\alpha_1 x_1 + x_3 + x_4 + x_4 + x_6 \leq 3$$

es válida para  $S = \{x \in \{0, 1\}^5 : \alpha_1 x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19\}$ .

Estudiamos los casos en los que  $x_1 \neq 0$ .

Si  $x_1 = 1$ , la desigualdad es válida si y solo si  $\alpha_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 3$  es válida para todo  $x \in \{0, 1\}^4$  que satisface  $6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19 - 11$ .

De manera equivalente podemos decir que la desigualdad es válida si y solo si  $\alpha_1 \leq 3 - M$  donde  $M = \max\{x_3 + x_4 + x_5 + x_6 : 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 8, x \in \{0, 1\}^4\}$ .

Sea  $M = 1$  en el punto  $x = (0, 0, 0, 1)$ , entonces  $\alpha_1 \leq 2$ . Entonces la desigualdad es válida para  $\alpha_1 \leq 2$  y  $\alpha_1 = 2$  nos da la desigualdad fuerte.

A continuación describimos un procedimiento que nos conduce a tales valores  $\alpha_j$  y, de hecho nos proporciona una desigualdad válida para  $\text{conv}(S)$  cuando  $C$  es cubrimiento minimal.

Dados  $j_1, \dots, j_r$  un conjunto ordenado de  $N \setminus C$ . Comenzamos el procedimiento con  $t = 1$ .

Tenemos la desigualdad válida  $\sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$  obtenida anteriormente.

Para calcular el mayor valor de  $\alpha_{j_t}$  para el cual la desigualdad  $\alpha_{j_t} x_{j_t} + \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$  sea válida, resolvemos una de las restricciones del problema de tipo mochila:

$$M_t = \text{máx} \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j$$

$$\sum_{i=1}^{t-1} a_{j_i} x_{j_i} + \sum_{j \in C} a_j x_j \leq b - a_{j_t}$$

$$x \in \{0, 1\}^{|C|+t-1}$$

Tomamos  $\alpha_{j_t} = |C| - 1 - M_t$  y paramos si  $t = r$ .

**Ejemplo 3.19.** Consideramos  $C = \{3, 4, 5, 6\}$ ,  $j_1 = 1, j_2 = 2$  y  $j_3 = 7$ , en el ejemplo anterior hemos calculado  $\alpha_1 = 2$ . Continuando con la variable  $x_2$ ,  $M_2 = \text{máx}\{2x_1 + x_3 + x_4 + x_5 + x_6 : 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19 - 6 = 13, x \in \{0, 1\}^5\} = 2$  y por lo tanto  $\alpha_{j_2} = \alpha_2 = 3 - 2 = 1$ .

Razonando de forma similar se obtiene  $M_3 = 3$  y por tanto  $\alpha_{j_3} = \alpha_7 = 0$ , y terminamos el procedimiento definiendo la desigualdad

$$2x_1 + 1x_2 + x_3 + x_4 + x_5 + x_6 + 0x_7 \leq 3$$

Las desigualdades de cubrimiento se han integrado en todos los programas de resolución de programación lineal entera desde la década de los 80. En particular, el trabajo de Crowder, Harlan and Johnson, Ellis L and Padberg, Manfred, "Solving large-scale zero-one linear programming problems" [2] se considera un punto de inflexión en la resolución de problemas de programación entera. A partir de este trabajo, se empezaron a obtener restricciones fuertes lo que supuso una mejora en la resolución de problemas de programación entera.

Este estudio no se centra en teoremas ni resultados teóricos importantes, simplemente presenta diferentes procedimientos computacionales para resolver problemas de programación lineal entera.

### 3.3.2. Aritmética modular

En esta sección vamos a encontrar desigualdades válidas para el conjunto de soluciones enteras no negativas de un sistema lineal.

Sea

$$S = \{x \in \mathbb{Z}^n : \sum_{j=1}^n a_j x_j = a_0, x_j \geq 0, a_j \in \mathbb{R}\}$$

Fijamos un número  $d$  entero positivo y definimos el conjunto

$$S_d = \{x \in \mathbb{Z}^n : \sum_{j=1}^n a_j x_j = a_0 + kd, x \geq 0, k \in \mathbb{Z}\}$$

De manera que  $S \subseteq S_d$ . Vamos a dar una desigualdad válida para  $S_d$ , que también será válida para  $S$ .

Es una desigualdad que no es trivial.

Tomamos  $a_j = b_j + \alpha_j d$  para  $j = 0, 1, \dots, n$  donde  $0 \leq b_j < d$  y  $\alpha_j \in \mathbb{Z}$ , es decir,  $b_j$  es el resto de dividir  $a_j$  por  $d$ .

Entonces  $S_d = \{x \in \mathbb{Z}^n : \sum_{j=1}^n b_j x_j = b_0 + kd, x \geq 0, k \in \mathbb{Z}\}$ .

Pues  $\sum_{j=1}^n a_j x_j = \sum_{j=1}^n b_j x_j + d \sum_{j=1}^n \alpha_j x_j$  y esto es un múltiplo de  $d$  si y solo si es igual a  $a_0$  más un múltiplo de  $d$ .

Además  $0 \leq b_j < d$  luego  $\sum_{j=1}^n b_j x_j \geq 0$ , es decir  $b_0 + kd \geq 0$  y como  $b_0 < d$  tenemos que  $k \geq 0$ . Pues si  $k < 0$  tendríamos  $b_0 + kd < 0$ .

Y por tanto obtenemos la siguiente desigualdad válida

$$\sum_{j=1}^n b_j x_j \geq b_0$$

que es no trivial si  $d$  no divide a  $a_0$ , es decir, si  $b_0 > 0$ .

**Ejemplo 3.20.** Sea

$$S = \{2x_1 + 3x_2 + x_3 + 4x_4 = 3, x \in \mathbb{Z}^4, x \geq 0\}$$

tomamos  $d = 2$  y el corte que obtenemos es  $x_2 + x_3 = 1 \pmod{2}$ .

Como  $x_2, x_3 \in \mathbb{Z}$  y mayores que cero se deduce  $x_2 + x_3 \geq 1$  que es una desigualdad válida para el conjunto inicial  $S$ .

Además el vector  $(\frac{3}{2}, 0, 0, 0)$  cumple la relajación lineal de  $S$ , pero no cumple la de  $x_2 + x_3 \geq 1$  de manera que esta ecuación es un plano de corte.

Para  $d = 1$  se obtiene la desigualdad

$$\sum_{j=1}^n F(a_j)x_j \geq F(b)$$

dónde  $F(a_j) = a_j - \lfloor a_j \rfloor$  son las partes fraccionarias. Esta desigualdad la llamamos plano de corte de Gomory, que estudiaremos en el siguiente capítulo.

### 3.3.3. Redondeo de Chvátal-Gomory

A continuación introducimos unas desigualdades que se obtienen redondeando combinaciones lineales de las restricciones originales del problema  $P_E$ .

Dado un problema  $P_E$ ,  $z_{P_E} = \min\{cx : x \in S\}$  con  $S = \{x \in \mathbb{Z}^n : Ax \leq b, x \geq 0\}$ . Tomamos un vector  $u \in \mathbb{R}^m$  y denotamos  $A = \{a_1, \dots, a_n\}$  las columnas de la matriz. A continuación construimos una desigualdad válida siguiendo tres pasos:

1. Multiplicando por  $u$  obtenemos la desigualdad  $\sum_{j=1}^n (ua_j)x_j \leq ub$ , para todo  $u \geq 0$ .
2. Considerando las partes enteras obtenemos la desigualdad  $\sum_{j=1}^n (\lfloor ua_j \rfloor)x_j \leq ub$ , para todo  $x \geq 0$ .
3. Por último tenemos la desigualdad  $\sum_{j=1}^n (\lfloor ua_j \rfloor)x_j \leq \lfloor ub \rfloor$  que es válida para  $S$ . Y como  $x \in \mathbb{Z}^n$ ,  $\sum_{j=1}^n (\lfloor ua_j \rfloor)x_j$  es entera.

Es importante el tercer paso, en el que, gracias a haber tomado las partes enteras en el lado izquierdo, podemos redondear hacia abajo el lado derecho de la desigualdad.

La desigualdad

$$\sum_{j=1}^n (\lfloor ua_j \rfloor)x_j \leq \lfloor ub \rfloor$$

que es válida para  $S$  se añade al sistema de desigualdades que ya tenemos y vamos repitiendo el procedimiento combinando las desigualdades originales con las que se han ido generando.

Este procedimiento se llama redondeo de Chvátal-Gomory (C-G), y las desigualdades que produce se llaman desigualdades C-G.

**Ejemplo 3.21.** Sea  $S = \{(x_1, x_2) \in \mathbb{Z}^2 :$

$$\begin{aligned} & -x_1 + 2x_2 \leq 4, \\ & 5x_1 + x_2 \leq 20, \\ & \left. -2x_1 - 2x_2 \leq -7 \right\} \end{aligned}$$

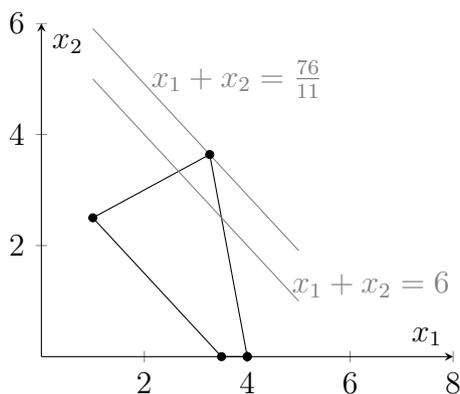


Figura 3.4: Desigualdades C-G

Tomamos  $u = (\frac{5}{11}, \frac{3}{22}, 0)$ . En el primer paso tenemos

$$A = \begin{pmatrix} -1 & 2 \\ 5 & 1 \\ -2 & -2 \end{pmatrix}, b = \begin{pmatrix} 4 \\ 30 \\ -7 \end{pmatrix}$$

Y haciendo  $uA$  obtenemos  $(\frac{5}{22}, \frac{23}{22})$  y haciendo  $ub$  tenemos  $\frac{100}{22}$ , luego la desigualdad que obtenemos es

$$\frac{5}{22}x_1 + \frac{23}{22}x_2 \leq \frac{100}{22}$$

Con el segundo paso obtenemos

$$x_2 \leq \frac{100}{22}$$

y por último nos queda la desigualdad válida

$$x_2 \leq 4$$

Tomamos ahora  $u = (\frac{4}{11}, \frac{3}{11}, 0)$  y realizando de nuevo el mismo procedimiento tendremos

1.  $x_1 + x_2 \leq \frac{76}{11}$
2. Como el lado izquierdo tiene coeficientes enteros la desigualdad no cambia.
3.  $x_1 + x_2 \leq 6$

(Figura 3.4) Es afortunado en este caso que la recta  $x_1 + x_2 = 6$  contenga un punto de  $S$ .

Algunos programas como Xpress limitan el número de cortes y algunas características de los mismos para obtener la relajación lineal fuerte en un número finito de iteraciones.

De hecho Xpress proporciona distintos tipos de cortes y estrategias para “personalizar” el algoritmo.

Estos son algunos de los cortes

- Cortes de redondeo de enteros mixtos
- Cortes de cubrimiento.
- Cortes de flujo.
- Cortes fuertes de C-G.
- Cortes de Farkas.

Y algunas estrategias

- Sin cortes.
- Estrategia de corte conservadora.
- Estrategia de corte moderada.
- Estrategia de corte agresiva.

### 3.4. Formulación fuerte y débil. Problema de la p-mediana

Como hemos visto a lo largo de este capítulo la formulación de los problemas de programación entera tiene una gran importancia a la hora de generar planos de corte para obtener la solución óptima.

También hemos visto una introducción de las formulaciones equivalentes[11] de un problema de programación entera, que, aunque varían en número de restricciones y su estructura, mantienen la solución óptima. Escoger una buena formulación puede simplificar la resolución del problema de forma significativa.

Vamos a tratar un ejemplo, en particular el problema de la p-mediana, para ilustrar las diferentes formulaciones, la formulación débil y la formulación fuerte.

#### Problema de la p-mediana

Como en otros modelos de localización, tomamos un conjunto  $M = \{1, 2, \dots, m\}$  de puntos de demanda de un cierto servicio, y otro conjunto  $N = \{1, 2, \dots, n\}$  de puntos donde es posible situar o abrir una instalación. Representamos las demandas con las variables  $h_i$  con  $i = 1, 2, \dots, m$  y conocemos la matriz de distancias de cada punto de demanda  $i$  a cada punto de instalación  $j$ ,  $(d_{ij})$ .

En este caso el modelo tiene como objetivo minimizar la distancia total que debe ser atravesada para que toda la demanda quede cubierta.

El **problema de la p-mediana** [3] es un problema de localización-asignación donde el número total de instalaciones que van a ser abiertas está prefijado a un valor  $p$ . Cada punto de demanda  $i = 1, 2, \dots, m$  tiene asignado el valor  $h_i$ , que generalmente es la población de dicho punto.

El problema consiste en decidir en qué puntos deben ser abiertas las instalaciones y cómo asignar cada punto de demanda a una instalación abierta de forma que la distancia total para cubrir todas las demandas sea mínima.

### Formulaciones del modelo

Para formular el problema introducimos variables de localización  $x_j \in \{0, 1\}$ ,  $j = 1, 2, \dots, n$  para las instalaciones, y variables de asignación  $y_{ij} \in \{0, 1\}$ , para  $i = 1, 2, \dots, m$  y  $j = 1, 2, \dots, n$  tal que  $y_{ij} = 1$  si el punto  $i$  está asignado a la instalación  $j$ , y  $y_{ij} = 0$  en caso contrario.

De esta manera formulamos el problema de la siguiente forma:

Minimizar

$$z = \sum_{i=1}^m \sum_{j=1}^n h_i d_{ij} y_{ij} \quad (3.1)$$

Sujeto a

$$\sum_{j=1}^n y_{ij} = 1, i = 1, \dots, m \quad (3.2)$$

$$\sum_{j=1}^n x_j = p \quad (3.3)$$

$$\sum_{i=1}^m y_{ij} \leq m x_j, j = 1, \dots, n \quad (3.4)$$

$$x_j \in \{0, 1\}, j = 1, \dots, n$$

$$y_{ij} \in \{0, 1\}, i = 1, \dots, m$$

Con la función objetivo (3.1) se minimiza la suma de las distancias promedio respecto a las distancias. Las restricciones (3.2) aseguran que todo punto de demanda quede asignado a una única instalación, y las restricciones (3.4) aseguran que si una instalación no está abierta, no puede servir a ningún punto de demanda. Es decir, estas últimas restricciones sirven para modelizar las condiciones  $x_j = 0 \Rightarrow y_{ij} = 0$  para todo  $i = 1, \dots, m$ .

Otra manera de formular las restricciones anteriores es

$$y_{ij} \leq x_j \quad (3.5)$$

para todo  $i = 1, \dots, m$  y  $j = 1, \dots, n$ .

Por lo tanto tenemos dos formulaciones diferentes para el problema de la p-mediana. La formulación con las restricciones (3.4) se denomina formulación fuerte, mientras que la formulación en la que sustituimos las restricciones (3.4) por las de (3.5) se denomina formulación débil.

Como hemos mencionado anteriormente, ambas son formulaciones válidas para el problema, son formulaciones equivalentes. La diferencia que encontramos entre ambas es que, pese a que el conjunto de soluciones enteras (en este caso binarias) es el mismo, el conjunto de soluciones continuas al implementar la formulación fuerte es mucho menor.

A priori podemos pensar que la formulación débil, al presentar menos restricciones, será más eficiente, sin embargo, a la hora de resolver la relajación lineal con la formulación fuerte observamos que el hecho de que presente más restricciones no supone una menor eficiencia. De hecho, a menudo, se obtiene la solución óptima entera directamente.

### Ejemplo del problema de la p-mediana

A continuación implementamos el modelo en Mosel-Xpress dando diferentes valores al número de instalaciones,  $p$  y  $n = 45$  puntos de demanda. Veamos como asignar a cada punto de demanda una instalación, para ello vamos a resolver el problema utilizando tres formulaciones diferentes, formulación débil, fuerte y de planos de corte.

Este es el código del programa.

```

1  !
2  !Resolucion optima del problema de la p-mediana
3  !
4  model "P-Mediana"
5  uses "mxxprs";
6  uses "mmsystem"
7  uses "mmive"
8  !a continuacion se declaran las variables
9  declarations
10     n:integer !numero de puntos leidos
11     archivo_datos="C:/Users/rocio/Desktop/mosel/s50_2.pln.txt"
12
13 end-declarations
14
15 !se lee el archivo con los datos para el problema
16 fopen(archivo_datos,F_INPUT)
17 readln(n,"\n")
18 declarations
19     p = 10
20     puntos = 1..n
21     !vectores de coordenadas de los puntos
22     c1:array(puntos)of integer          ! coordenada 1
23     c2:array(puntos)of integer          ! coordenada 2
24     !vector de demanda de cada punto
25     dem:array(puntos)of integer
26     !matriz de distancias entre puntos

```

```
27  dist:array(puntos,puntos)of integer
28
29  !variables de decision
30  x:array(puntos)of mpvar
31  y:array(puntos,puntos)of mpvar
32
33  !vector de restricciones lineales
34  res_vub:array(puntos,puntos)of lincstr
35
36  !tipo de formulacion que utilizamos (1 =debil, 2= fuerte
      vub explicitas, 3=planos de corte)
37  formulacion = 2
38  !definimos variables para controlar el algoritmo
39  final, nc:integer
40  tole = 0.0001
41  end-declarations
42
43  !leemos los datos del archivo
44  forall(j in puntos)readln(c1(j),c2(j),dem(j))
45  fclose(F_INPUT)
46
47  !A continuacion calculamos las distancias entre los puntos
48  !y las almacenamos en la matriz de distancias
49  forall(i in puntos,j in puntos)do
50      dist(i,j):=round(sqrt((c1(i)-c1(j))^2+(c2(i)-c2(j))^2))
51  end-do
52
53  !Configuramos el modelo
54
55  writeln("Problema P-Mediana, datos: ",archivo_datos)
56  writeln("n = ",n,", p = ",p,"\n")
57
58  dem_total:=sum(j in puntos)dem(j) !suma de las demandas
59  writeln("dem_total = ",dem_total)
60
61  forall(j in puntos)x(j) is_binary
62  forall(i in puntos,j in puntos)y(i,j) is_binary
63
64  !funcion objetivo
65  obj:=sum(i in puntos,j in puntos)dist(i,j)*dem(i)*y(i,j)
66
67  !con res_dem nos aseguramos de que cada punto sea asignado
68  !exactamente a una instalacion
69  forall(i in puntos)res_dem(i):=
70      sum(j in puntos)y(i,j)=1
71  !con res_p nos aseguramos de que haya exactamente p
      instalaciones
72  res_p:= sum(j in puntos)x(j)=p
```

```
73
74 t1:=gettime
75
76 !Resolvemos el problema en funcion de la formulacion escogida
77 !Con la formulacion debil se resuelve el problema usando una
    relajacion lineal
78 if(formulacion = 1)then
79     forall(j in puntos)res_debil(j):= sum(i in puntos)y(i,j)
        <=n*x(j)
80     minimize(XPRS_LIN,obj)
81     writeln("Formulacion debil, Relajacion Lineal z_LP = ",
        getobjval)
82
83     forall(j in puntos|x(j).sol>0.000001)writeln("x(",j,") =
        ",x(j).sol)
84     minimize(obj)
85     writeln("\nValor optimo entero = ",strfmt(getobjval,10,2)
        ", tiempo = ",strfmt(gettime-t1,5,2))
86 !Con la formulacion fuerte se resuelve el problema usando
    restricciones adicionales
87 elif(formulacion = 2)then
88     forall(i,j in puntos)res_vub(i,j):=y(i,j)<=x(j)
89     minimize(XPRS_LIN,obj)
90     writeln("Formulacion fuerte explicita, Relajacion Lineal
        z_LP = ",getobjval)
91     forall(j in puntos|x(j).sol>0.000001)writeln("x(",j,") =
        ",x(j).sol)
92     minimize(obj)
93     writeln("\nValor optimo entero = ",strfmt(getobjval,10,2)
        ", tiempo = ",strfmt(gettime-t1,5,2))
94 !Con la formulacion de planos de corte se itera el proceso
    introduciendo nuevos planos de corte hasta encontrar la
    solucion optima
95 elif(formulacion = 3)then
96     writeln("Formulacion fuerte con planos de corte:\n ")
97     forall(j in puntos)res_debil(j):= sum(i in puntos)y(i,j)<=
        n*x(j)
98     final:=0
99     nc:=0
100    iter:=0
101    while(final = 0)do
102        minimize(XPRS_LIN,obj)
103        iter+=1
104        writeln("iter ",iter," z = ",getobjval," nc = ",nc)
105        nci:=0
106        forall(i,j in puntos)do
107            if(y(i,j).sol > x(j).sol+tole)then
108                nc:=nc+1
```

```

109         nci+=1
110         res_vub(i,j):=y(i,j)<=x(j)
111     end-if
112 end-do
113     if(nci=0)then final:=1;end-if
114 end-do
115     writeln("Formulacion fuerte con planos de corte,
116           Relajacion Lineal z_LP = ",getobjval)
117     forall(j in puntos|x(j).sol>0.000001)writeln("x(",j,") = "
118           ,x(j).sol)
119     writeln("Numero de cortes: ",nc)
120     minimize(obj)
121     writeln("\nValor optimo entero = ",strfmt(getobjval,10,2),
122           ", tiempo = ",strfmt(gettime-t1,5,2))
123 end-if
124
125 !Para cada formulacion se minimiza la funcion objetivo
126
127 !===== Grafico de la solucion
128 =====
129
130 forall(i in puntos)do
131     dau:=999999;
132     forall(j in puntos|x(j).sol >=0.999)do
133         if(dist(i,j)<dau)then
134             dau:=dist(i,j)
135             jau:=j;
136         end-if
137     end-do
138     fi1(i):=jau;
139 end-do
140
141 IVEerase
142 id1:=IVEaddplot("Puntos_demanda",IVE_BLACK)
143 id2:=IVEaddplot("Asignacion",IVE_RED)
144
145 forall(i in puntos)do
146     IVEdrawpoint(id1,c1(i),c2(i))
147     IVEdrawline(id2,c1(i),c2(i),c1(fi1(i)),c2(fi1(i)))
148 end-do
149
150 end-model

```

Listing 3.1: Problema de la p-mediana en Mosel Xpress

Tras implementar el problema para cada valor de  $p$  obtenemos los resultados de cada formulación y un gráfico que relaciona cada punto con cada instalación. En la siguiente imagen se muestra la asignación de dichos puntos de demanda con  $p = 10$  instalaciones.

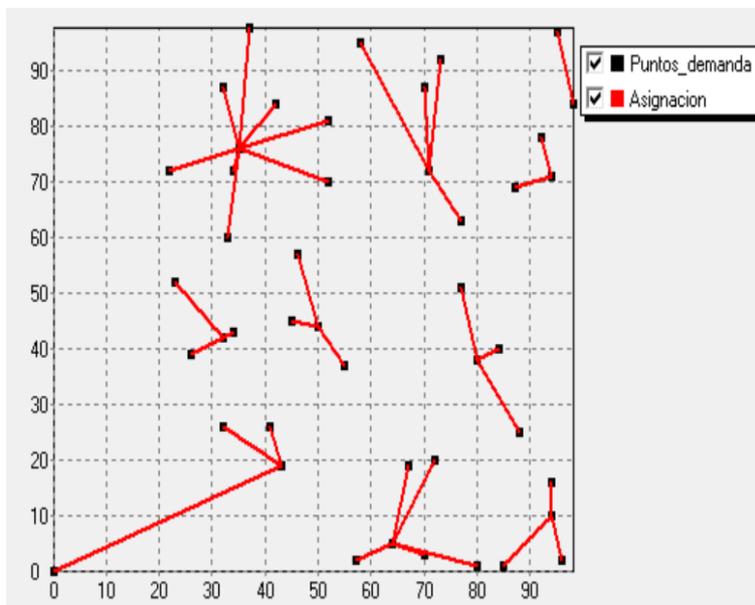


Figura 3.5: Asignación de instalaciones a cada punto,  $p = 10$

<b>p</b>	<b>relajación lineal</b>	<b>nº de restricciones</b>	<b>óptimo entero</b>	<b>valor</b>
10	$z_1 = 0$	91	No	15112.00
	$z_2 = 15112$	2071	Sí	
	$z_3 = 15112$	261	Sí	
15	$z_1 = 0$	91	No	8722.00
	$z_2 = 8722$	2071	Sí	
	$z_3 = 8722$	199	Sí	
25	$z_1 = 0$	91	No	2517.00
	$z_2 = 2517$	2071	Sí	
	$z_3 = 2517$	150	Sí	
40	$z_1 = 0$	91	No	47.00
	$z_2 = 47$	2071	Sí	
	$z_3 = 47$	136	Sí	

Tabla 3.1: Comparativa de resultados con formulaciones equivalentes

En esta tabla la fila  $z_1$  se corresponde con los resultados obtenidos al resolver la relajación lineal con la formulación débil, la fila  $z_2$  con la formulación fuerte y la fila  $z_3$  mediante planos de corte.

La primera columna indica el valor de  $p$  escogido en cada caso y para cada uno de estos valores tenemos los resultados de las tres formulaciones. La columna correspondiente a la relajación lineal nos da la solución obtenida mediante dicha relajación, a

continuación el número de restricciones de cada una de ellas y si llega o no al óptimo entero con la relajación lineal y el valor correspondiente.

Observando los resultados obtenidos en la tabla podemos pensar que la formulación débil es más sencilla ya que el número de restricciones es menor, y podemos pensar que es mejor que la fuerte para la resolución del problema, sin embargo también observamos que al elegir la formulación fuerte obtenemos directamente la solución óptima entera, mientras que con la formulación débil no es tan inmediata su obtención.

Por otra parte, pese a la eficiencia del algoritmo de planos de corte ya comentada, ya que con cada iteración se mejora significativamente la solución óptima, el tiempo de cómputo es mucho mayor, debido precisamente a las múltiples iteraciones y a la generación de nuevos cortes.

Actualmente el algoritmo que se utiliza para obtener el óptimo entero en un problema de programación lineal entera se llama “branch and cut”, pues es un método eficaz que combina el “branch and bound” (enumeración y ramificación) y los planos de corte.

La parte de enumeración y ramificación está detallada en otro trabajo de fin de grado [6], en esta memoria nos centramos en la parte de algoritmos de corte.



# Capítulo 4

## Cortes de Gomory

La noción de algoritmos de planos de corte para resolver problemas de optimización fue aparentemente utilizada por primera vez por Dantzig, Fulkerson y Jonhson(1954), pero la primera implementación de dichos algoritmos fue en 1958 por R.E. Gomory.

Las desigualdades propuestas por Dantzig buscan resolver un problema  $P_E$ ,  $z_{P_E} = \min\{cx : x \in S\}$  con  $S = \{x \in \mathbb{Z}^n : Ax \leq b, x \geq 0\}$ .

Para ello la idea es pasar a un problema relajado lineal  $P_R$ , encontrar la solución óptima, y generar una desigualdad válida que corte a la solución óptima que habíamos obtenido de  $P_R$ .

Estas desigualdades se generan a partir de combinaciones lineales de las restricciones originales y se añaden al problema para “cortar” las soluciones fraccionarias y acercarse más a la solución factible entera.

Se generan combinaciones lineales de la forma

$$\sum_{j=1}^m (\lambda_j a_j) x \leq \sum_{j=1}^m \lambda_j b_j$$

Estas desigualdades proporcionan una manera inicial de generar nuevas restricciones, pero no garantizan nada sobre la solución ya que no se llega a una relajación fuerte.

Más adelante, Gomory introduce los llamados cortes de Gomory, una técnica más efectiva, pues proporcionan desigualdades válidas que “cortan” las soluciones fraccionarias y se acercan con una relajación fuerte a la solución óptima.

Sin embargo, los algoritmos de Gomory tuvieron un éxito computacional limitado y por ello derivaron en diferentes casos especiales (descritos también por Gomory) que estudiaremos en esta sección. [8] [1]

## 4.1. Cortes Fundamentales

Dado el problema de programación lineal  $P_E$ ,  $z_{P_E} = \min\{cx : x \in S\}$  con  $S = \{x \in \mathbb{Z}^n : Ax \leq b\}$ .

Es bien conocido que en programación lineal se reescribe el problema añadiendo variables de holgura y en una iteración del método simplex [1] se obtienen las variables básicas, B, y no básicas, NB.

Por lo tanto, suponemos que tenemos una representación del problema dada por

$$\{x_i = \bar{x}_i - \sum_{j \in NB} \bar{a}_{ij}x_j, x_j \geq 0, i = 1, \dots, m\} \quad (4.1)$$

donde denotamos por  $a_{ij}$  el coeficiente de la variable no básica  $x_j$ . Esta descomposición de cada  $x_i$  es lo que llamamos descomposición canónica.

Multiplicando la igualdad anterior por  $h \neq 0$  obtenemos

$$hx_i + \sum_{j \in NB} h\bar{a}_{ij}x_j = h\bar{x}_i$$

y como  $x_i \geq 0$ , tomando partes enteras tenemos

$$\lfloor h \rfloor x_i + \sum_{j \in NB} \lfloor h\bar{a}_{ij} \rfloor x_j = \lfloor h\bar{x}_i \rfloor$$

Como  $x_i \in \mathbb{Z}^n$  el lado izquierdo de la desigualdad es entero, y por tanto no puede ser mayor que la parte entera del lado derecho, es decir

$$\lfloor h \rfloor x_i + \sum_{j \in NB} \lfloor h\bar{a}_{ij} \rfloor x_j = \lfloor h\bar{x}_i \rfloor \quad (4.2)$$

Si multiplicamos  $\lfloor h \rfloor(4.1)$  y le restamos (4.2) tenemos

$$\sum_{j \in NB} (\lfloor h \rfloor a_{ij} - \lfloor h\bar{a}_{ij} \rfloor)x_j \geq \lfloor h \rfloor \bar{x}_i - \lfloor h\bar{x}_i \rfloor$$

La desigualdad obtenida es lo que llamamos corte fundamental. [4]  
Utilizando esta desigualdad de varias maneras, se pueden desarrollar una serie de cortes y algoritmos de corte diferentes. En particular tomando  $h = 1$  obtenemos los cortes de Gomory que estudiaremos a continuación.

## 4.2. Problema entero puro

Dado un problema de programación lineal  $P_E$  puro  $\min\{cx : x \in S\}$ . Suponemos que obtenemos el problema relajado lineal  $P_R$  y que podemos resolverlo. Tenemos

una solución óptima  $\bar{x}$  de  $P_R$ .

Como hemos visto anteriormente podemos reescribir el problema de la siguiente forma

$$\{x_i = \bar{x}_i - \sum_{j \in NB} \bar{a}_{ij} x_j, x_j \geq 0\}$$

que llamamos descomposición canónica.

Si la solución  $\bar{x}$  de  $(P_R)$  es entera, entonces resuelve el problema  $(P_E)$  y hemos terminado.

Si no es entera estudiamos las partes fraccionarias.[8]

Definimos  $F(z) = z - \lfloor z \rfloor$  y reescribimos  $\bar{x}$ . Como  $\bar{x}$  no es entera, elegimos una fila  $r$  para la cual  $\bar{x}_r$  no es entero, es decir  $F(\bar{x}_r) > 0$  y obtenemos  $x_r = \bar{x}_r - \sum_{j \in NB} \bar{a}_{ij} x_j$  (por construcción) y sustituyendo por  $F(\bar{x}_r)$  y nos queda

$$x_r = F(\bar{x}_r) + \lfloor \bar{x}_r \rfloor - \sum_{j \in NB} (F(\bar{a}_{ij}) - \lfloor a_{ij} \rfloor) x_j$$

y reagrupando

$$\sum_{j \in NB} F(\bar{a}_{ij}) x_j - F(\bar{x}_r) = \lfloor \bar{x}_r \rfloor - \sum_{j \in NB} \lfloor a_{ij} \rfloor x_j - x_r$$

Basta probar que ambos lados de la igualdad son positivos para afirmar la validez de los cortes fraccionarios de Gomory que son

$$\sum_{j \in NB} F(\bar{a}_{ij}) x_j \geq F(\bar{x}_r)$$

**Teorema 4.1.** *Validez de los cortes fraccionarios de Gomory.*[8]

*Sean  $P_E$  y  $P_R$  los problemas definidos anteriormente, la desigualdad*

$$\sum_{j \in NB} F(\bar{a}_{ij}) x_j \geq F(\bar{x}_r)$$

*es un corte válido para  $(P_E)$ , es decir, es válida para todo entero  $x \in S$  y corta a la solución actual.*

*Demostración.* El conjunto de soluciones factibles de  $P_E$ ,  $S$  satisface que  $x \geq 0$ , y además se tiene que para cualquier  $z \in \mathbb{R}^n$ ,  $1 > F(z) \geq 0$ . Por tanto  $\sum_{j \in NB} F(\bar{a}_{ij}) x_j - F(\bar{x}_r) > -1$  para todo  $x \in S$ .

Además cada  $x \in S$  tiene componentes enteras  $x_r$  y  $x_j$  con  $j \in \mathbb{N}$ , por lo que

$\lfloor \bar{x}_r \rfloor - \sum_{j \in NB} \lfloor a_{ij} \rfloor x_j - x_r$  es entero para cualquier  $x \in S$ .

Podemos concluir que  $\sum_{j \in NB} F(\bar{a}_{ij})x_j - F(\bar{x}_r) > -1 \geq 0$  para todo  $x \in S$ , es decir

$$\sum_{j \in NB} F(\bar{a}_{ij})x_j \geq F(\bar{x}_r)$$

es válida para  $(P_E)$ .

□

A continuación describimos el algoritmo de planos de cortes fraccionarios de Gomory.[8]

### Algoritmo de planos de cortes fraccionarios de Gomory

*Notación.* Sea  $z_P$  el valor óptimo de un problema  $P$ :

- $z_P = +\infty$  si el problema  $P$  es un problema de minimizar no factible.
- $z_P = -\infty$  si el problema  $P$  es un problema de minimizar sin acotar.

Sea  $P_E$  el problema ya definido, y  $P_R$  el problema relajado lineal con valor óptimo  $z(P_R)$  acotado.

**Inicialización:** Tenemos las variables  $x_j$  con  $j \in \{1, 2, \dots, n\}$  del problema  $P_E$ , y pasamos al problema  $(P_R)^1$ .

**Pasamos a la iteración general, t:**

- **Etapa 1:** Resolución del problema  $(P_R)^t$

Si el valor óptimo  $z(P_R)^t = +\infty$  entonces hemos terminado y tanto el problema  $(P_R)^t$  como  $(P_E)$  son no factibles.

Si  $z((P_R)^t) = x^t$  es solución óptima de  $(P_R)^t$  pasamos a la siguiente etapa.

- **Etapa 2:** Test de optimalidad

Si  $x^t$  es entera entonces es también solución óptima de  $(P_E)$  y hemos terminado.

Si  $x^t$  no es entera pasamos a la siguiente etapa.

- **Etapa 3:** Buscamos el corte

Como  $x^t$  no es entera, existe un  $r, 1 \leq r \leq n$  tal que  $x_r^t$  no es entero, de

manera que podemos reescribir la fila  $r$  de las ecuaciones de la siguiente forma:

$$x_r = \bar{x}_r^t - \sum_{j \in (NB)^t} \bar{a}_{ij}^t x_j$$

Obtenemos un nuevo problema  $(P_R)^{t+1}$  añadiendo la desigualdad

$$\sum_{j \in (NB)^t} F(\bar{a}_{ij}^t) x_j \leq F(\bar{x}_r^t)$$

y añadiendo variables adicionales  $x_{n+t+1} \geq 0$  obtenemos la igualdad

$$\sum_{j \in (NB)^t} F(\bar{a}_{ij}^t) x_j - x_{n+t+1} = F(\bar{x}_r^t)$$

Finalmente reemplazamos  $t$  por  $t + 1$  y volvemos a la etapa 1.

**Ejemplo 4.2.** Consideramos el problema  $P_E$

$$\text{mín}\{7x_1 + x_2$$

sujeto a

$$x_1 - 3x_2 \geq -1, 3x_1 + 2x_2 \geq 4, x_1, x_2 > 0\}$$

Añadimos variables adicionales y obtenemos el problema relajado  $(P_R)^1$

$$\text{mín } 7x_1 + x_2$$

sujeto a

$$x_1 - 3x_2 - x_3 = -1 \tag{4.3}$$

$$3x_1 + 2x_2 - x_4 = 4 \tag{4.4}$$

$$x_1, x_2, x_3, x_4 > 0$$

Comenzamos la primera etapa, es decir, resolvemos el problema relajado  $(P_R)^1$  y llegamos a la solución  $x^1 = (\frac{10}{11}, \frac{7}{11}, 0, 0)$ . Comprobamos que  $x^1 \notin \mathbb{Z}^4$  y pasamos a buscar el corte.

Recordamos la fórmula  $x_i = \bar{x}_i - \sum_{j \in NB} \bar{a}_{ij} x_j$  y reescribimos

$$\begin{aligned} x_0 &= -7 - (1x_3 + 2x_4) \\ x_1 &= \frac{10}{11} - \left(-\frac{2}{11}x_3 - \frac{3}{11}x_4\right) \\ x_2 &= \frac{7}{11} - \left(\frac{3}{11}x_3 - \frac{1}{11}x_4\right) \end{aligned}$$

Tanto  $x_1$  como  $x_2$  son válidas para generar el plano de corte, elegimos  $x_1$  y el corte de Gomory correspondiente es

$$F\left(-\frac{2}{11}\right)x_3 + F\left(-\frac{3}{11}\right)x_4 \geq F\left(\frac{10}{11}\right)$$

y añadiendo una nueva variable  $x_5$  tenemos

$$\frac{9}{11}x_3 + \frac{8}{11}x_4 - x_5 = \frac{10}{11} \quad (4.5)$$

	$x_1$	$x_2$	$x_3$	$x_4$	b	
$x_0$	1	0	0	1	2	7
$x_1$	0	1	0	$\frac{9}{11}$	$\frac{8}{11}$	$\frac{10}{11}$
$x_2$	0	0	1	$-\frac{8}{11}$	$\frac{10}{11}$	$\frac{7}{11}$

Tabla 4.1: Tabla correspondiente al método simplex

Además, utilizando (4.3) y (4.4) y sustituyendo  $x_3$  y  $x_4$  obtenemos

$$\frac{9}{11}(x_1 - 3x_2 + 1) + \frac{8}{11}(3x_1 + 2x_2 - 4) \geq \frac{10}{11}$$

De manera que tenemos un nuevo problema  $(P_R)^2$  y comenzamos de nuevo el algoritmo.

Resolvemos  $(P_R)^2$  y obtenemos  $x^2 = (\frac{10}{11}, \frac{1}{3}, \frac{10}{9}, 0, 0) \notin \mathbf{Z}^5$ . Reiterando el proceso anterior y, añadiendo una nueva variable  $x_6$  encontramos un nuevo corte

$$\frac{1}{9}x_4 + \frac{2}{9}x_5 - x_6 = \frac{8}{9} \quad (4.6)$$

De nuevo utilizando (4.3), (4.4) y (4.5) y sustituyendo  $x_3, x_4$  y  $x_5$  obtenemos

$$x_1 \geq 2$$

Tenemos un nuevo problema relajado  $(P_R)^3$  con  $x^3 = (2, 0, 3, 2, 3, 0) \in \mathbf{Z}^6$ , solución óptima. Hemos llegado a una solución óptima para  $P_E$  y por lo tanto hemos terminado.

Es importante remarcar que a medida que avanza el algoritmo se puede llegar a acumular un número bastante grande de planos de corte, lo que conlleva una gran carga de cálculo y almacenamiento.

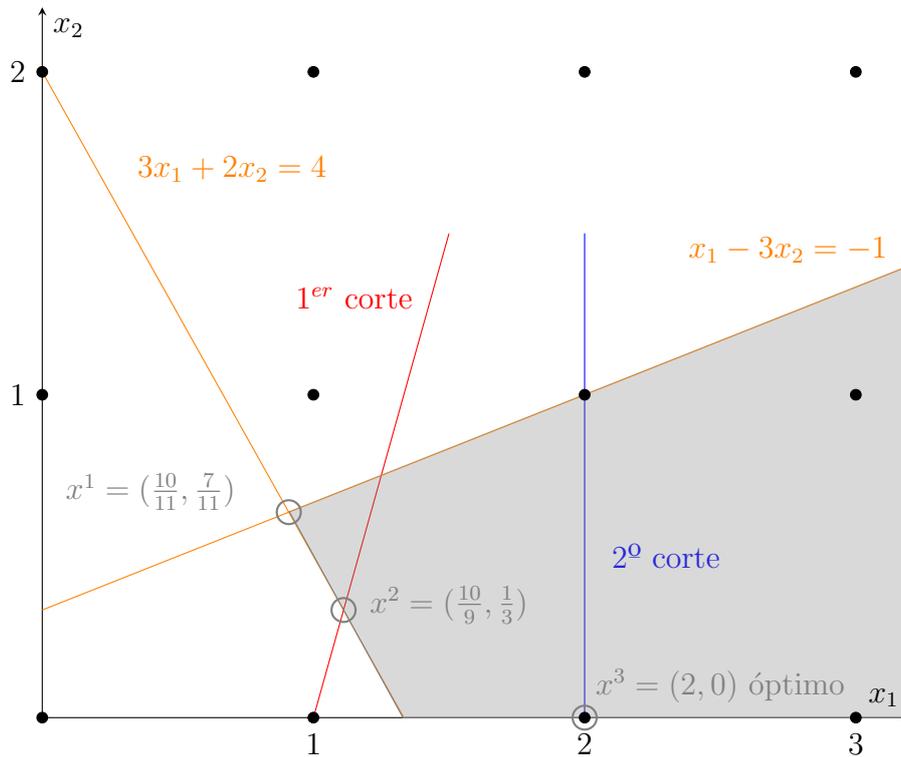


Figura 4.1: Representación de los cortes de Gomory (ejemplo 4.2)

La etapa 3 del algoritmo incluye una técnica para aliviar esta acumulación de planos de corte.

Cuando un nuevo plano de corte no es vinculante podemos desecharlo sin afectar a la convergencia. Es posible que una desigualdad eliminada sea regenerada más tarde, pero este costo no supone un gran problema frente a la acumulación de cortes.

Por tanto, el logro clave de Gomory fue demostrar la convergencia finita de dicho algoritmo de planos de corte.

### 4.3. Convergencia finita del algoritmo

Vamos a ver que el algoritmo de planos de corte fraccionarios de Gomory tiene convergencia finita a la que hemos llamado solución óptima entera.[4][8]

La demostración depende del concepto de orden lexicográfico. Se dice que el vector  $z_1$  es lexicográficamente mayor que  $z_2$ , denotado  $z_1 >_L z_2$ , si el primer componente no nulo de  $z_1 - z_2$  es positivo.

**Teorema 4.3.** *Convergencia finita del algoritmo de cortes fraccionarios.*

Sea  $P_E$  el problema de programación entera definido anteriormente, con un valor óptimo para el problema relajado  $P_R$ ,  $z(P_R) > -\infty$ , y supongamos que el algoritmo descrito se implementa a través de un procedimiento de simplex pasando por las

soluciones básicas  $x^0 := w^0, w^1, \dots, w^{t_0}$ ,  $x^1 := w^{t_0+1}, w^{t_0+2}, \dots, w^{t_1}$ , etc. Entonces, si

i)  $(w_0^t, \dots, w_n^t) >_L (w_0^{t+1}, \dots, w_n^{t+1})$  para todo  $t$ , y

ii) se elige el menor  $r$  admisible en cada ejecución de la etapa 3 del algoritmo (es decir,  $r = \min\{j = 0, 1, \dots, n : F(x_j^k) > 0\}$ )

el algoritmo termina después de un número finito de iteraciones del método simplex,  $t_p$ , con  $(x_1^p, \dots, x_n^p)$  solución óptima para  $P_E$  o por el contrario concluyendo que no existe una solución factible para el problema  $P_E$ .

*Demostración.* Si el algoritmo termina en la primera etapa concluyendo que el problema  $P_E$  no es factible, o en la segunda etapa con una solución óptima entera para  $P_E$ , es claro que el algoritmo termina en un número finito de iteraciones.

Las restricciones añadidas en la tercera etapa,  $x_{n+t+1} \geq 0$  y  $\sum_{j \in (NB)^t} F(a_{ij}^t) x_j - x_{n+t+1} = F(\bar{x}_r^t)$  son válidas, por lo que  $(P_R)^t$  sigue siendo una relajación del problema  $P_E$ . Queremos demostrar que el procedimiento no puede continuar indefinidamente.

Para ello razonamos con las componentes  $0, 1, 2, \dots, n$  de la solución en orden. Consideramos la secuencia  $\{w_0^t\}$ . Esta secuencia está acotada inferiormente y es monótona decreciente debido al orden lexicográfico, lo que significa que alcanza un límite que denotamos por  $w_0^\infty$ , y en un número finito de iteraciones del método simplex, la secuencia se acercará arbitrariamente a  $w_0^\infty$ .

Elegimos una iteración  $t_q$ , con solución óptima para  $(P_R)^q$ ,  $x^q = w^{t_q}$  y tal que  $w_0^{t_q} - \lfloor w_0^\infty \rfloor < 1$ . Si  $w_0^{t_q} = \lfloor w_0^\infty \rfloor$  pasamos a la siguiente componente. Si no, eso quiere decir que  $F(w_0^{t_q}) > 0$  y (ii) nos lleva a generar una nueva restricción de la forma  $\sum_{j \in (NB)^t} F(a_{ij}^t) x_j - x_{n+t+1} = F(\bar{x}_r^t)$  para  $r = 0$ .

La nueva restricción se introduce en la fila correspondiente al problema, y en la siguiente iteración  $t_q + 1$ , el pivote se realizará en la nueva fila de corte, que es la única fila no factible para la solución actual, y en la columna de alguna variable no básica  $x_h$  con  $F(a_{0h}^q) > 0$ , donde  $a_{0h}^q$  es un coeficiente fraccionario asociado a la nueva restricción.

De esta manera la componente 0 de la solución  $w_0^{t_q}$  queda actualizado de la forma siguiente:

$$w_0^{t_q+1} \leftarrow w_0^{t_q} - a_{0h}^q \frac{F(w_0^{t_q})}{F(a_{0h}^q)} \quad (4.7)$$

$F(w_0^{t_q})$  y  $F(a_{0h}^q)$  son ambos positivos y  $a_{0h}^q$  es fraccionario, luego es distinto de cero.

Si  $a_{0h}^q < 0$ , en la desigualdad anterior tendríamos  $w_0^{t_q+1} > w_0^{t_q}$  que contradice el

orden lexicográfico, por lo tanto  $a_{0h}^q > 0$ . Pero entonces  $a_{0h}^q \geq F(a_{0h}^q)$  y junto con (4.7) implican que

$$w_0^{t_q} - F(w_0^{t_q}) \geq w_0^{t_q+1} \geq \lfloor w_0^\infty \rfloor.$$

Dado que elegimos  $t_q$  con  $w_0^{t_q} - F(w_0^{t_q}) = \lfloor w_0^\infty \rfloor$ , podemos concluir que  $w_0^{t_q+1} = \lfloor w_0^\infty \rfloor$ . El límite (entero) de la serie no creciente  $\{w_0^t\}$  se ha alcanzado después de un número finito de iteraciones del método simplex.

Una vez que la sucesión  $\{w_0^t\}$  alcanza su límite, el orden lexicográfico asegura que ese valor que se alcanza se mantendrá en todas las futuras iteraciones. Podemos pasar a analizar la siguiente componente de la solución  $w_1^t$ , y la secuencia  $w_1^{t_q+1}, w_1^{t_q+2}, \dots$ , que como antes es monótona decreciente. Además debe estar acotada inferiormente por 0, pues si algún  $w_1^t$  con  $t > t_q$  es negativo, la secuencia no volverá a una solución factible del problema relajado. Por lo tanto  $\{w_1^t\}$  es monótona y está acotada inferiormente, y también tiene límite. Por el mismo razonamiento de antes, este límite (entero) se alcanzará después de un número finito de iteraciones.

Continuando con el mismo argumento para  $w_2, w_3, \dots, w_n$ , todas las variables originales del problema necesariamente alcanzarán sus valores límite después de un número finito de iteraciones. Solo necesitamos completar el problema relajado  $(P_R)^k$  durante el cual se alcanza el último límite para terminar el algoritmo en un número finito de pasos.

□

Para probar la convergencia hemos asumido que el algoritmo de planos de corte de Gomory, puede realizarse con las propiedades (i) y (ii).

Aunque en la práctica no siempre es la mejor opción se puede implementar (ii) al realizar el algoritmo, y mediante una versión adecuada del método simplex también se puede implementar (i). [1]

## 4.4. Problema entero mixto

Aunque hemos probado la validez de los cortes fraccionarios de Gomory para el caso entero puro, Gomory demostró que se podía extender el resultado al caso de problemas enteros mixtos.

Dado un problema  $P_M$  mixto  $\min\{cx : x \in S\}$ . Al igual que en el caso anterior los cortes fraccionarios de Gomory se representan de la siguiente forma

$$x_r = \bar{x}_r - \sum_{j \in N_I} \bar{a}_{ij} x_j - \sum_{j \in N_C} \bar{a}_{ij} x_j$$

con  $x_r$  la variable correspondiente con valor fraccionario del problema  $(P_R)$   $\bar{x}_r$ , y el conjunto de variables no básicas  $NB$  lo dividimos en dos;  $N_I$ , el conjunto de variables no básicas enteras y  $N_C$  de variables no básicas continuas.

Reescribiendo la igualdad anterior en términos de sus partes fraccionarias y reagrupando obtenemos

$$\sum_{j \in N_I} F(\bar{a}_{ij})x_j - \sum_{j \in N_C} \bar{a}_{ij}x_j - F(\bar{x}_r) = \lfloor \bar{x}_r \rfloor - \sum_{j \in N_I} \lfloor \bar{a}_{ij} \rfloor x_j - x_r \quad (4.8)$$

Observando que el lado izquierdo de la igualdad (4.8) tiene que ser un valor entero  $\leq -1$  o  $\geq 0$  podemos dar una demostración de la validez de los cortes fraccionarios de Gomory en el caso mixto, que son de la forma

$$\sum_{j \in N_I} F(\bar{a}_{ij})x_j + \sum_{j \in N_C^+} \bar{a}_{ij}x_j + \frac{F(\bar{x}_r)}{\frac{1}{2}F(\bar{x}_r)} \sum_{j \in N_C^-} (-\bar{a}_{ij})x_j \geq F(\bar{x}_r)$$

**Teorema 4.4.** *Validez de los cortes fraccionarios de Gomory en el caso mixto:*

Sean  $P_M$  y  $P_R$  los problemas definidos anteriormente, la desigualdad

$$\sum_{j \in N_I} F(\bar{a}_{ij})x_j + \sum_{j \in N_C^+} \bar{a}_{ij}x_j + \frac{F(\bar{x}_r)}{\frac{1}{2}F(\bar{x}_r)} \sum_{j \in N_C^-} (-\bar{a}_{ij})x_j \geq F(\bar{x}_r)$$

es válida para  $(P_E)$ , es decir, todo entero  $x \in S$  satisface dicha ecuación.

*Demostración.* Todas las constantes y variables del lado derecho de (4.8) son enteras para  $x$  factible en  $P_M$ , por lo tanto el lado izquierdo también es entero. En particular es  $\geq 0$  o  $\leq -1$ , es decir, es

$$\sum_{j \in N_I} F(\bar{a}_{rj})x_j + \sum_{j \in N_C} \bar{a}_{rj}x_j - F(\bar{x}_r) \geq 0 \quad (4.9)$$

o

$$\sum_{j \in N_I} (-F(\bar{a}_{rj}))x_j + \sum_{j \in N_C} (-\bar{a}_{rj})x_j + F(\bar{x}_r) \geq 1 \quad (4.10)$$

Para  $x \geq 0$  los términos con coeficientes  $\leq 0$  en el lado izquierdo de (4.9) y (4.10) no pueden contribuir para satisfacer las desigualdades. Por lo tanto, reescribimos las dos desigualdades y tenemos

$$\sum_{j \in N_I} F(\bar{a}_{rj})x_j + \sum_{j \in N_C^+} \bar{a}_{rj}x_j \geq F(\bar{x}_r) \quad (4.11)$$

o

$$\sum_{j \in N_C^-} (-\bar{a}_{rj})x_j \geq 1 - F(\bar{x}_r) \quad (4.12)$$

Teniendo en cuenta que  $0 < F(\bar{x}_r) < 1$  ya que  $\bar{x}$  es fraccionario, las desigualdades anteriores son equivalentes a

$$\frac{F(\bar{x}_r)}{1 - F(\bar{x}_r)} \sum_{j \in N_C^-} (-a_{rj}^-) x_j \geq F(\bar{x}_r) \quad (4.13)$$

Concluimos que tanto el lado izquierdo de (4.11) como el de (4.13) deben ser al menos iguales a  $F(\bar{x}_r)$  para alguna solución del problema  $P_M$ . Pero entonces, efectivamente la suma de estos dos lados es  $F(\bar{x}_r)$  y el lado izquierdo de (4.8) es exactamente esa suma.

□

En este caso también encontramos el problema de la acumulación de planos de corte. Si el valor de  $z(P_M)$  se puede asumir restringido a enteros, entonces de manera análoga al teorema 4.3 se prueba la convergencia finita de los algoritmos mixtos de Gomory. Sin embargo, dado que  $z(P_M)$  puede incluir variables continuas con valores fraccionarios, es posible que no siempre podamos asumir que sean valores enteros. En este caso, el algoritmo puede no converger de manera finita.

## 4.5. Dificultades computacionales

En secciones anteriores hemos visto que el Algoritmo de planos de cortes fraccionarios de Gomory converge en un número finito de iteraciones. Desafortunadamente, la limitada experiencia computacional al realizar las implementaciones de dicho algoritmo en los ordenadores ha sido insatisfactoria.

La dificultad principal no viene dada por gran número de iteraciones, si no por los errores numéricos en aritmética computacional.

Los elementos de cada corte que daban estos problemas eran las partes fraccionarias,  $F(a_{rj}^-)$  de los coeficientes  $a_{rj}^-$ .

Gomory, en 1963 y otros matemáticos propusieron un enfoque para evitar tales errores de redondeo, sin embargo estos nuevos planos de corte son más débiles que los cortes fraccionarios.

Teóricamente esta demostrado que el algoritmo de planos de corte de Gomory converge de manera finita a la solución óptima de un problema de programación lineal entera. Sin embargo, en la práctica Gomory se dio cuenta de que dicho algoritmo tardaba mucho en concluir y muchas veces ni siquiera terminaba, además en algunas iteraciones se daban cortes repetidos o que no aportaban nada nuevo a la solución del problema.

Es por esto que los cortes fraccionarios de Gomory se dejaron a parte a la hora de buscar soluciones óptimas para estos problemas.

Actualmente los cortes de Gomory si que se utilizan, pero no con el propósito de encontrar una solución óptima, como era la idea inicial. Se emplean para encontrar relajaciones lineales fuertes que permitan describir lo mejor posible la envolvente convexa de la región factible de un problema.

# Lista de figuras y tablas

## Figuras

- Figura 1.1: *Poliedro acotado (ejemplo 1.6).*
- Figura 1.2: *Cara de dimensión uno del poliedro (ejemplo 1.12).*
- Figura 1.3: *Envolverte convexa de  $S$  (ejemplo 1.24).*
- Figura 2.1: *Problema de flujo de redes.*
- Figura 2.2: *Primera fase del método simplex y base factible.*
- Figura 3.1: *Mejor formulación de  $S$ .*
- Figura 3.2: *Desigualdades válidas fuertes.*
- Figura 3.3: *Representación de corte.*
- Figura 3.4: *Desigualdades de C-G.*
- Figura 3.5: *Asignación de instalaciones a cada punto con  $p = 10$*
- Figura 4.1: *Representación de los cortes de Gomory (ejemplo 4.2).*

## Tablas

- Tabla 2.1: *Tabla método simplex (ejemplo 2.16).*
- Tabla 3.1: *Comparativa de resultados con formulaciones equivalentes*
- Tabla 4.1: *Tabla método simplex (ejemplo 4.2).*



# Bibliografía

- [1] Mokhtar S Bazaraa, John J Jarvis, and Hanif D Sherali. *Linear programming and network flows*. John Wiley & Sons, 2011.
- [2] Harlan Crowder, Ellis L Johnson, and Manfred Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31(5):803–834, 1983.
- [3] Mark Daskin. *Network and discrete location: models, algorithms and applications*. 1997.
- [4] Bernard Max Garfinkel and George L Nemhauser. *Integer programming*. 1972.
- [5] Michael Jünger, Thomas M Liebling, Denis Naddef, George L Nemhauser, William R Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A Wolsey. *50 Years of integer programming 1958-2008: From the early years to the state-of-the-art*. Springer Science & Business Media, 2009.
- [6] Carlos Noriega González et al. *El algoritmo branch and cut en programación lineal entera*. 2023.
- [7] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.
- [8] R Gary Parker and Ronald L Rardin. *Discrete optimization*. Elsevier, 2014.
- [9] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
- [10] Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.
- [11] Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 2014.