Bases de Gröbner y prescripción de medicamentos

Erika Ruiz Frauca

Trabajo de fin de grado de Matemáticas
Universidad de Zaragoza

Febrero de 2025

Abstract

This work shows the usefulness of Grobner's basis in the prescription of drugs. It seems that these concepts are far apart, but this is not the case. We will see how these basis are used in integer linear programming and thus solve minimisation problems. In addition, we introduce Graver's basis, which are much more efficient than Gröbner's and moreover they can be applied in nonlinear integer programming. With all this, we will solve real-life examples of optimising the packaging of drugs prescribed to patients.

The first chapter is an introductory chapter where fundamental definitions and results of polynomials, ideals and monomial orders are discussed. The Division Algorithm and Dickson's Lemma are presented, and the terminology that will be used throughout the work is established.

In chapter 2, we study Gröbner basis and their application in integer linear programming. We show how to transform a linear system into a problem of polynomial equations and to be able to solve it by means of Gröbner basis. Numerous results are presented with their respective proofs and are accompanied by examples that help to understand them. This chapter is very important for the development of chapter 4 since the algorithms and results seen here will be used.

The third chapter introduces Graver's basis which are a tool for solving linear programming problems with negative coefficients as well as non-linear programming problems. These basis are related to Gröbner basis but are more effective for solving linear programming problems.

Finally, we have a chapter on the application of all the theory seen. It shows the resolution of real problems such as the optimisation of the packaging of medicines prescribed to a patient. For this type of problems, a minimisation problem is posed which will be transformed into polynomial equations and solved by means of Gröbner basis. In addition, there is an example that uses Graver's basis to solve one of those problems quickly and easily. Thanks to these two tools, the optimal solutions are calculated, i.e. the containers that a doctor should prescribe to a patient in order to waste as little medicine as possible.

In conclusion, Gröbner's and Graver's basis allow to solve algebraic systems and to optimise problems such as drug prescription, facilitating efficient solutions.

Índice general

Capítulo 1

Preliminares

Este capítulo proporciona los conceptos básicos necesarios para comprender la teoría de las bases de Gröbner y de Graver. Se abordan nociones fundamentales de polinomios, ideales y órdenes monomiales. Las demostraciones de los resultados no están incluidas porque son vistas en una asignatura del grado pero sí se incluyen referencias.

1.1. Nociones básicas

Definición 1.1. Un *monomio* en varias variables $x_1, \ldots, x_n$ es un producto de la forma $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ donde los exponentes $\alpha = (\alpha_1, \ldots, \alpha_n)$ son enteros no negativos. El *grado total* de un monomio viene dado por la suma $|\alpha| = \sum_{i=1}^n \alpha_i$.

Definición 1.2. Se denomina *soporte de un monomio* $w \in \mathbb{K}[x_1, \ldots, x_n]$ al conjunto

$$supp(w) = \{x_i : x_i \text{ divide a } w\}.$$

Es decir, el soporte de un monomio w está formado por las variables x^i que aparecen en w con exponente distinto de cero.

Definición 1.3. Un *polinomio* f en varias variables $x_1, \ldots, x_n$ con coeficientes en un cuerpo $\mathbb{K}$ es una combinación lineal finita de monomios. Se denota $f = \sum_\alpha a_\alpha x^\alpha$ con $a_\alpha \in \mathbb{K}$ y para un número finito de n-tuplas $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}^n_{\geq 0}$. Se denomina grado total de f al máximo de los grados totales de los monomios, $|\alpha|$, con coeficiente $a_\alpha \neq 0$.

El conjunto de polinomios con variables $x_1, \ldots, x_n$ y con coeficientes en $\mathbb{K}$ cuerpo, forman el anillo de polinomios denotado $\mathbb{K}[x_1, \ldots, x_n]$.

Definición 1.4. Se dice que un subconjunto $I \subset \mathbb{K}[x_1, \cdots, x_n]$ es un *ideal* si contiene al 0, es cerrado bajo suma interna y bajo multiplicación por elementos de $\mathbb{K}[x_1, \cdots, x_n]$.

Definición 1.5. Sean $f_1, \cdots, f_s$ polinomios en $\mathbb{K}[x_1, \cdots, x_n]$. El conjunto $\langle f_1, \cdots, f_s \rangle = \{\sum_{i=1}^s h_i f_i : h_1, \cdots, h_s \in \mathbb{K}[x_1, \cdots, x_n]\}$ es el *ideal generado* por $f_1, \cdots, f_s$.

Teorema 1.6 (Teorema de la base de Hilbert). *Todo ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$ tiene un conjunto generador finito. Es decir, $I = \langle g_1, \ldots, g_s \rangle$ para ciertos polinomios $g_1, \ldots, g_s$.*

Demostración. [3, p.77]

Lema 1.7. *Sea $a_1, \ldots, a_n, b_1, \ldots, b_n$ elementos de un anillo conmutativo R. Entonces, los elementos de la forma $a_1^{k_1} \cdots a_n^{k_n} - b_1^{k_1} \cdots b_n^{k_n}$ están en el ideal $\langle a_1 - b_1, \ldots, a_n - b_n \rangle$.*

Demostración. [2, p.80]

1.2. Orden de los monomios e ideales monomiales

En esta sección vamos a estudiar formas de ordenar los monomios, esto nos servirá para determinar cuando un polinomio está en un ideal. Las demostraciones de los resultados de esta sección los podemos encontrar en la referencia [3, p.64 - p.73].

Definición 1.8. Un *orden monomial* en $\mathbb{K}[x_1, \cdots, x_n]$ es una relación $>$ en $\mathbb{Z}^n_{\geq 0}$ que satisface

i) $>$ es un orden total en $\mathbb{Z}^n_{\geq 0}$.

ii) Si $\alpha > \beta$ y $\gamma \in \mathbb{Z}^n_{\geq 0}$, se tiene que $\alpha + \gamma > \beta + \gamma$.

iii) $>$ es un buen orden en $\mathbb{Z}^n_{\geq 0}$. Es decir, todo subconjunto no vacío $A \subseteq \mathbb{Z}^n_{\geq 0}$ tiene un elemento que es menor que todos los demás respecto del orden $>$.

Dados monomios x^α y x^β con $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$ escribiremos $x^\alpha > x^\beta$ si $\alpha > \beta$.
Existen muchos órdenes monomiales, pero vamos a estudiar solamente dos de ellos.

Definición 1.9. Sean $\alpha = (\alpha_1, \ldots, \alpha_n), \beta = (\beta_1, \ldots, \beta_n) \in \mathbb{Z}^n_{\geq 0}$. Decimos que $\alpha >_{lex} \beta$ si en el vector $\alpha - \beta$ la primera componente no nula empezando por la izquierda es positiva. A $>_{lex}$ se le llama *orden lexicográfico.*

Definición 1.10. Sean $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$, decimos que $\alpha >_{grlex} \beta$ si o bien $|\alpha| > |\beta|$, o bien $|\alpha| = |\beta|$ y $\alpha >_{lex} \beta$. A $>_{grlex}$ se le llama *orden lexicográfico graduado.*

A partir de ahora usaremos la siguiente terminología.

Definición 1.11. Sea $f = \sum_\alpha a_\alpha x^\alpha$ un polinomio no nulo en $\mathbb{K}[x_1, \ldots, x_n]$ y sea $>$ un orden monomial.

i) El *multigrado (multidegree)* de f es $multideg_>(f) = \max_>\{\alpha \in \mathbb{Z}^n_{\geq 0} : a_\alpha \neq 0\}$.

ii) El *coeficiente director (leading coefficient)* de f es $LC_>(f) = a_{multideg_>(f)} \in \mathbb{K}$.

iii) El *monomio director (leading monomial)* de f es $LM_>(f) = x^{multideg_>(f)}$.

iv) El *término director (leading term)* de f es $LT_>(f) = LC_>(f) \cdot LM_>(f)$.

Para no sobrecargar la notación no pondremos el subíndice $>$ si está claro por el contexto.

Teorema 1.12 (Algoritmo de la división en $\mathbb{K}[x_1, \ldots, x_n]$). *Sea $>$ un orden monomial en $\mathbb{Z}^n_{\geq 0}$ y sea $F = (f_1, \ldots, f_s)$ una s-tupla ordenada de polinomios en $\mathbb{K}[x_1, \ldots, x_n]$. Entonces todo $f \in \mathbb{K}[x_1, \ldots, x_n]$ puede escribirse como*

$$f = a_1 f_1 + \cdots + a_s f_s + r$$

donde $a_1, \ldots, a_s, r \in \mathbb{K}[x_1, \ldots, x_n]$ y, o bien $r = 0$, o bien r es una combinación lineal con coeficientes en $\mathbb{K}$ de monomios que no son divisibles por ningún $LT(f_1), \ldots, LT(f_s)$.
Además, si $a_i f_i \neq 0$ con $i \in \{1, \ldots, s\}$, se tiene que $multideg(f) \geq multideg(a_i f_i)$.

Notemos que importa el orden en el que se toman los polinomios f_i al realizar la división de f, ya que si se cambia el orden se podrían obtener distintos a_i. Más adelante, veremos que las bases de Gröbner conseguirán la unicidad de estos coeficientes.

Definición 1.13. Un ideal I de $\mathbb{K}[x_1, \ldots, x_n]$ es un *ideal monomial* si se puede generar por monomios, es decir, si existe un subconjunto $A \subset \mathbb{Z}^n_{\geq 0}$ tal que $I = \langle x^\alpha : \alpha \in A \rangle$. I está formado por todos los polinomios que son sumas finitas de la forma $\sum_{\alpha \in A} h_\alpha x^\alpha$ donde $h_\alpha \in \mathbb{K}[x_1, \ldots, x_n]$.

Por lo tanto vemos que un ideal monomial I está generado por un conjunto de monomios y que sus elementos son polinomios.

Proposición 1.14. *Sea $I = \langle x^\alpha : \alpha \in A\rangle$ un ideal monomial con $A \subset \mathbb{Z}^n_{\geq 0}$. Entonces, para $\beta \in \mathbb{Z}^n_{\geq 0}$, el monomio x^β está en I si y solo si x^β es múltiplo de x^α para algún $\alpha \in A$.*

Proposición 1.15. *Sea I un ideal monomial y sea un polinomio $f \in \mathbb{K}[x_1, \ldots, x_n]$. Entonces, $f \in I$ si, y solo si, todos los términos de f están en I, o equivalentemente, f es una combinación lineal con coeficientes en $\mathbb{K}$ de monomios en I.*

Teorema 1.16 (**Lema de Dickson**). *Sea $A \subset \mathbb{Z}^n_{\geq 0}$ y un ideal monomial $I = \langle x^\alpha : \alpha \in A\rangle$. Entonces I puede escribirse de la siguiente forma $I = \langle x^{\alpha_1}, \ldots, x^{\alpha_s}\rangle$ con $\alpha_1, \ldots, \alpha_s \in A$.*
En particular, I está generado por una familia finita de monomios.

Capítulo 2

Bases de Gröbner

A continucación se introducen las bases de Gröbner, una herramienta fundamental en álgebra computacional y programación entera. Se presentan sus propiedades fundamentales y se muestran aplicaciones prácticas, como la resolución de sistemas de ecuaciones polinómicas. Destaca la importancia de las bases de Gröbner en la resolución de problemas de programación lineal entera.

2.1. Definición y propiedades de las bases de Gröbner

Podemos encontrar las demostraciones de los resultados que se enuncian a continuación en el libro [3, p.77 - p.94]. Estos resultados y sus demostraciones también se han visto en el grado. En toda esta sección fijamos un orden monomial $>$.

Definición 2.1. Sea S un subconjunto no vacío de $\mathbb{K}[x_1, \ldots, x_n]$.

i) Denotamos por $LT(S)$ al conjunto de términos principales de los elementos de S. Es decir, $LT(S) = \{cx^\alpha : \text{ existe } f \in S \text{ con } cx^\alpha = LT(f)\}$.

ii) Denotamos $\langle LT(S)\rangle$ al ideal generado por los elementos de $LT(S)$.

Tenemos que el conjunto $LT(S)$ genera un ideal monomial. Si tenemos $S = \langle f_1, \ldots, f_s\rangle$, obviamente se tiene que $\langle LT(f_1), \ldots, LT(f_s)\rangle \subset \langle LT(S)\rangle$, pero el otro contenido no siempre se cumple.

Proposición 2.2. *Sea $I \subset \mathbb{K}[x_1, \ldots, x_n]$ ideal no nulo.*

i) $\langle LT(I)\rangle$ es un ideal monomial.

ii) Existen $g_1, \ldots, g_s \in I$ tales que $\langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_s)\rangle$.

Definición 2.3. Sea I un ideal en $\mathbb{K}[x_1, \ldots, x_n]$. Una familia generadora $G = \{g_1, \ldots, g_s\} \subseteq I$ se dice *base de Gröbner* de I con respecto a $>$ si $\langle LT(I)\rangle = \langle LT(G)\rangle$.

Proposición 2.4. *Sea $G = \{g_1, \ldots, g_s\}$ una base de Gröbner del ideal I de $\mathbb{K}[x_1, \ldots, x_n]$. Para todo $f \in \mathbb{K}[x_1, \ldots, x_n]$ existe un único $r \in \mathbb{K}[x_1, \ldots, x_n]$ verificando*

i) Existe $g \in I$ tal que $f = g + r$.

ii) Ningún término de r es divisible por ningún $LT(g_i) \quad i = 1, \ldots, s$.

Esta proposición afirma que dada una base de Gröbner $G = \{g_1, \ldots, g_s\}$, si dividimos un polinomio f entre G, no importa entre qué g_i dividamos primero, el resto obtenido será siempre único.

Corolario 2.5. *Sea $G = \{g_1, \ldots, g_s\}$ una base de Gröbner del ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$ y sea $f \in \mathbb{K}[x_1, \ldots, x_n]$. Entonces $f \in I$ si y solo si el resto de la división de f por G es nulo.*

Como consecuencia, toda base de Gröbner de un ideal I es una familia generadora de I.

Notación. A partir de ahora el resto de f al dividir por la base de Gröbner G se denotará $r = \bar{f}^G$.

Ahora nuestro objetivo es enunciar un algoritmo que caracterice cuando una familia generadora de un ideal es una base de Gröbner. Para ello, construiremos esta nueva base a partir de una familia generadora del ideal dada.

Definición 2.6. Sean f y g polinomios no nulos en $\mathbb{K}[x_1, \ldots, x_n]$, con $multideg(f) = \alpha$ y $multideg(g) = \beta$.

- Definimos $\gamma = (\gamma_1, \ldots, \gamma_n)$ con $\gamma_i = \max\{\alpha_i, \beta_i\}$ para cada i. Llamamos a x^γ *mínimo común múltiplo (least common multiple)* de $LM(f)$ y $LM(g)$, se denota como $x^\gamma = LCM(LM(f), LM(g))$.
- El *S-polinomio* de f y g es $\quad S(f,g) = \dfrac{x^\gamma}{LT(f)} f - \dfrac{x^\gamma}{LT(g)} g.$

Teorema 2.7 (Criterio de Buchberger). *Sea I un ideal no nulo de $\mathbb{K}[x_1, \ldots, x_n]$. Una base $G = \{g_1, \ldots, g_s\}$ de I es base de Gröbner para I si y solo si para todo $i \neq j$ se tiene que $\overline{S(g_i, g_j)}^G = 0$.*

Este criterio no solo proporciona una forma de decidir si un conjunto generador es una base de Gröbner, sino que a partir de él obtenemos el siguiente resultado que nos ayuda a construir una base de Gröbner cuando el resto obtenido al dividir $S(g_i, g_j)$ entre G no es cero.

Teorema 2.8 (Algoritmo de Buchberger). *Sea $I = \langle f_1, \ldots, f_s \rangle$ un ideal de $\mathbb{K}[x_1, \ldots, x_n]$. Entonces una base de Gröbner de I se puede construir en un número finito de pasos como sigue*

1. *Para cada par i,j se considera el resto que se obtiene al dividir $S(f_i, f_j)$ entre $f_1, \ldots, f_s$.*
2. *Añadimos a la familia todos los restos no nulos.*

Repetimos estos pasos hasta que todos los restos sean nulos, entonces la familia obtenida es una base de Gröbner de I.

Ejemplo 2.9. Consideramos en $\mathbb{R}[x, y, z]$ el orden graduado lexicográfico con $x > y > z$. Buscamos una base de Gröbner del ideal $I = \langle x + y + z, x + 2y - 3 \rangle$. Comenzamos calculando el S-polinomio de $f_1 = x + y + z$ y $f_2 = x + 2y - 3$. Se obtiene $S(f_1, f_2) = -y + z + 3$, como ninguno de los monomios que lo forman es divisible por $LT(f_1) = LT(f_2)$, la división es trivial y el resto es el propio polinomio $-y + z + 3$. Por tanto, añadimos a la familia $f_3 = -y + z + 3$ y calculamos sus respectivos S-polinomios. Se tiene que $S(f_1, f_3) = xz + y^2 + yz + 3x$, al dividirlo por f_1 y f_3 obtenemos que el resto es 0; y $S(f_2, f_3) = xz + 2y^2 + 3x - 3y$ que al dividirlo por f_1,f_2 y f_3 tenemos que el resto es también 0. Por tanto, tenemos que $\{f_1, f_2, f_3\}$ es base de Gröbner para el ideal I.

A veces, las bases de Gröbner construidas con el algoritmo de Buchberger son más grandes de lo necesario y pueden no ser únicas. A continuación vamos a ver cómo reducir estas bases de Gröbner y veremos que cierto tipo de base de Gröbner es única.

Definición 2.10. Una *base de Gröbner minimal* para un ideal I es una base de Gröbner G del ideal I que cumple

i) $LC(p) = 1$ para todo $p \in G$.

ii) Para todo $p \in G$, se tiene que $LT(p) \notin \langle LT(G-p)\rangle$.

Un ideal puede tener distintas bases de Gröbner minimales. Para conseguir la unicidad consideraremos las siguientes bases de Gröbner minimales.

Definición 2.11. Una *base de Gröbner reducida* para un ideal I es una base de Gröbner G del ideal I que cumple

i) $LC(p) = 1$ para todo $p \in G$.

ii) Para todo $p \in G$, ningún monomio de p está en $\langle LT(G-p)\rangle$.

Notemos que por definición toda base reducida es también minimal.

Proposición 2.12. *Sea $I \neq 0$ un ideal de $\mathbb{K}[x_1, \ldots, x_n]$. Entonces, I tiene una única base de Gröbner reducida.*

Ejemplo 2.13. Vamos a calcular la base de Gröbner minimal y reducida del Ejemplo 2.9. Partimos de la base de Gröbner $\{x+y+z, x+2y-3, -y+z+3\}$ del ideal $I = \langle x+y+z, x+2y-3\rangle$. Ahora vamos a calcular la base minimal, para ello tenemos que el ideal de términos líderes es $\langle LT(I)\rangle = \langle x, x, -y\rangle = \langle x, y\rangle$. Entonces, nos quedamos solo con un polinomio con término líder x y además, transformamos $-y + z + 3$ en un polinomio con coeficiente líder 1. Así, $\{x + y + z, y - z - 3\}$ es base de Gröbner minimal de I. A partir de esta base, vamos a conseguir la reducida. Para ello, tenemos que modificar los polinomios que tengan términos que sean múltiplos de algún término líder. Entonces, tomando $g_1 = x + y + z$ y $g_2 = y - z - 3$ y haciendo $g_1 - g_2 = x + 2z + 3$ se tiene un polinomio que cumple lo que buscábamos. Por tanto, $\{x + 2z + 3, y - z - 3\}$ es base de Gröbner reducida de I. También podríamos haber tomado como base minimal $\{x + 2y - 3, y - z - 3\}$ llegando a la misma base de Gröbner reducida.

2.2. Teoría de la Eliminación

Para poder aplicar las bases de Gröbner a problemas de programación lineal entera necesitamos algunos resultados de teoría de la Eliminación.

Definición 2.14. Sea $I \trianglelefteq \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$, se denomina *ideal de eliminación y-ésimo* de I al ideal de $\mathbb{K}[y_1, \ldots, y_m]$ definido por

$$I_y = I \cap \mathbb{K}[y_1, \ldots, y_m].$$

Teorema 2.15 (Teorema de Eliminación). *Sea $I \trianglelefteq \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ y sea G una base de Gröbner reducida de I para cualquier orden $>$ donde las variables x sean mayores que las variables y. Entonces, $G_y = G \cap \mathbb{K}[y_1, \ldots, y_m]$ es una base de Gröbner reducida de I_y para el orden inducido por $>$.*

Demostración. Sea $I \subset \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ un ideal y G una base de Gröbner de I respecto del orden definido $>$. Sabemos que $G \subset I$, entonces $G_y \subset I_y$. Por definición de base de Gröbner tenemos que ver que $\langle LT(I_y)\rangle = \langle LT(G_y)\rangle$. Lo probamos por doble contenido.
$\supseteq$) Como $G_y \subset I_y$, es obvio que $\langle LT(G_y)\rangle \subseteq \langle LT(I_y)\rangle$.
$\subseteq$) Veamos que para cualquier $f \in I_y$, el término líder $LT(f)$ es divisible por $LT(g)$ para algún $g \in G_y$. Como $f \in I_y \subset I$ y G es base de Gröbner de I, existirá algún $g \in G$ tal que $LT(f)$ es divisible por $LT(g)$. Ya que $f \in I_y$, entonces $LT(f) \in \mathbb{K}[y_1, \ldots, y_m]$ y necesariamente $LT(g) \in \mathbb{K}[y_1, \ldots, y_m]$. Como estamos usando un orden donde $x > y$ y $LT(g) \in \mathbb{K}[y_1, \ldots, y_m]$, entonces los demás términos de g no contienen ninguna variable $x_1, \ldots, x_n$. Por tanto, $g \in \mathbb{K}[y_1, \ldots, y_m]$, luego $g \in G_y$. Así, para cualquier $f \in I_y$ su término líder $LT(f)$ es divisible por $LT(g)$ para algún $g \in G_y$. Concluimos que G_y es una base de Gröbner de I_y, además es una base de Gröbner reducida por ser G reducida. □

2.3. Introducción a la aplicación de las bases de Gröbner

Para entender la aplicación de las bases de Gröbner y de Graver necesitamos recordar algunas nociones de optimización y programación entera.

Definición 2.16. Un *problema de minimización*, o *problema de optimización*, tiene la forma

$$\begin{aligned} &\text{minimiza } f(x) \\ &\text{sujeto a } g_i(x) = b_i \quad i = 1, \dots, m. \end{aligned}$$

Aquí, x es un vector $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ que se denomina variable de decisión del problema y f es una función $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ que se llama función objetivo. Las funciones $g_i : \mathbb{R}^n \longrightarrow \mathbb{R} \quad i = 1, \dots, m$ son las restricciones y las variables $b_1, \dots, b_m \in \mathbb{R}$ son las constantes.
El conjunto $\{x \in \mathbb{R}^n : g_i(x) = b_i \quad i = 1, \dots, m\}$ se llama *región de factibilidad* del problema y se denota como $\mathcal{F}$. Además, nos podemos restringir al caso de minimizar funciones ya que máx $(f(x)) = -$mín $(-f(x))$.

Nota. En la versión más general, estos problemas de optimización pueden tener restricciones de la forma $g_i(x) \geq b_i$ o $g_i(x) \leq b_i$ pero utilizando las llamadas variables de holgura, estas se suman a uno de los miembros de cada restricción consiguiendo que los miembros sean iguales.

Definición 2.17. Un problema de optimización se denomina *problema de programación lineal* si la función objetivo y las funciones de restricción son lineales, es decir, si satisfacen la igualdad $g_i(\alpha x + \beta y) = \alpha g_i(x) + \beta g_i(y)$ para todo $x, y \in \mathbb{R}^n$ y todo $\alpha, \beta \in \mathbb{R}$.
Si esto no ocurre, se llama *problema de programación no lineal.*

Otro tipo importante de problemas de optimización son los *problemas de optimización convexos* donde la función objetivo y las funciones de restricción son convexas, es decir, satisfacen la inecuación $g_i(\alpha x + \beta y) \leq \alpha g_i(x) + \beta g_i(y)$ para todo $x, y \in \mathbb{R}^n$ y todo $\alpha, \beta \in \mathbb{R}$, con $\alpha + \beta = 1$ y $\alpha, \beta \geq 0$.

La convexidad es más general que la linealidad. Dado que cualquier problema de programación lineal es, por tanto, un problema de optimización convexo, podemos considerar que la optimización convexa es una generalización de la programación lineal.

Definición 2.18. Un problema de Programación Lineal Entera (PLE) se escribe de forma estándar como

$$\begin{aligned} \text{mín} \quad & c(\sigma_1, \dots, \sigma_m) \\ \text{s.a:} \quad & a_{11}\sigma_1 + a_{12}\sigma_2 + \cdots + a_{1m}\sigma_m = b_1 \\ & a_{21}\sigma_1 + a_{22}\sigma_2 + \cdots + a_{2m}\sigma_m = b_2 \\ & \quad \vdots \qquad \vdots \qquad\quad \vdots \qquad \vdots \\ & a_{n1}\sigma_1 + a_{n2}\sigma_2 + \cdots + a_{nm}\sigma_m = b_n \end{aligned} \tag{2.1}$$

donde se busca una solución $\sigma = (\sigma_1, \dots, \sigma_m) \in \mathbb{Z}^m$ que satisfaga el sistema $A \cdot \sigma^T = b$, con $A \in \mathbb{Z}^{n \times m}$ la matriz de coeficientes son $a_{ij} \in \mathbb{Z}$ y b el vector con coeficientes $b_i \in \mathbb{Z}$. Además, σ debe minimizar la función costo dada por $c(\sigma_1, \dots, \sigma_m) = \sum_{j=1}^{m} c_j \sigma_j$ con $c_i \in \mathbb{Z}$.

Ahora nos vamos a centrar en cómo resolver este problema utilizando bases de Gröbner. Comenzamos considerando el caso en el que tanto las entradas de A, como las de b, como las de la solución, son no negativas, es decir, $A \in \mathbb{Z}_{\geq 0}^{n \times m}, b \in \mathbb{Z}_{\geq 0}^{n}$ y $\sigma \in \mathbb{Z}_{\geq 0}^{m}$ y transformamos el PLE estándar en un problema de ecuaciones polinómicas. Para expresar el problema en términos de polinomios, asignamos una variable x_i a cada restricción de (2.1) y denotamos cada fila de A como a_i. Así, cada ecuación se escribe de forma equivalente como

$$x_i^{a_i \cdot \sigma} = x_i^{b_i} \quad \text{con } i = 1, \dots, n. \tag{2.2}$$

donde $a_i \cdot \sigma$ es el producto escalar habitual.

Multiplicamos entre sí las n expresiones de (2.2) correspondientes a cada restricción y se tiene

$$x_1^{a_1\cdot\sigma}\cdots x_n^{a_n\cdot\sigma} = x_1^{b_1}\cdots x_n^{b_n}.$$

Podemos reescribir esta expresión y escribirla como

$$\prod_{i=1}^{n}\left(\prod_{j=1}^{m} x_i^{a_{ij}\sigma_j}\right) = \prod_{j=1}^{m}\left(\prod_{i=1}^{n} x_i^{a_{ij}}\right)^{\sigma_j} = \prod_{i=1}^{n} x_i^{b_i}. \tag{2.3}$$

Entonces, que $(\sigma_1,\ldots,\sigma_m)$ cumpla todas las restricciones del sistema (2.1) es equivalente a que se cumpla la igualdad de (2.3).

Definimos el homomorfismo de $\mathbb{K}$-álgebras $\phi : \mathbb{K}[y_1,\ldots,y_m] \longrightarrow \mathbb{K}[x_1,\ldots,x_n]$ dado por

$$\phi(y_j) = f_j = \prod_{i=1}^{n} x_i^{a_{ij}} = x^{a_j}, \quad \text{con } j = 1,\ldots,m. \tag{2.4}$$

Tenemos que la imagen de ϕ está dada por $Im(\phi) = \mathbb{K}[f_1,\ldots,f_m]$ y el núcleo de ϕ es el ideal $Ker(\phi) = \{h \in \mathbb{K}[y_1,\ldots,y_m] : \phi(h) = 0\}$.

Proposición 2.19. *Sea $\mathbb{K}$ un cuerpo y supongamos que estamos en las condiciones del problema (2.1). Consideramos el homomorfismo ϕ definido en (2.4). Entonces, $\sigma = (\sigma_1,\ldots,\sigma_m)$ es solución del sistema (2.1) si y solo si $\phi(y^\sigma) = \phi(y_1^{\sigma_1}\cdots y_m^{\sigma_m}) = x_1^{b_1}\cdots x_n^{b_n} = x^b$.*

Demostración. Sabemos que $(\sigma_1,\ldots,\sigma_m)$ es solución del sistema (2.1) si se cumple la siguiente igualdad en el anillo $\mathbb{K}[x_1,\ldots,x_n]$

$$\prod_{j=1}^{m}\left(\prod_{i=1}^{n} x_i^{a_{ij}}\right)^{\sigma_j} = \prod_{i=1}^{n} x_i^{b_i}.$$

Por definición de $\phi(y_j)$ sustituyendo se obtiene la igualdad buscada. □

2.4. Programación lineal entera general

Para poder considerar coeficientes enteros posiblemente negativos en el problema (2.1) es necesario introducir la siguiente teoría, ya que en un polinomio ordinario no podemos tener exponentes negativos. Definimos así los polinomios de Laurent, combinaciones lineales finitas $f = \sum_{\alpha\in\mathbb{Z}^n} c_\alpha x^\alpha$, donde $c_\alpha \in \mathbb{K}$ y $\alpha = (\alpha_1,\ldots,\alpha_n) \in \mathbb{Z}^n_{\geq 0}$ es un vector entero. El anillo conmutativo que forman estos polinomios se denota $\mathbb{K}[x_1^{\pm 1},\ldots,x_n^{\pm 1}]$.

Sean $A \in \mathbb{Z}^{n\times m}, b \in \mathbb{Z}^n$ como en (2.1), consideramos la aplicación $\pi : \mathbb{Z}^m \longrightarrow \mathbb{Z}^n$ dada por $\pi(u) = A \cdot u^T$. Por tanto, el problema de programación lineal de minimización visto en (2.1) es equivalente a encontrar un punto $\pi^{-1}(b)$ que minimice la función costo dada.

Trabajando con polinomios de Laurent podemos generalizar el homomorfismo ϕ

$$\begin{aligned}\phi : \mathbb{K}[y_1,\ldots,y_m] &\longrightarrow \mathbb{K}[x_1^{\pm 1},\ldots,x_n^{\pm 1}]\\ y_j &\longmapsto x^{a_j}.\end{aligned}$$

Relacionando π con ϕ tenemos que $\phi(y^u) = x^{\pi(u)}$. El núcleo de ϕ se denomina *ideal tórico de A* y se denota I_A. Ahora, nos interesaría saber cómo son las bases de Gröbner reducidas de este ideal.

Lema 2.20. *El ideal tórico I_A está generado como $\mathbb{K}$-espacio vectorial por el conjunto de binomios*

$$\{y^u - y^v : u, v \in \mathbb{N}^m \text{ con } \pi(u) = \pi(v)\}.$$

Demostración. Un binomio $y^u - y^v$ está en I_A si y solo si $\phi(y^u - y^v) = \phi(y^u) - \phi(y^v) = x^{\pi(u)} - x^{\pi(v)} = 0$, es decir, si y solo si $\pi(u) = \pi(v)$. Nos queda probar que todo polinomio de I_A es una combinación lineal de binomios de la forma $y^u - y^v$ con coeficientes en $\mathbb{K}$. Fijamos un orden monomial $>$ en $\mathbb{K}[y_1, \ldots, y_m]$ y razonamos por reducción al absurdo. Suponemos $f \in I_A$ que no puede ser escrito como combinación lineal de binomios de la forma anterior. De entre todos los posibles f en esas condiciones elegimos f tal que $LM_>(f) = y^u$ es lo más pequeño posible con respecto a $>$. Como $f \in I_A$ sabemos que

$$0 = \phi(f) = x^{\pi(u)} + \text{ otros términos}$$

En particular sabemos que el término $x^{\pi(u)}$ debe cancelarse. Entonces, existe un monomio y^v en f, con $y^u > y^v$ tal que $\pi(u) = \pi(v)$. Sabemos que $f' = f - (y^u - y^v)$ no puede escribirse como combinación $\mathbb{K}$-lineal de binomios de la forma anterior ya que sino f podría escribirse también. Por definición tenemos que $LM_>(f) < LM_>(f')$, por tanto tenemos una contradicción. □

Todo vector $u \in \mathbb{Z}^m$ se puede poner de forma única como $u = u^+ - u^-$ donde u^+ y u^- son no negativos y tienen soporte disjunto. Esto es, $u^+(j) = \text{máx}(0, u(j))$ y $u^-(j) = \text{máx}(0, -u(j))$ con $j = 1, \ldots, m$.

Notemos que $\pi(u) = \pi(u^+) - \pi(u^-)$. Por definición, $ker(\pi)$ es el conjunto de vectores $u \in \mathbb{Z}^m$ tales que $\pi(u) = 0$, equivalentemente, $\pi(u^+) = \pi(u^-)$. Así, podemos reformular el Lema 2.20 como sigue.

Corolario 2.21.

$$I_A = \langle y^{u^+} - y^{u^-} : u \in ker(\pi) \rangle.$$

Como $ker(\pi) \subseteq \mathbb{Z}^m$, existen vectores $v_1, \ldots, v_t \in ker(\pi)$ que forman una base, es decir, para todo $u \in ker(\pi)$ existen coeficientes $a_1, \ldots, a_t \in \mathbb{Z}$ únicos con $u = a_1 v_1 + \cdots a_t v_t$. Siendo $\mathcal{B} = \{v_1, \ldots, v_t\}$, podemos enunciar entonces el siguiente teorema.

Teorema 2.22.

$$I_A = \langle y^{v^+} - y^{v^-} : v \in \mathcal{B} \rangle.$$

Demostración. $\supseteq$) Obvio porque la base está formada por elementos de I_A.
$\subseteq$) Sea $y^{u^+} - y^{u^-} \in I_A$ y $u = u^+ - u^- = a_1 v_1 + \cdots a_t v_t$. Reordenando si es necesario, podemos suponer $a_1, \ldots, a_s \geq 0$, $a_{s+1}, \ldots, a_t < 0$. Luego, $u^+ = a_1 v_1^+ + \cdots + a_s v_s^+ - a_{s+1} v_{s+1}^- - \cdots - a_t v_t^-$ y $u^- = a_1 v_1^- + \cdots + a_s v_s^- - a_{s+1} v_{s+1}^+ - \cdots - a_t v_t^+$. Así que por el Lema 1.7, $y^{u^+} - y^{u^-} \in \langle y^{v_1^+} - y^{v_1^-}, \ldots, y^{v_t^+} - y^{v_t^-} \rangle$. □

Lema 2.23. *Sean $f = y^{u^+} - y^{u^-}$ y $g = y^{v^+} - y^{v^-}$ con $u, v \in ker(\pi)$ no nulos. Entonces, el resto al dividir f entre g es otro binomio de la misma forma, o es nulo.*

Demostración. Supongamos sin pérdida de generalidad que $y^{u^+} > y^{u^-}$ y $y^{v^+} > y^{v^-}$, veamos cómo es la división de f entre g. Sabemos que $LT(g) = y^{v^+}$, luego si y^{v^+} no divide ni a y^{u^+} ni a y^{u^-} entonces f es el propio resto de la división, que es un binomio. Si y^{v^+} divide a y^{u^+} pero no a y^{u^-}, entonces $y^{u^+} = y^{v^+} y^\beta$ siendo β no nulo y de entradas no negativas. Entonces al dividir f entre g se tiene $f = y^\beta g + y^\beta y^{v^-} - y^{u^-}$, por lo que el resto de la division es el binomio $y^\beta y^{v^-} - y^{u^-}$. Si y^{v^+} divide a y^{u^-} pero no a y^{u^+} es análogo al anterior. Por último, si y^{v^+} divide a y^{u^+} y a y^{u^-}, entonces $y^{u^+} = y^{v^+} y^{\beta_1}$ y $y^{u^-} = y^{v^+} y^{\beta_2}$ y dividiendo se tiene $f = (y^{\beta_1} - y^{\beta_2}) g + y^{\beta_1} y^{v^-} - y^{\beta_2} y^{v^-}$ luego vemos que claramente el resto es un binomio de la misma forma. □

Proposición 2.24. *Para cada orden monomial $>$, hay un conjunto finito de vectores $\mathcal{V}_> \subset ker(\pi)$ tal que la base de Gröbner reducida de I_A con respecto a $>$ es $\{y^{u^+} - y^{u^-} : u \in \mathcal{V}_>\}$.*

Demostración. Por el Teorema 2.22 sabemos que existe un subconjunto finito $\mathcal{B}$ de $ker(\pi)$ de manera que los binomios $y^{u^+} - y^{u^-}$ con $u \in \mathcal{B}$ generan I_A. Una vez tenemos el ideal I_A generado por un número finito de binomios de la forma $y^{u^+} - y^{u^-}$ les aplicamos el algoritmo de Buchberger 2.8 para encontrar la base de Gröbner con respecto al orden monomial $>$. Tenemos que probar que durante la ejecución del algoritmo, las operaciones de reducción y la formación de S-polinomios preservan la estructura binomial. Sean $f = y^{u^+} - y^{u^-}$ y $g = y^{v^+} - y^{v^-}$ y podemos suponer sin pérdida de generalidad que $y^{u^+} > y^{u^-}$ y $y^{v^+} > y^{v^-}$. Sean entonces $multideg(f) = u^+$ y $multideg(g) = v^+$, entonces tomamos γ con $\gamma_i = máx\{u_i^+, v_i^+\}$ tal que $x^\gamma = LCM(u^+, v^+)$. Por tanto, $x^\gamma = y^{\delta_u} y^{u^+}$ y $x^\gamma = y^{\delta_v} y^{v^+}$ para ciertos δ_u, δ_v vectores con entradas no negativas. Calculamos el S-polinomio de f y g como sigue

$$S(f,g) = y^{\delta_u}(y^{u^+} - y^{u^-}) - y^{\delta_v}(y^{v^+} - y^{v^-}) = y^{\delta_v} y^{v^-} - y^{\delta_u} y^{u^-}.$$

Vemos que se conserva la estructura binomial. Ahora, debemos dividir el resultado del S-polinomio que sabemos que es un binomio entre los elementos que generan I_A, como son binomios, por el Lema 2.23 tenemos que el resto también es un binomio. Por tanto, la base de Gröbner de I_A claramente va a estar formada por binomios. A su vez, cualquier base de Gröbner minimal de I_A va a estar formada por binomios ya que solo debemos eliminar aquellos binomios cuyo término líder divida al término líder de otro binomio de la base. Por último, veamos como es la base de Gröbner reducida. Supongamos que los binomios f y g están en la base minimal y y^{u^-} divide a y^{v^+}, luego $y^{u^-} = y^\beta y^{v^+}$. Entones, podemos tomar una nueva f dada por $f = y^{u^+} - y^{u^-} + y^\beta(y^{v^+} - y^{v^-}) = y^{u^+} - y^\beta y^{v^-}$ que ya no tiene ningún término que divida a y^{v^+} y también es un binomio. Por tanto, la base de Gröbner reducida de I_A también estará formada por binomios. □

2.5. Programación lineal entera no negativa

En la sección anterior hemos visto cómo generalizar al caso donde $A, b \in \mathbb{Z}$, ahora nos restringimos al caso donde $A, b \in \mathbb{N}$. Ya hemos visto cómo se transforma un PLE en polinomios, ahora vamos a utilizar estos polinomios para calcular bases de Gröbner, pero primero necesitamos conocer algunos resultados.

Recordemos que por la Proposición 2.19 para resolver nuestro problema necesitamos ver cuando el producto $x_1^{b_1} \cdots x_n^{b_n}$ está en la imagen de ϕ y buscar sus antiimágenes.

Teorema 2.25. *Sea $\phi : \mathbb{K}[y_1, \ldots, y_m] \longrightarrow \mathbb{K}[x_1, \ldots, x_n]$ definida como en la Proposición 2.19 y sea el ideal $I = \langle y_1 - f_1, \ldots, y_m - f_m \rangle \subseteq \mathbb{K}[y_1, \ldots, y_m, x_1, \ldots, x_n]$. Sea G una base de Gröbner reducida de I respecto a un orden monomial donde las variables x son mayores que las variables y. Sea ϕ el homomorfismo definido en (2.4) y sea $f \in \mathbb{K}[y_1, \ldots, y_m, x_1, \ldots, x_n]$ y $g = \bar{f}^G$. Entonces*

a) Si $g \in \mathbb{K}[y_1, \ldots, y_m]$, $\phi(g) = f$.

b) Si $f \in Im(\phi)$, entonces $g \in \mathbb{K}[y_1, \ldots, y_m]$ y si además $f = \phi(y_1^{\sigma_1} \cdots y_m^{\sigma_m})$ para ciertos $\sigma_1, \ldots, \sigma_m$, entonces g es un monomio.

Demostración. a) Suponemos $g \in \mathbb{K}[y_1, \ldots, y_m]$ donde $\overline{f}^G = g$ y G base de Gröbner reducida de I. Entonces, $\overline{f - g}^G = 0$ y así $f - g \in I$. Por definición del ideal I podemos escribir

$$f(x_1, \ldots, x_n) - g(y_1, \ldots, y_m) = \sum_{i=1}^{m} (y_i - f_i(x_1, \ldots, x_n)) q_i(y_1, \ldots, y_m, x_1, \ldots, x_n). \tag{2.5}$$

Tomamos el homomorfismo ϕ y lo extendemos a la aplicación $\Phi : \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m] \longrightarrow \mathbb{K}[x_1, \ldots, x_n]$ dada por $\Phi(x_i) = x_i, \Phi(y_j) = f_j$. Aplicamos Φ primero a la parte derecha de la igualdad dada en (2.5)

$$\begin{aligned}
&\Phi(\sum_{i=1}^{m}(y_i - f_i(x_1, \ldots, x_n))q_i(y_1, \ldots, y_m, x_1, \ldots, x_n)) \\
&= \sum_{i=1}^{m}\Phi((y_i - f_i(x_1, \ldots, x_n))q_i(y_1, \ldots, y_m, x_1, \ldots, x_n)) \\
&= \sum_{i=1}^{m}(f_i(x_1, \ldots, x_n) - f_i(x_1, \ldots, x_n))q_i(f_1, \ldots, f_m, x_1, \ldots, x_n) = 0.
\end{aligned}$$

Entonces, juntando lo obtenido y aplicando Φ a la otra expresión de la igualdad de (2.5) tenemos

$$\begin{aligned}
0 &= \Phi(f(x_1, \ldots, x_n) - g(y_1, \ldots, y_m)) = \Phi(f(x_1, \ldots, x_n)) - \Phi(g(y_1, \ldots, y_m)) \\
&= f(x_1, \ldots, x_n) - g(f_1, \ldots, f_m)
\end{aligned}$$

Por tanto, $f(x_1, \ldots, x_n) = g(f_1, \ldots, f_m) = \Phi(g(y_1, \ldots, y_m)) = \phi(g(y_1, \ldots, y_m))$.

b) Suponemos que $f \in \mathbb{K}[x_1, \ldots, x_n]$ está en la imagen de ϕ, es decir, existe algún $h \in \mathbb{K}[y_1, \ldots, y_m]$ tal que $\phi(h) = f$. Por definición de imagen tenemos que $f = \phi(h(y_1, \ldots, y_m)) = h(f_1, \ldots, f_m)$.
Consideramos $f(x_1, \ldots, x_n) - h(y_1, \ldots, y_m) \in \mathbb{K}[y_1, \ldots, y_m, x_1, \ldots, x_n]$. Ya que $f(x_1, \ldots, x_n) - h(y_1, \ldots, y_m) = h(f_1, \ldots, f_m) - h(y_1, \ldots, y_m)$, vemos que este polinomio está formado por monomios de la forma $f_1^{\alpha_1} \cdots f_m^{\alpha_m} - y_1^{\alpha_1} \cdots y_m^{\alpha_m}$. Entonces, por el Lema 1.7 se tiene que $f(x_1, \ldots, x_n) - h(y_1, \ldots, y_m) \in I$. Ahora tomamos G una base de Gröbner reducida como en el enunciado. Sabemos que

$$\overline{f(x_1, \ldots, x_n) - h(y_1, \ldots, y_m)}^{G} = 0$$

porque hemos visto que $f(x_1, \ldots, x_n) - h(y_1, \ldots, y_m) \in I$. Esto implica que $\overline{f(x_1, \ldots, x_n)}^{G} = \overline{h(y_1, \ldots, y_m)}^{G}$, y por la Proposición 2.4 sabemos que este resto es único.
Sea $\overline{f(x_1, \ldots, x_n)}^{G} = \overline{h(y_1, \ldots, y_m)}^{G} = g$. Por estar h en $\mathbb{K}[y_1, \ldots, y_m]$, h solo puede reducirse por polinomios de G que tengan como término líder solo variables y, es decir, los polinomios de G que aparecen con coeficiente no nulo en el Algoritmo de la división, Teorema 1.12, tienen que tener término líder con variables y solamente. Como hemos considerado un orden monomial donde $x > y$ entonces estos polinomios no tienen variables x, es decir, están en $\mathbb{K}[y_1, \ldots, y_m]$. Como h está en $\mathbb{K}[y_1, \ldots, y_m]$ y estos polinomios también, deducimos que $g \in \mathbb{K}[y_1, \ldots, y_m]$. Ya tenemos la primera parte.

Ahora suponemos $f = \phi(y_1^{\sigma_1} \cdots y_m^{\sigma_m}) = f_1^{\sigma_1} \cdots f_m^{\sigma_m}$. Tenemos $f - y_1^{\sigma_1} \cdots y_m^{\sigma_m} = f_1^{\sigma_1} \cdots f_m^{\sigma_m} \in I$ por el Lema 1.7, luego $\bar{f}^G = \overline{y_1^{\sigma_1} \cdots y_m^{\sigma_m}}^{G}$. El mismo razonamiento de la primera parte implica que la reducción $\overline{y_1^{\sigma_1} \cdots y_m^{\sigma_m}}^{G}$ se puede hacer en $\mathbb{K}[y_1, \ldots, y_m]$ y mediante polinomios de $G_y = G \cap \mathbb{K}[y_1, \ldots, y_m]$ que es base de Gröbner por el Teorema 2.15 luego

$$\bar{f}^G = \overline{y^\sigma}^{G} = \overline{y^\sigma}^{G_y} \tag{2.6}$$

donde $\sigma = (\sigma_1, \ldots, \sigma_m)$. Ahora, por la Proposición 2.24, la base reducida G_y está formada por binomios $y^{u^+} - y^{u^-}$. Supongamos que al reducir un monomio como en (2.6) no siempre se obtiene un monomio. Elegimos y^σ lo menor posible de manera que esto falle (se puede hacer por el principio de buen orden). Entonces $\overline{y^\sigma}^{G} \neq y^\sigma$ luego se puede reducir mediante algún $y^{u^+} - y^{u^-} \in G_y$ con $y^{u^+} > y^{u^-}$, es decir, $y^\sigma = y^\alpha(y^{u^+} - y^{u^-}) + y^\alpha y^{u^-}$ con $y^\alpha y^{u^-} < y^\sigma = y^\sigma y^{u^+}$. Como $y^\alpha y^{u^-}$ es un monomio tenemos una contradicción (ya que esto implica que se reduce a un monomio, con lo cual y^σ también). □

Dados A y b con entradas no negativas y combinando los últimos resultados vistos y considerando I y G como los hemos definido en el Teorema 2.25, deducimos el siguiente teorema.

Teorema 2.26. *Dados A, b no negativos como en (2.1) existe una solución $\sigma = (\sigma_1, \ldots, \sigma_m) \in \mathbb{Z}^m_{\geq 0}$ de (2.1) si y solo si $\bar{f}^G \in \mathbb{K}[y_1, \ldots, y_m]$ es un monomio.*

Hemos visto varios resultados que nos ayudarán a encontrar soluciones del sistema (2.1), ahora queremos encontrar una solución que minimice la función costo. Para ello, será útil el siguiente lema.

Lema 2.27. *Sea $I = \langle y_1 - f_1, \ldots, y_m - f_m \rangle \subseteq \mathbb{K}[y_1, \ldots, y_m, x_1, \ldots, x_n]$ el ideal considerado en el Teorema 2.25. Entonces,*

$$ker(\phi) = I \cap \mathbb{K}[y_1, \ldots, y_m].$$

Demostración. Lo vemos por doble contenido.
$\subseteq$) Sea $g \in ker(\phi) \subseteq \mathbb{K}[y_1, \ldots, y_m]$, como $g \in \mathbb{K}[y_1, \ldots, y_m]$ podemos escribir

$$g = \sum_v c_v y_1^{v_1} \cdots y_m^{v_m}$$

siendo $c_v \in \mathbb{K}, v = (v_1, \ldots, v_m) \in \mathbb{N}^m$ y solo un número finito de coeficientes c_v que no son cero. Ya que $g \in ker(\phi)$, se tiene que $\phi(g(y_1, \ldots, y_m)) = g(f_1, \ldots, f_m) = 0$. Entonces tenemos,

$$g = g - g(f_1, \ldots, f_m) = \sum_v c_v y_1^{v_1} \cdots y_m^{v_m} - \sum_v c_v f_1^{v_1} \cdots f_m^{v_m} = \sum_v c_v (y_1^{v_1} \cdots y_m^{v_m} - f_1^{v_1} \cdots f_m^{v_m}).$$

Por tanto, por el Lema 1.7, cada término del sumatorio está en I. Así, $g \in I \cap \mathbb{K}[y_1, \ldots, y_m]$.
$\supseteq$) Sea $g \in I \cap \mathbb{K}[y_1, \ldots, y_m]$. Como $I = \langle y_1 - f_1, \ldots, y_m - f_m \rangle$ podemos expresar g como sigue

$$g(y_1, \ldots, y_m) = \sum_{i=1}^{m} (y_i - f_i(x_1, \ldots, x_n)) h_i$$

con $h_i \in \mathbb{K}[y_1, \ldots, y_m, x_1, \ldots, x_n]$. Entonces,

$$\phi(g) = g(f_1, \ldots, f_m) = \sum_{i=1}^{m} (f_i(x_1, \ldots, x_n) - f_i(x_1, \ldots, x_n)) h_i = 0.$$

Ya tenemos que $g \in ker(\phi)$. Así queda demostrada la igualdad. □

Con este resultado combinado con la teoría de Eliminación podríamos calcular bases de Gröbner de $ker(\phi)$ a partir de bases de Gröbner de I, como $ker(\phi) = I \cap \mathbb{K}[y_1, \ldots, y_m]$, se obtiene una base de Gröbner de $ker(\phi)$ intersecando la base de Gröbner de I con $\mathbb{K}[y_1, \ldots, y_m]$.

A continuación, buscamos una buena elección del orden monomial. Hasta ahora, el único requisito que usábamos era que las variables x fueran mayores que las variables y. La nueva idea es usar un orden monomial en las variables y que está relacionado con la función costo.

Definición 2.28. Un orden monomial $<_c$ en las variables y se dice *compatible con la función costo* c y la aplicación ϕ si

i) $\phi(y_1^{\sigma'_1} \cdots y_m^{\sigma'_m}) = \phi(y_1^{\sigma_1} \cdots y_m^{\sigma_m})$

ii) $c(\sigma'_1, \ldots, \sigma'_m) < c(\sigma_1, \ldots, \sigma_m)$

implican que $y_1^{\sigma'_1} \cdots y_m^{\sigma'_m} <_c y_1^{\sigma_1} \cdots y_m^{\sigma_m}$.

Este orden monomial nos dará las soluciones del sistema (2.1) con coste mínimo, lo veremos en el siguiente resultado en el que usamos la misma notación que en los Teoremas 2.27 y 2.25.

Proposición 2.29. *Sean G una base de Gröbner del ideal I con respecto a un orden monomial donde $x > y$ y con un orden $>_c$ en las variables y como en la Definición 2.28. Si se cumple que $\overline{x_1^{b_1}\cdots x_n^{b_n}}^{G} = y_1^{\sigma_1}\cdots y_m^{\sigma_m}$, entonces $(\sigma_1,\ldots,\sigma_m)$ es una solución del sistema (2.1) que minimiza la función costo c.*

Demostración. Sea $\overline{x_1^{b_1}\cdots x_n^{b_n}}^{G} = y_1^{\sigma_1}\cdots y_m^{\sigma_m}$. Entonces, $(\sigma_1,\ldots,\sigma_m)$ es solución del sistema (2.1) debido al Teorema 2.25.
Para ver que $(\sigma_1,\ldots,\sigma_m)$ minimiza la función costo c usamos reducción al absurdo. Suponemos que existe otra solución $(\sigma'_1,\ldots,\sigma'_m)$ de (2.1) tal que $\sum_{j=1}^m c_j\sigma'_j < \sum_{j=1}^m c_j\sigma_j$.
Por definición del orden $>_c$ tenemos que $y_1^{\sigma_1}\cdots y_m^{\sigma_m} >_c y_1^{\sigma'_1}\cdots y_m^{\sigma'_m}$, además como $\phi(y_1^{\sigma_1}\cdots y_m^{\sigma_m}) = \phi(y_1^{\sigma'_1}\cdots y_m^{\sigma'_m})$, se tiene que $y_1^{\sigma_1}\cdots y_m^{\sigma_m} - y_1^{\sigma'_1}\cdots y_m^{\sigma'_m} \in ker(\phi)$.
Ahora, usando el Teorema 2.27 sabemos que $ker(\phi) = I\cap\mathbb{K}[y_1,\ldots,y_m]$, por tanto, $y_1^{\sigma_1}\cdots y_m^{\sigma_m} - y_1^{\sigma'_1}\cdots y_m^{\sigma'_m} \in I$.
Como G es una base de Gröbner de I deducimos que $\overline{y_1^{\sigma_1}\cdots y_m^{\sigma_m} - y_1^{\sigma'_1}\cdots y_m^{\sigma'_m}}^{G} = 0$. Por otro lado, tenemos que $y_1^{\sigma_1}\cdots y_m^{\sigma_m} >_c y_1^{\sigma'_1}\cdots y_m^{\sigma'_m}$, entonces $LT(y_1^{\sigma_1}\cdots y_m^{\sigma_m} - y_1^{\sigma'_1}\cdots y_m^{\sigma'_m}) = y_1^{\sigma_1}\cdots y_m^{\sigma_m}$, que está reducido respecto a G. Es decir, hemos llegado a una contradicción ya que $y_1^{\sigma_1}\cdots y_m^{\sigma_m} - y_1^{\sigma'_1}\cdots y_m^{\sigma'_m}$ no puede reducirse a 0 por G.
Concluimos por tanto que $(\sigma_1,\ldots,\sigma_m)$ minimiza la función costo c. □

Ahora vamos a ver cómo definir el orden $<_c$ en los monomios $y's$ para que sea compatible. Queremos que se cumplan las condiciones vistas en la Definición 2.28, vamos a distinguir dos casos:

1. Si los coeficientes c_i de la función costo son no negativos para todo i, elegimos un orden monomial cualquiera $<_\gamma$ en los monomios $y's$ y entonces definimos:

$$y_1^{\sigma_1}\cdots y_m^{\sigma_m} <_c y_1^{\sigma'_1}\cdots y_m^{\sigma'_m} \quad \text{si} \quad c(\sigma_1,\ldots,\sigma_m) < c(\sigma'_1,\ldots,\sigma'_m)$$
$$\text{ó} \quad c(\sigma_1,\ldots,\sigma_m) = c(\sigma'_1,\ldots,\sigma'_m) \quad \text{y} \quad y_1^{\sigma_1}\cdots y_m^{\sigma_m} <_\gamma y_1^{\sigma'_1}\cdots y_m^{\sigma'_m}.$$

   De esta forma ya tenemos definido el orden $<_c$ que necesitábamos. Sin embargo, esto no funciona si alguno de los c_i es negativo ya que no se produce un buen orden.

2. Si tenemos algún c_i negativo hacemos lo siguiente:
   Definimos los elementos d_j como la suma de elementos de la columna j de la matriz A,

$$d_j = \sum_{i=1}^{n} a_{ij} \geq 0 \quad \text{para todo } j = 1,\ldots,m$$

   donde $d_j \geq 0$ ya que hemos asumido que $a_{ij} \geq 0$. Así, construimos una nueva función costo $\tilde{c}$ de la siguiente forma

$$\tilde{c_j} := c_j + \mu d_j \quad \text{para todo } j,$$

   donde $\mu \in \mathbb{Z}^+$ se elige de forma que todos los nuevos coeficientes $\tilde{c_j}$ sean mayores que 0. Entonces, la función $\tilde{c}$ se puede utilizar como en el caso 1 para definir un orden $<_{\tilde{c}}$ en los monomios y que se cumpla de nuevo que el orden $<_{\tilde{c}}$ es compatible.

   Ahora, probamos que el orden $<_{\tilde{c}}$ también cumple las condiciones de la Definición 2.28 para la función c original con lo que podremos tomar $<_c$:=$<_{\tilde{c}}$. Para demostrarlo, hacemos lo siguiente:
   Dadas soluciones $(\sigma_1,\ldots,\sigma_m)$ y $(\sigma'_1,\ldots,\sigma'_m)$ que cumplen

   i) $\phi(y_1^{\sigma_1}\cdots y_m^{\sigma_m}) = \phi(y_1^{\sigma'_1}\cdots y_m^{\sigma'_m})$

ii) $c(\sigma_1,\dots,\sigma_m) < c(\sigma'_1,\dots,\sigma'_m)$

queremos probar que $\tilde{c}(\sigma_1,\dots,\sigma_m) < \tilde{c}(\sigma'_1,\dots,\sigma'_m)$. Con probar esto será sufiente ya que esto implica que $y_1^{\sigma_1}\cdots y_m^{\sigma_m} <_{\tilde{c}} y_1^{\sigma'_1}\cdots y_m^{\sigma'_m}$ por ser $<_{\tilde{c}}$ compatible.

Entonces, sabemos que

$$\tilde{c}(\sigma_1,\dots,\sigma_m) = c(\sigma_1,\dots,\sigma_m) + \mu\sum_{j=1}^{m} d_j\sigma_j < c(\sigma'_1,\dots,\sigma'_m) + \mu\sum_{j=1}^{m} d_j\sigma_j.$$

Por tanto, basta probar que $\sum_{j=1}^{m} d_j\sigma_j = \sum_{j=1}^{m} d_j\sigma'_j$. Para ello, tenemos en cuenta que $(\sigma_1,\dots,\sigma_m)$ y $(\sigma'_1,\dots,\sigma'_m)$ son soluciones y por tanto, $A\cdot\sigma = A\cdot\sigma'$.

Se tiene que

$$\sum_{j=1}^{m} d_j\sigma_j = \sum_{j=1}^{m}\left(\sum_{i=1}^{n} a_{ij}\right)\sigma_j = \sum_{i=1}^{n}\left(\sum_{j=1}^{m} a_{ij}\sigma_j\right) = \sum_{i=1}^{n}\left(\sum_{j=1}^{m} a_{ij}\sigma'_j\right) = \sum_{j=1}^{m} d_j\sigma'_j.$$

Con esto, hemos demostrado que cuando transformamos c en $\tilde{c}$ nada cambia con respecto a la compatibilidad.

A continuación mostramos un algoritmo que resume los resultados anteriores. Será importante encontrar un orden monomial compatible para conseguir la solución óptima.

Algoritmo de bases de Gröbner en programación lineal entera.

1. Elegimos un orden monomial compatible $>$.
2. Calcular una base de Gröbner $\mathcal{G}$ para el ideal $I = \langle y_j - x_1^{\alpha_{1j}}\cdots x_n^{\alpha_{nj}} : j = 1,\dots,m\rangle$ con respecto a un orden monomial compatible con la función costo y donde $x > y$.
3. Encontrar el resto r de la división de $x_1^{b_1}\cdots x_n^{b_n}$ entre G.
4. Si $r \notin \mathbb{K}[y_1,\dots,y_m]$, entonces el sistema (2.1) no tiene solución entera no negativa.
   Si $r = y_1^{\sigma_1}\cdots y_m^{\sigma_m}$, entonces $(\sigma_1,\dots,\sigma_m)$ es una solución del sistema (2.1) que minimiza c.

Hemos visto de forma teórica cómo construir un orden compatible $<_c$. En ocasiones, en la práctica, es mejor transformar el problema de programación entera para poder usar un orden universal como el orden lexicográfico. Para ello, cambiamos el sistema de forma que la función costo sea precisamente una de las variables, concretamente, será la variable a minimizar.

Teorema 2.30. *Cualquier problema de minimización en el que los coeficientes de la función costo sean no negativos se puede transformar en otro problema donde la función costo sea una de las variables, y así podremos aplicar el algoritmo con el orden lexicográfico.*

Demostración. Buscamos una solución σ del problema dado en (2.1) que minimice $c\cdot\sigma$, donde la función costo tiene coeficientes no negativos.

Sean $a_1,\dots,a_n$ las filas de la matriz A. Sabemos que A no tiene ninguna columna nula ya que cada columna representa los valores de una variable. Por ello, podemos razonar que existen enteros $\beta_i \geq 0 \quad i = 1,\dots,n$ tales que

$$\beta_1 a_1 + \cdots + \beta_n a_n = w + c \tag{2.7}$$

para cierto vector $w = (w_1,\dots,w_m)$ con $w_i \geq 0$. Es decir, podemos multiplicar cada fila de A por un escalar de forma que se obtenga un vector mayor o igual que c.

Dados β_i y w como en (2.7), consideramos el siguiente sistema

$$\left(\begin{array}{c|c} A & \begin{matrix}0\\ \vdots\\ 0\end{matrix} \\ \hline w_1 \cdots w_m & 1 \end{array}\right) \left(\begin{array}{c} \sigma_1 \\ \vdots \\ \sigma_m \\ \hline t \end{array}\right) = \left(\begin{array}{c} b_1 \\ \vdots \\ b_m \\ \hline \gamma \end{array}\right) \tag{2.8}$$

siendo $\gamma = \beta_1 b_1 + \cdots + \beta_n b_n$.

Vamos a probar que si $(\sigma_1, \ldots, \sigma_m, t)$ es una solución del sistema (2.8) con el valor de t mínimo, entonces $\sigma = (\sigma_1, \ldots, \sigma_m)$ es solución de (2.1) y $t = c \cdot \sigma$. Por la propia definición del sistema es obvio que se cumple que $A \cdot \sigma = b$. Para ver que $t = c \cdot \sigma$ multiplicamos la ecuación dada en (2.7) por σ y obtenemos

$$\beta_1 a_1 \cdot \sigma + \cdots + \beta_n a_n \cdot \sigma = w \cdot \sigma + c \cdot \sigma. \tag{2.9}$$

Por cumplirse la igualdad $A \cdot \sigma = b$ se tiene que $a_i \cdot \sigma = b_i$ para cada $i = 1, \cdots, n$, además, por ser $(\sigma_1, \ldots, \sigma_m, t)$ solución tenemos también la igualdad $w_1\sigma_1 + \cdots + w_n\sigma_n + t = \beta_1 b_1 + \cdots + \beta_n b_n$. Luego, la ecuación (2.9) es equivalente a $w \cdot \sigma + t = w \cdot \sigma + c \cdot \sigma$ y tenemos que $t = c \cdot \sigma$. Por tanto, este resultado implica que la solución σ de (2.1) con $c \cdot \sigma$ mínimo se obtiene de la solución de (2.8) con t mínimo. □

Ejemplo 2.31. Sean A= $\begin{pmatrix} 1 & 3 & 0 & 5 \\ 2 & 0 & 1 & 0 \end{pmatrix}$, $b = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$ y $c = \begin{pmatrix} 27 & 8 & 40 & 25 \end{pmatrix}$.

Tenemos que buscar una solución σ del sistema $A \cdot \sigma = b$ que minimice la función $c \cdot \sigma$. Tomamos el vector $\beta = \begin{pmatrix} 5 & 40 \end{pmatrix}$ y lo multiplicamos por A obteniendo $\beta \cdot A = \begin{pmatrix} 85 & 15 & 40 & 25 \end{pmatrix}$. Para que se cumpla la igualdad (2.7) tomamos $w = \begin{pmatrix} 58 & 7 & 0 & 0 \end{pmatrix}$, y ahora buscamos una solución del siguiente sistema donde la variable t sea mínima.

$$\left(\begin{array}{cccc|c} 1 & 3 & 0 & 5 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ \hline 58 & 7 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c} \sigma_1 \\ \vdots \\ \sigma_4 \\ \hline t \end{array}\right) = \left(\begin{array}{c} 4 \\ 3 \\ \hline 140 \end{array}\right) \tag{2.10}$$

A continuación resolvemos este sistema usando bases de Gröbner. Para ello, transformamos el sistema en las siguientes ecuaciones polinomicas

$$\begin{cases} x_1^{\sigma_1+3\sigma_2+5\sigma_4} = x_1^4 \\ x_2^{2\sigma_1+\sigma_3} = x_2^3 \\ x_3^{58\sigma_1+7\sigma_2+t} = x_3^{140} \end{cases}$$

Reordenando estas variables tenemos $(x_1x_2^2x_3^{58})^{\sigma_1}(x_1^3x_3^7)^{\sigma_2}(x_2)^{\sigma_3}(x_1^5)^{\sigma_4}(x_3)^{\sigma_5}$ y así definimos la siguiente aplicación $\phi : \mathbb{Q}[y_1, y_2, y_3, y_4, y_5] \longrightarrow \mathbb{K}[x_1, x_2, x_3]$ dada por

$$y_1 \longmapsto x_1x_2^2x_3^{58} \qquad y_3 \longmapsto x_2 \qquad y_5 \longmapsto x_3$$
$$y_2 \longmapsto x_1^3x_3^7 \qquad y_4 \longmapsto x_1^5$$

Definimos el ideal $I = \langle y_1 - x_1x_2^2x_3^{58}, y_2 - x_1^3x_3^7, y_3 - x_2, y_4 - x_1^5, y_5 - x_3 \rangle$ y calculamos su base de Gröbner reducida. Tomamos el orden lexicográfico de la siguiente forma $x_1 > x_2 > x_3 > y_5 > y_1 > y_2 > y_3 > y_4$, este orden personalizado de las variables y_j se debe a que queremos minimizar la variable t que equivale a y_5. Usando Sagemath obtenemos una base de Gröbner reducida formada por 30 polinomios. A continuación dividimos el monomio $x_1^4x_2^3x_3^{140}$ entre los elementos de esta base y obtenemos como resultado $y_5^{75}y_1y_2y_3$. Por tanto, la solución del sistema viene dada por $\sigma = (1, 1, 1, 0, 75)$, entonces $\sigma = (1, 1, 1, 0)$ es solución de $A\sigma = b$ y $c \cdot \sigma = 75$, así σ es la solución óptima.

Capítulo 3

Bases de Graver

En el capítulo anterior hemos visto la utilidad de las bases de Gröbner en problemas de programación lineal entera con coeficientes no negativos. Pero a veces, hay problemas en los que los coeficientes son negativos o que no tienen funciones costo lineales pero sí restricciones lineales, para resolverlos se pueden utilizar las bases de Graver.

3.1. Introducción

Comenzamos recordando la notación y algunos resultados vistos en la Sección 2.4. Vamos a considerar una matriz $A \in \mathbb{Z}^{n\times m}$ y las siguientes aplicaciones

$$\pi : \mathbb{Z}^m \longrightarrow \mathbb{Z}^n \quad \text{dada por} \quad \pi(u) = A \cdot u^T,$$

$$\phi : \mathbb{Z}[y_1, \ldots, y_m] \longrightarrow \mathbb{Z}[x_1^{\pm 1}, \ldots, x_n^{\pm 1}] \quad \text{con} \quad \phi(y^u) = x^{\pi(u)}.$$

Como antes, el núcleo de ϕ es el ideal tórico I_A y además, en el Teorema 2.22 hemos visto que este ideal está formado por la siguiente familia generadora, donde $\mathcal{B}$ una base de $ker(\pi)$

$$I_A = \langle y^v - y^v : v \in \mathcal{B} \rangle.$$

Definición 3.1. Sea $I \trianglelefteq \mathbb{Z}[y_1, \ldots, y_m]$ un ideal. Se llama *base de Gröbner universal* $\mathcal{U}$ a la unión de todas las bases de Gröbner reducidas $\mathcal{G}_>$ del ideal I con respecto a todos los órdenes monomiales. Es decir, $\mathcal{U}$ es una base de Gröbner de I con respecto a cualquier orden monomial.

Denotaremos por $\mathcal{U}_A$ a la base de Gröbner universal del ideal tórico $I_A = ker(\phi)$.

Definición 3.2. Un binomio $y^{u^+} - y^{u^-}$ en I_A se denomina *circuito* si el soporte de u es minimal con respecto a la inclusión entre todos los $u \in ker(\pi)$, y las coordenadas de u son relativamente primas. El conjunto de circuitos de I_A se denota $\mathcal{C}_A$.

Definición 3.3. Un binomio $y^{u^+} - y^{u^-}$ en I_A se llama *primitivo* si no existe otro binomio $y^{v^+} - y^{v^-}$ en I_A tal que y^{v^+} divide a y^{u^+} y y^{v^-} divide a y^{u^-}. El conjunto de todos los binomios primitivos de I_A se llama *base de Graver de* A y se denota Gr_A.

Proposición 3.4. *Todo binomio $y^{u^+} - y^{u^-}$ que pertenece a la base universal $\mathcal{U}_A$ es primitivo.*

Demostración. Sea $>$ cierto orden monomial y $\mathcal{G}_>$ la base de Gröbner reducida correspondiente, queremos probar que $\mathcal{G}_> \subseteq Gr_A$. Sea $p = y^{u^+} - y^{u^-} \in \mathcal{G}_>$ con $y^{u^+} > y^{u^-}$, veamos que es primitivo. Suponemos que no lo es, entonces existirá $v \neq u \in ker(\pi)$ tal que y^{v^+} divide a y^{u^+} y y^{v^-} divide a y^{u^-}, sea $q = y^{v^+} - y^{v^-}$.
Suponemos primero $y^{v^+} > y^{v^-}$, es decir $LT(q) = y^{v^+}$. Tenemos entonces que

$$y^{v^+} \in \langle LT(I_A) \rangle = \langle LT(\mathcal{G}_>) \rangle. \tag{3.1}$$

Además, $y^{u^+} \notin LT(\mathcal{G}_> - p)$ por ser $\mathcal{G}_>$ reducida y y^{v^+} divide a y^{u^+} luego

$$y^{v^+} \notin \langle LT(\mathcal{G}_> - p)\rangle. \tag{3.2}$$

Juntando (3.1) y (3.2) se deduce que y^{u^+} divide a y^{v^+} por tanto, $u^+ = v^+$.
Tenemos $u^+ = v^+, u^- = v^- + \beta$ con $\beta \neq 0$ y entradas no negativas. Además,$\pi(v^+) = \pi(u^+) = \pi(u^-) = \pi(v^-)+\pi(\beta) = \pi(v^+)+\pi(\beta)$. Luego, $\beta \in ker(\pi)$, así que $y^\beta - 1 \in I_A$ y $LT(y^\beta - 1) = y^\beta$, esto implica que

$$y^\beta \in \langle LT(\mathcal{G}_>)\rangle. \tag{3.3}$$

También, y^β divide a y^{u^-} y $y^{u^-} \notin \langle LT(\mathcal{G}_> - p)\rangle$, entonces

$$y^\beta \notin \langle LT(\mathcal{G}_> - p)\rangle. \tag{3.4}$$

Como antes, (3.3) y (3.4) implican que y^{u^+} divide a y^β por tanto, $supp(u^+) \subseteq supp(\beta) \subseteq supp(u^-)$ y esto es una contradicción porque $supp(u^+)$ y $supp(u^-)$ son disjuntos y $supp(u^+) \neq 0$. Queda probado que $\mathcal{G}_> \subseteq Gr_A$.

Ahora suponemos que $y^{v^-} > y^{v^+}$, es decir, $LT(q) = y^{v^-}$. Procedemos igual que en la primera parte. Tenemos que $y^{v^-} \in \langle LT(\mathcal{G}_>)\rangle$ y como $y^{u^-} \notin \langle LT(\mathcal{G}_> - p)\rangle$ y también y^{v^-} divide a y^{u^-} tenemos que $y^{v^-} \notin \langle LT(\mathcal{G}_> - p)\rangle$. Por el mismo razonamiento que antes, se deduce que y^{u^+} divide a y^{v^-}, es decir, $supp(u^+) \subseteq supp(v^-) \subseteq supp(u^-)$, obtenemos de nuevo una contradicción. □

Proposición 3.5. *Para toda matriz $A \in \mathbb{Z}^{n\times m}$ tenemos $\mathcal{C}_A \subseteq \mathcal{U}_A$.*

Demostración. Tenemos que ver que todo circuito de I_A está en alguna base de Gröbner reducida. Sea $y^{u^+} - y^{u^-}$ un circuito en I_A y fijamos un orden monomial $>$ tal que $\{y_i : i \notin supp(u)\} > \{y_j : j \in supp(u)\}$ y $y^{u^+} > y^{u^-}$. Asumimos que $y^{u^+} - y^{u^-}$ no está en la base de Gröbner reducida $\mathcal{G}_>$ de I_A y llegaremos a una contradicción.
Sabemos que $LT(y^{u^+} - y^{u^-}) = y^{u^+}$ luego $y^{u^+} \in LT(\mathcal{G}_>)$, así que existe $y^{v^+} - y^{v^-} \in \mathcal{G}_>$ con $y^{v^+} > y^{v^-}$ tal que y^{v^+} divide a y^{u^+} y como $y^{u^+} - y^{u^-} \notin \mathcal{G}_>$ entonces $u \neq v$. Luego como y^{v^+} divide a y^{u^+} tenemos que $supp(v^+) \subseteq supp(u^+) \subseteq supp(u) = supp(u^+) \cup supp(u^-)$. Por la elección del orden monomial y teniendo en cuenta que $y^{v^+} > y^{v^-}$ esto implica que $supp(v^-) \subseteq supp(u^+) \subseteq supp(u)$. Por tanto, $supp(v) \subseteq supp(u)$. Como u es un circuito, su soporte es minimal, luego esto solo es posible si $u = v$, lo cual es una contradicción. Por tanto, tenemos que $y^{u^+} - y^{u^-}$ está en $\mathcal{G}_>$. □

Por los dos resultados anteriores tenemos $\mathcal{C}_A \subseteq \mathcal{U}_A \subseteq Gr_A$.

Ejemplo 3.6. Dada una matriz $A = \begin{pmatrix} 1 & 2 & 4 \end{pmatrix} \in \mathbb{Z}^{1\times 3}$, vamos a calcular $\mathcal{C}_A$,$\mathcal{U}_A$ y Gr_A. Para ello, vamos a buscar los vectores $u = (u_1, u_2, u_3) \in \mathbb{Z}^{3\times 1}$ tales que $u \in ker(\pi)$, luego $u_1 + 2u_2 + 4u_3 = 0$. Por tanto, $u_1 = -2u_2 - 4u_3$ y se deduce que $\{(2, -1, 0), (4, 0, -1)\}$ es una base de $ker(\pi)$.
El conjunto de circuitos viene dado por aquellos que tengan soporte minimal por tanto,

$$\mathcal{C}_A = \{y_1^2 - y_2, \quad y_2^2 - y_3, \quad y_1^4 - y_3\}.$$

Estos polinomios son primitivos y se puede comprobar que $y_1^2y_2 - y_3$ también lo es y que no hay más binomios primitivos, así se tiene

$$Gr_A = \mathcal{C}_A \cup \{y_1^2y_2 - y_3\}.$$

En este caso $\mathcal{U}_A = \mathcal{C}_A$, esto se debe a que el binomio primitivo $\{y_1^2y_2 - y_3\}$ no está en ninguna base de Gröbner reducida ya que si $p = y_1^2y_2 - y_3 \in \mathcal{G}_\mathcal{A}$ base de Gröbner reducida para algún orden, $\langle y_1, y_2, y_3\rangle = \langle LT(\mathcal{C}_A)\rangle \subseteq \langle LT(\mathcal{G}_\mathcal{A})\rangle$ luego $y_2 \in \langle LT(\mathcal{G}_A)\rangle$ y teniendo en cuenta la forma de p, esto implica $y_2 \in \langle LT(\mathcal{G}_A - p)\rangle$ así que $y_1^2y_2 \in \langle LT(\mathcal{G}_A - p)\rangle$ y esto es una contradicción.

3.2. Relación entre las bases de Gröbner y las bases de Graver

En esta sección vemos la conexión entre las bases de Gröbner y las bases de Graver. El resultado más importante es el Teorema 3.10 que proporciona un algoritmo para calcular bases de Graver de una matriz $A \in \mathbb{Z}^{n\times m}$. Denotaremos $ker(A) = \{u \in \mathbb{Z}^m : a \cdot u^T = 0\}$, es decir $ker(A) = ker(\pi)$ con π definido como en las secciones anteriores.

Definición 3.7. Se llama *extensión de Lawrence* de A a la matriz Λ de dimensiones $(n+m)\times 2m$ dada por

$$\Lambda(A) = \begin{pmatrix} A & \mathbf{0} \\ I & I \end{pmatrix}$$

donde I es la matriz identidad $n \times n$ y $\mathbf{0}$ es una matriz $d \times n$ con todas sus entradas 0. Cualquier matriz de este tipo se dice de *Lawrence*.

El núcleo de la matriz $\Lambda(A)$ se puede calcular como sigue, dados $u, v \in \mathbb{Z}^m$ se tiene

$$\begin{pmatrix} A & \mathbf{0} \\ I & I \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} Au \\ u+v \end{pmatrix}$$

Por tanto,

$$\begin{pmatrix} Au \\ u+v \end{pmatrix} \in ker(\Lambda(A)) \text{ si y solo si } \begin{cases} u \in ker(A) \\ v = -u. \end{cases}$$

Luego, $ker(\Lambda(A)) = \{(u, -u) : u \in ker(A)\}$.

Definición 3.8. El *ideal tórico* $I_{\Lambda(A)}$ es el ideal

$$I_{\Lambda(A)} = \langle y^{u^+}x^{u^-} - y^{u^-}x^{u^+} : u \in ker(A)\rangle \subset \mathbb{K}[y_1, \ldots, y_m, x_1, \ldots, x_n].$$

Claramente, $(u, -u)^+ = (u^+, u^-)$ y $(u, -u)^- = (u^-, u^+)$, así se explican los exponentes de los generadores de $I_{\Lambda(A)}$.

Lema 3.9. *El ideal tórico $I_{\Lambda(A)}$ se puede expresar como $I_{\Lambda(A)} = \langle y^{u^+}x^{u^-} - y^{u^-}x^{u^+} : v \in \mathcal{B}\rangle$ siendo $\mathcal{B}$ una base de $ker(A)$.*

Demostración. Por el Teorema 2.22 sabemos que $I_A = \langle y^{v^+} - y^{v^-} : v \in \mathcal{B}\rangle$ y por definición $I_{\Lambda(A)} = \langle y^{u^+}x^{u^-} - y^{u^-}x^{u^+} : u \in ker(A)\rangle$. Luego, $u \in ker(A)$, es decir, $u = \sum_{v\in\mathcal{B}} a_v v$. Podemos definir $\mathcal{B}^+ = \{v \in \mathcal{B} : a_v \geq 0\}$ y $\mathcal{B}^- = \{v \in \mathcal{B} : a_v < 0\}$ y así podemos separar las coordenadas de u como sigue

$$u^+ = \sum_{v\in\mathcal{B}^+} a_v v^+ + \sum_{v\in\mathcal{B}^-} a_v v^- \qquad u^- = \sum_{v\in\mathcal{B}^+} a_v v^- + \sum_{v\in\mathcal{B}^-} a_v v^+$$

Entonces, $y^{u^+}x^{u^-} - y^{u^-}x^{u^+} \in \langle y^{u^+}x^{u^-} - y^{u^-}x^{u^+} : v \in \mathcal{B}\rangle$.

□

Teorema 3.10. *Para una matriz de Lawrence $\Lambda(A)$ los siguientes conjuntos de binomios coinciden*

i) La base de Graver de $\Lambda(A)$.

ii) La base de Gröbner universal de $\Lambda(A)$.

iii) Cualquier base de Gröbner reducida de $I_{\Lambda(A)}$.

Demostración. El binomio $y^{u^+} - y^{u^-}$ en I_A es primitivo si y solo si el binomio $y^{u^+}x^{u^-} - y^{u^-}x^{u^+}$ en $I_{\Lambda(A)}$ es primitivo. Entonces, la base de Graver de $\Lambda(A)$ y la base de Graver de A están relacionadas como sigue

$$Gr_{\Lambda(A)} = \{y^{u^+}x^{u^-} - y^{u^-}x^{u^+} : \quad y^{u^+} - y^{u^-} \in Gr_A\}.$$

Sabemos que para cada orden monomial $>$ si $\mathcal{G}_>$ es la base reducida asociada, $\mathcal{G}_>$ es base de Gröbner y por tanto genera el ideal $I_{\Lambda(A)}$. Como $\mathcal{G}_> \subseteq \mathcal{U}_{\Lambda(A)} \subseteq Gr_{\Lambda(A)}$, tenemos que $Gr_{\Lambda(A)}$ es un conjunto generador de $I_{\Lambda(A)}$.
Ahora veamos que $Gr_{\Lambda(A)}$ es un conjunto generador mínimo de $I_{\Lambda(A)}$ (salvo múltiplos escalares), es decir, vamos a probar que si eliminamos cualquier elemento de $Gr_{\Lambda(A)}$ entonces ese conjunto ya no genera $I_{\Lambda(A)}$.
Sea $g := y^{u^+}x^{u^-} - y^{u^-}x^{u^+}$ un elemento en $Gr_{\Lambda(A)}$ y sea $B = Gr_{\Lambda(A)} \setminus \{g\}$. Suponemos que B genera $I_{\Lambda(A)}$ y vamos a llegar a una contradicción. Como g está en $Gr_{\Lambda(A)}$, tenemos que $g \in I_{\Lambda(A)}$ y entonces g puede escribirse como combinación lineal de elementos de B. Es decir, hay un binomio $h = y^{v^+}x^{v^-} - y^{v^-}x^{v^+} \in B$ cuyo término líder divide al de g. Este hecho implica que g no es primitivo ya que según sean los términos líderes tenemos estos casos. Suponemos $LT(h) = y^{v^+}x^{v^-}$, entonces, si $y^{v^+}x^{v^-}$ divide a $y^{u^+}x^{u^-}$, y^{v^+} divide a y^{u^+} y y^{v^-} divide a y^{u^-}, contradicción por ser $y^{u^+} - y^{u^-}$ primitivo; si en cambio, $y^{v^+}x^{v^-}$ divide a $y^{u^-}x^{u^+}$, entonces y^{v^+} divide a y^{u^-} y y^{v^-} divide a y^{u^+}, que es de nuevo una contradicción. Si suponemos $LT(h) = y^{v^-}x^{v^+}$ el razonamiento es análogo. Por tanto, hemos llegado a una contradicción. Concluimos que $Gr_{\Lambda(A)}$ es el conjunto generador mínimo de $I_{\Lambda(A)}$ y por la Proposiciones 3.4 y 3.5 tenemos que $\mathcal{G}_> = \mathcal{U}_{\Lambda(A)} = Gr_{\Lambda(A)}$. □

3.3. Cálculo de bases de Graver

El Teorema 3.10 nos proporciona el siguente algoritmo para calcular bases de Graver

1. Calcular $ker(A)$ para obtener el ideal tórico $I_{\Lambda(A)}$.

2. Fijar algún orden monomial en $\mathbb{K}[y_1, \ldots, y_m, x_1, \ldots, x_n]$ y calcular la base de Gröbner reducida $\mathcal{G}$ de $I_{\Lambda(A)}$.

3. Sustituir $x_1, \ldots, x_n \longmapsto 1$ en $\mathcal{G}$. El conjunto resultante de $\mathbb{K}[y_1, \ldots, y_m]$ es una base de Graver Gr_A de A.

   La corrección del segundo paso se debe al hecho de que

   $$Gr_{\Lambda(A)} = \{y^{u^+}x^{u^-} - y^{u^-}x^{u^+} : \quad y^{u^+} - y^{u^-} \in Gr_A\}.$$

   Entonces vemos que si sustitumos $x_i = 1$ para todo i en $Gr_{\Lambda(A)}$ obtenemos una base de Graver de A.

Por justificar el algoritmo que vamos a ver a continuación vamos a ver una definición y un resultado cuya demostración no se ha podido incluir por extensión pero la podemos encontrar en la referencia [10, p.65].

Definición 3.11. Un conjunto $\tau \subseteq \mathbb{Z}^m$ se denomina *conjunto test* de un problema de minimización como (2.1) si para cada solución factible no óptima σ_0 del problema existe un vector $t \in \tau$ y un entero no negativo α tal que $\sigma_0 + \alpha t$ es factible y $c(\sigma_0 + \alpha t) < c(\sigma_0)$.

Lema 3.12. *Dado un problema como en (2.1) y una familia finita $\Omega \in \mathbb{Z}^m$ de soluciones factibles se tiene que Gr_A es un conjunto test para Ω.*

Ahora enunciamos el algoritmo para encontrar una solución óptima del problema.

Algoritmo para buscar una solución óptima en un problema de programación lineal.
Paso inicial: Se calcula una solución factible $\sigma_0 \in \Omega$ del problema (2.1).
Paso k+1: Dada x_k solución en Ω, se busca $g \in Gr_A$ y $\lambda \in \mathbb{Z}_{\geq 0}$ tal que $x_k + \lambda g \in \Omega$ y $c \cdot g < 0$. Si no existe ninguna solución que cumpla, entonces se termina el algoritmo y x_k es la solución óptima. Si existe tomamos $x_{k+1} = x_k + \lambda g$. Repetimos el algoritmo hasta que no exista una solución que cumpla la desigualdad.

Ejemplo 3.13. Dada el problema de minimización dado por $A = \begin{pmatrix} 3 & 5 & -1 \end{pmatrix}$, $b = 17$ y $c = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$. Vamos a hallar la solución óptima del sistema usando bases de Graver.
Primero calculamos el núcleo $ker(A)$, su base es $\{\begin{pmatrix} 1 & 0 & 3 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 5 \end{pmatrix}\}$. Entonces, tenemos que el ideal tórico $I_{\Lambda(A)}$ viene dado por

$$I_{\Lambda(A)} = \langle y_1 y_3^3 - x_1 x_3^3, \quad y_2 y_3^5 - x_2 x_3^5 \rangle.$$

Fijamos el orden lexicográfico en $\mathbb{K}[y_1, y_2, y_3, x_1, x_2, x_3]$ y calculamos la base de Gröbner reducida de $I_{\Lambda(A)}$ obteniendo $\mathcal{G} = \{y_1 y_3^3 - x_1 x_3^3, \quad y_1 x_2 x_3^5 - y_2 y_3^2 x_1 y_3^3, \quad y_2 y_3^5 - x_2 x_3^5\}$. Sustituyendo $x_i = 1$ con $i = 1, 2, 3$ y obtenemos la base de Graver de A

$$Gr_A = \{y_1 y_3^3 - 1, \quad y_1 - y_2 y_3^2, \quad y_2 y_3^5 - 1\}.$$

Una vez tenemos la base de Graver necesitamos una solución factible de $3\sigma_1 + 5\sigma_2 - \sigma_3 = 17$, fácilmente se comprueba que $\sigma_0 = (2, 3, 4)$ es una solución con función costo 10. Necesitamos saber si esta es la solución que minimiza la función objetivo o hay alguna solución mejor. Para ello, consideramos σ_0 y le vamos a sumar los elementos de la base de Graver. Vemos que $\sigma_0 + (1, -1, -2) = (3, 2, 2)$ con función objetivo 7 así que mejora nuestra solución. Repetimos el proceso con $(3, 2, 2)$ y vemos que $(3, 2, 2) + (1, -1, -2) = (4, 1, 0)$ con función costo 5. Finalmente repetimos el algoritmo con $(4, 1, 0)$ y vemos que ya no se puede mejorar, por tanto, la solución ópitma es $(4, 1, 0)$.

Capítulo 4

Aplicación

Ahora vemos cómo aplicar la teoría previamente estudiada a situaciones de la vida cotidiana. En particular, utilizaremos las bases de Gröbner y de Graver para resolver el siguiente problema: optimizar la cantidad de medicación dispensada para desperdiciar lo mínimo posible.

4.1. Contexto

La Receta Electrónica es un servicio digital que permite al profesional la prescripción de medicamentos que posteriormente pueden ser dispensados en las farmacias sin necesidad de portar una receta física.

Este sistema ofrece numerosas ventajas como por ejemplo mayor agilidad y reducción de errores en la prescripción y dispensación, control del gasto, planificación sanitaria y mejor servicio al ciudadano. Sin embargo, los avances médicos y tecnológicos requieren que estos sistemas evolucionen para seguir ofreciendo una asistencia de calidad. Entre estos avances, surge la necesidad de controlar el uso de los medicamentos para que se haga de manera racional; y una de las formas de conseguirlo es optimizando la medicación que se le dispensa al paciente.

Esto significa, que cuando un profesional sanitario prescribe "Ibuprofeno 600 mg sobres", se pueden optimizar las recetas que se generan para el paciente, para que cumpla todo su tratamiento y desperdicie la menor cantidad de medicación. Es decir, vamos a optimizar la secuencia de envases que se le da al paciente.

Previamente a optimizar esta secuencia, debemos saber el contenido que tienen los envases de un medicamento, información que está disponible en el Ministerio de Sanidad [1]. Por ejemplo, podemos tener envases de ibuprofeno de 10, 20 o 28 sobres. Por tanto, si queremos saber la secuencia óptima de envases, deberemos conocer todos los posibles tamaños de envases que hay.

4.2. Planteamiento general

Nuestro objetivo es ver cómo optimizar la cantidad de medicamento para que sobre el menor excedente posible. Para ello, vamos a plantear problemas de programación lineal entera de minimización, los modelaremos con el siguiente planteamiento y luego los resolveremos con la teoría vista. Se proporcionan prescripciones médicas de casos reales que tienen la siguiente forma.

Un paciente ha ido al médico y supondremos que le ha prescrito un medicamento que tiene unas ciertas características (su forma: sobres, comprimidos, solución...; si caduca; si se puede reaprovechar...). Debemos calcular la dosis total que es necesaria para cumplir el tratamiento,

[1] La información está disponible en la web: `https://www.sanidad.gob.es/profesionales/nomenclator.do`. Esa información se actualiza mensualmente y se envía a las Comunidades Autónomas para que actualicen sus bases de datos.

pongamos X y por otro lado, tenemos que saber que ese medicamento se presenta, por ejemplo, en envases de a_1 comprimidos o de a_2 comprimidos.

Por tanto, tenemos que encontrar la solución óptima, es decir, tenemos que encontrar la mejor combinación de estos envases de medicamentos para que sobre la menor cantidad posible y el paciente pueda tomarse los X comprimidos. Para ello, planteamos este problema como un problema de programación lineal entera donde vamos a minimizar la cantidad de medicación sobrante.

Vamos a suponer que el paciente finalmente toma

- b_1 comprimidos de envases de a_1 comprimidos.
- b_2 comprimidos de envases de a_2 comprimidos.

Entonces, definimos las siguientes ecuaciones

$$b_1 = a_1 z_1 + r_1 \qquad\qquad b_2 = a_2 z_2 + r_2$$

donde las nuevas variables definidas son

- z_1 =número de envases de a_1 comprimidos que toma completos.
- z_2 =número de envases de a_2 comprimidos que toma completos.
- r_1 =número de comprimidos que toma de un envase de a_1 comprimidos sin terminar.
- r_2 =número de comprimidos que toma de un envase de a_2 comprimidos sin terminar.

Por tanto, como el paciente debe tomar X comprimidos, entonces

$$b_1 + b_2 = a_1 z_1 + r_1 + a_2 z_2 + r_2 = X$$

Además, por como han sido elegidas las variables tenemos las siguientes restricciones

$$\begin{array}{lll} 0 < r_1 \leq a_1 & r_1 + t_1 = a_1 & 0 \leq t_1 < a_1 \\ 0 < r_2 \leq a_2 & r_2 + t_2 = a_2 & 0 \leq t_2 < a_2 \end{array}$$

donde las nuevas variables son

- t_1 = número de comprimidos que sobran de un envase de a_1 comprimidos sin terminar.
- t_2 =número de comprimidos que sobran de un envase de a_2 comprimidos sin terminar.

Con todas estas variables definidas podemos plantear el siguiente problema de programación lineal entera de minimización

$$\begin{array}{ll} \text{mín} & t_1 + t_2 \\ \text{s.a:} & a_1 z_1 + a_2 z_2 + r_1 + r_2 = X \\ & r_1 + t_1 = a_1 \\ & r_2 + t_2 = a_2 \end{array} \tag{4.1}$$

Pero con este planteamiento forzamos que se usen envases de los dos tipos ya que imponemos $r_i + t_i \neq 0$. Para considerar el caso en el que solo se use un tipo de envase, vamos a añadir unas nuevas variables c_i y c_i' . Además, para poder usar el orden lexicográfico y minimizar así una

variable, vamos a renombrar algunas variables como $t = t_1 + t_2$, por ello añadimos la última ecuación redundante. Entonces tenemos

$$\begin{aligned} \text{mín} \quad & t \\ \text{s.a:} \quad & a_1z_1 + a_2z_2 + r_1 + r_2 = X \\ & r_1 + t_1 + a_1c_1 = a_1 \\ & c_1 + c_1' = 1 \\ & r_2 + t_2 + a_2c_2 = a_2 \\ & c_2 + c_2' = 1 \\ & r_1 + r_2 + a_1c_1 + a_2c_2 + t = a_1 + a_2 \end{aligned} \tag{4.2}$$

Una vez planteado el problema, para resolverlo vamos a usar la teoría de bases de Gröbner vista, entonces vamos a transformar el sistema formado por las restricciones en el siguiente

$$\begin{cases} a_1\sigma_1 + a_2\sigma_2 + \sigma_3 + \sigma_4 = X \\ \sigma_3 + \sigma_5 + a_1\sigma_8 = a_1 \\ \sigma_8 + \sigma_9 = 1 \\ \sigma_4 + \sigma_6 + a_2\sigma_{10} = a_2 \\ \sigma_{10} + \sigma_{11} = 1 \\ \sigma_3 + \sigma_4 + a_1\sigma_8 + a_2\sigma_{10} + \sigma_7 = a_1 + a_2 \end{cases}$$

Trabajamos en un anillo de polinomios con una variable por cada ecuación.

$$\begin{cases} x_1^{a_1\sigma_1+a_2\sigma_2+\sigma_3+\sigma_4} = x_1^X \\ x_2^{\sigma_3+\sigma_5+a_1\sigma_8} = x_2^{a_1} \\ x_3^{\sigma_8+\sigma_9} = x_3 \\ x_4^{\sigma_4+\sigma_6+a_2\sigma_{10}} = x_4^{a_2} \\ x_5^{\sigma_{10}+\sigma_{11}} = x_5 \\ x_6^{\sigma_3+\sigma_4+a_1\sigma_8+a_2\sigma_{10}+\sigma_7} = x_6^{a_1+a_2} \end{cases}$$

Reordenando

$$\begin{aligned} &(x_1^{a_1})^{\sigma_1}(x_1^{a_2})^{\sigma_2}(x_1x_2x_6)^{\sigma_3}(x_1x_4x_6)^{\sigma_4}(x_2)^{\sigma_5}(x_4)^{\sigma_6}(x_6)^{\sigma_7}(x_2^{a_1}x_3x_6^{a_1})^{\sigma_8}(x_3)^{\sigma_9}(x_4^{a_2}x_5x_6^{a_2})^{\sigma_{10}}(x_5)^{\sigma_{11}} \\ &= x_1^X x_2^{a_1} x_3 x_4^{a_2} x_5 x_6^{a_1+a_2} \end{aligned}$$

Entonces, la correspondiente aplicación se define como
$\phi : \mathbb{Q}[y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}] \longrightarrow \mathbb{K}[x_1, x_2, x_3, x_4, x_5, x_6]$ con

$$\begin{array}{llll} y_1 \longmapsto x_1^{a_1} & y_4 \longmapsto x_1x_4x_6 & y_7 \longmapsto x_6 & y_{10} \longmapsto x_4^{a_2}x_5x_6^{a_2} \\ y_2 \longmapsto x_1^{a_2} & y_5 \longmapsto x_2 & y_8 \longmapsto x_2^{a_1}x_3x_6^{a_1} & y_{11} \longmapsto x_5 \\ y_3 \longmapsto x_1x_2x_6 & y_6 \longmapsto x_4 & y_9 \longmapsto x_3 & \end{array}$$

Por tanto, consideramos el ideal:

$$\begin{aligned} I = \langle & y_1 - x_1^{a_1}, y_2 - x_1^{a_2}, y_3 - x_1x_2x_6, y_4 - x_1x_4x_6, y_5 - x_2, y_6 - x_4, y_7 - x_6, y_8 - x_2^{a_1}x_3x_6^{a_1}, \\ & y_9 - x_3, y_{10} - x_4^{a_2}x_5x_6^{a_2}, y_{11} - x_5 \rangle. \end{aligned}$$

Para aplicar el algoritmo de Buchberger vamos a tomar el orden lexicográfico de la siguiente forma: $x_1 > x_2 > x_3 > x_4 > x_5 > x_6 > y_7 > y_2 > y_5 > y_4 > y_3 > y_1 > y_8 > y_9 > y_{10} > y_{11}$. Este orden personalizado de las y_j se debe a que queremos minimizar la variable t que equivale a y_7 por lo que la queremos eliminar la primera. Luego, el resto de variables no importa cómo las ordenemos, ponemos y_2 en segundo lugar porque queremos que haya envases completos y sean de mayor tamaño mejor. De esta manera, encontraremos la solución óptima.

Una vez tenemos el ideal y el orden monomial que utilizaremos calculamos la base de Gröbner.

4.3. Resolución de casos reales

Una vez visto el planteamiento a seguir, partimos de casos reales y vamos a modelizar la situación a un problema matemático. A continuación, iremos resolviéndolos conforme a la teoría vista en bases de Gröbner y con la ayuda de Sage calcularemos estas bases.

4.3.1. Ejemplo 1

Un paciente tiene dolor e inflamación en la rodilla debido al desgaste del cartílago articular. El médico le prescribe Ibuprofeno 600 mg en comprimidos, cada 8 horas, durante 45 días, para disminuir el dolor crónico y reducir la inflamación.

En el momento que el médico prescribe esa pauta en el sistema, se debe calcular la dosis diaria, 3 comprimidos; la dosis total, 135 comprimidos; y por último, nos debe indicar el sistema que para el ibuprofeno de 600 mg hay envases que contienen 30 comprimidos y otros que tienen 40 comprimidos.

Por tanto, queremos que el paciente se tome los 135 comprimidos y sobre lo menor posible usando envases de 30 y 40 comprimidos. Procedemos usando el planteamiento propuesto. En este caso, quedaría definido por el siguiente problema de minimización

$$\begin{array}{ll} \text{mín} & t \\ \text{s.a:} & 40z_1 + 30z_2 + r_1 + r_2 = 135 \\ & r_1 + t_1 + 40c_1 = 40 \\ & c_1 + c_1' = 1 \\ & r_2 + t_2 + 30c_2 = 30 \\ & c_2 + c_2' = 1 \\ & r_1 + r_2 + 40c_1 + 30c_2 + t = 70 \end{array} \tag{4.3}$$

donde las variables con subíndice 1 hacen referencias a los envases de 40 comprimidos y las de subíndice 2 a los de 30 comprimidos.

Ahora debemos transformar el sistema formado por las restricciones para aplicar la teoría vista. Se tiene

$$\begin{cases} 40\sigma_1 + 30\sigma_2 + \sigma_3 + \sigma_4 = 135 \\ \sigma_3 + \sigma_5 + 40\sigma_8 = 40 \\ \sigma_8 + \sigma_9 = 1 \\ \sigma_4 + \sigma_6 + 30\sigma_{10} = 30 \\ \sigma_{10} + \sigma_{11} = 1 \\ \sigma_3 + \sigma_4 + 40\sigma_8 + 30\sigma_{10} + \sigma_7 = 70 \end{cases}$$

Por tanto, introduciendo las varibles necesarias y definiendo la función ϕ se obtiene el siguiente ideal

$$\begin{aligned} I = \langle & y_1 - x_1^{40}, y_2 - x_1^{30}, y_3 - x_1x_2x_6, y_4 - x_1x_4x_6, y_5 - x_2, y_6 - x_4, y_7 - x_6, y_8 - x_2^{40}x_3x_6^{40}, \\ & y_9 - x_3, y_{10} - x_4^{40}x_5x_6^{30}, y_{11} - x_5 \rangle. \end{aligned}$$

Ver su implementación en el Apéndice A.1.
Tomando el orden lexicográfico definido anteriormente, obtenemos una base de Gröbner de 1545 polinomios. A continuación, reducimos el monomio $x_1^{135}x_2^{40}x_3x_4^{30}x_5x_6^{70}$ dividiéndolo por los polinomios de la base G y obtenemos como resultado $y_7^5y_2y_5^5y_4^{30}y_3^{35}y_1y_9y_{11}$.

Por tanto, $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7, \sigma_8, \sigma_9, \sigma_{10}, \sigma_{11})$ es solución del sistema (4.3), además es la solución que minimiza la función costo.

Así, tenemos que combinando envases de 40 y 30 comprimidos una solución óptima es coger 2 envases de 40 comprimidos y 2 envases de 30 comprimidos, de esta manera sobrarían solo 5 pastillas de un envase de 40.

4.3.2. Ejemplo 2

A una paciente le han diagnosticado edema. El médico le prescribe Furosemida 40 mg comprimidos con una ingesta de un comprimido diario. Deberá seguir esta pauta durante 3 meses y los siguientes 28 días reducirá la ingesta a medio comprimido diario.

En el momento que el médico prescribe esa pauta en el sistema, se debe calcular la dosis diaria, 1 comprimido durante tres meses y medio comprimido durante 28 días; la dosis total, 104 comprimidos; y por último, nos debe indicar el sistema que para la Furosemida 40 mg hay envases que contienen 21 comprimidos y otros que tienen 30 comprimidos.

Procedemos de la misma manera que en el ejemplo anterior y buscamos la solución óptima para que sobre la menor cantidad posible. Planteamos el problema siguiendo los mismos pasos, tomamos como $a_1 = 30$ y como $a_2 = 21$, es decir, las variables con subíndice 1 hacen referencia a los envases de 30 comprimidos y las de subíndice 2 a los de 21 comprimidos. Con el mismo razonamiento, llegamos al siguiente ideal

$$\begin{aligned} I =&\langle y_1 - x_1^{30}, y_2 - x_1^{21}, y_3 - x_1x_2x_6, y_4 - x_1x_4x_6, y_5 - x_2, y_6 - x_4, y_7 - x_6, y_8 - x_2^{30}x_3x_6^{30}, \\ & y_9 - x_3, y_{10} - x_4^{21}x_5x_6^{21}, y_{11} - x_5\rangle. \end{aligned}$$

Tomamos de nuevo el orden lexicográfico definido y se obtiene una base de Gröbner de 2917 polinomios. A continuación, reducimos el monomio $x_1^{104}x_2^{30}x_3x_4^{21}x_5x_6^{51}$ dividiéndolo por los polinomios de la base G y obtenemos como resultado $y_7y_2^4y_6y_4^{20}y_8y_{11}$. Consultar Apéndice A.2.

Por tanto, $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7, \sigma_8, \sigma_9, \sigma_{10}, \sigma_{11})$ es solución del sistema (4.2), además es la solución que minimiza la función costo.

Así, tenemos que combinando envases de 30 y 21 comprimidos lo óptimo es coger 5 envases de 21 comprimidos y así solo sobra 1 comprimido.

4.3.3. Ejemplo 3

Un niño tiene otitis aguda y el médico le ha prescrito Amoxicilina/Clavulanico (100/12,5) mg/1ml suspensión oral. Debe tomar 4 ml cada 12 horas durante 4 semanas.

El paciente debe tomar diariamente 8 ml, entonces para cumplir la dosis necesita 224 ml. El sistema comprueba que este medicamento tiene envases de 40 ml y de 60 ml. Además, estos envases caducan a los 7 días, por tanto, del envase de 60 ml solo podrán usarse 56 ml.

Este ejemplo va a ser resuelto usando bases de Graver por lo que el planteamiento general vamos a simplificarlo mucho ya que podemos usar coeficientes negativos en las restricciones. Ver Apéndice A.3.
En este caso, z_i va a ser el número de envases, ya se consuman enteros o no; la variable t serán los mililitros sobrantes totales; y por último, la variable a minimizar será $4z_1+t$ ya que de los envases de 60 ml por caducidad siempre sobran 4 ml. Tenemos entonces el siguiente planteamiento

$$\begin{array}{ll} \text{mín} & 4z_1 + t \\ \text{s.a:} & 56z_1 + 40z_2 - t = 224 \end{array} \tag{4.4}$$

Procedemos tomando la matriz $A = \begin{pmatrix} 56 & 40 & -1 \end{pmatrix}$ y calculamos su núcleo $ker(A)$ para después hallar el ideal tórico $I_{\Lambda(A)}$. La base de $ker(A)$ es $\{\begin{pmatrix} 1 & 0 & 56 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 40 \end{pmatrix}\}$. Por tanto,

se obtiene el siguiente ideal

$$I_{\Lambda(A)} = \langle y_1 y_3^{56} - x_1 x_3^{56}, y_2 y_3^{40} - x_2 x_3^{40} \rangle.$$

Fijamos el orden lexicográfico en $\mathbb{K}[y_1, y_2, y_3, x_1, x_2, x_3]$ y calculamos la base de Gröbner reducida de $I_{\Lambda(A)}$ obteniendo

$$\mathcal{G} = \{y_1 y_3^{56} - x_1 x_3^{56}, y_1 y_3^{16} x_2 x_3^{40} - y_2 x_1 x_3^{56}, y_1 x_2^2 x_3^{80} - y_2^2 y_3^{24} x_1 x_3^{56}, y_2 y_3^{40} - x_2 x_3^{40}\}.$$

A continuación, sustituimos $x_i = 1$ con $i = 1, 2, 3$ y así tenemos la base de Graver de A

$$Gr_A = \{y_1 y_3^{56} - 1, y_1 y_3^{16} - y_2, y_1 - y_2^2 y_3^{24}, y_2 y_3^{40} - 1\}.$$

Una vez tenemos la base de Graver necesitamos una solución factible del sistema (4.4), fácilmente se comprueba que $\sigma_0 = \begin{pmatrix} 3 & 2 & 24 \end{pmatrix}$ es una solución con 36 ml sobrantes. Necesitamos saber si esta es la solución que minimiza los mililitros sobrantes o hay alguna solución mejor. Para ello, a esta solución le vamos a ir sumando elementos de la base de Graver y comprobando si $4z_1 + t$ es menor que 36 y se trata de una solución factible. Es decir, consideramos σ_0 y le vamos sumando los elementos $g \in Gr_A$. Al comprobarlo vemos que $\sigma_0 + \begin{pmatrix} 1 & -2 & -24 \end{pmatrix} = \begin{pmatrix} 4 & 0 & 0 \end{pmatrix}$ con función objetivo 16. Repetimos el proceso con $\begin{pmatrix} 4 & 0 & 0 \end{pmatrix}$ y vemos que ya no se puede mejorar. Entonces, la solución óptima es comprar 4 envases de 60 ml y así sobran solo 16 ml.

4.4. Conclusión

Las Bases de Gröbner y las Bases de Graver pueden emplearse para resolver sistemas algebraicos y optimizar problemas relacionados con la prescripción de medicamentos. Estas herramientas han sido efectivas para simplificar sistemas complejos y encontrar soluciones óptimas bajo restricciones específicas.

Mientras que las bases de Gröbner resuelven sistemas de ecuaciones lineales, las bases de Graver son capaces de resolver sistemas lineales y sistemas no lineales, así como sistemas lineales con coeficientes negativos. Por ello, estas bases han simplificado mucho más el desarrollo de la práctica, siendo mucho más eficientes que las bases de Gröbner en las que hemos tenido que añadir muchas variables para que fuera posible la resolución del sistema. SageMath ha sido un recurso clave para aplicar estas técnicas en la práctica, ya que al trabajar con valores reales las bases eran muy extensas. No obstante, en un sistema digital el código se debería implementar con otras tecnologías capaces de optimizar el rendimiento del algoritmo.

La creación de un algoritmo que diese solución a los ejemplos anteriormente descritos ha sido una necesidad real de uno de los sistemas de Receta Electrónica. La importancia de optimizar la medicación dispensada beneficia indirectamente al uso racional de los medicamentos. Precisamente, minimizar el sobrante de medicación evita en muchos casos la automedicación. Erróneamente cuando un paciente percibe síntomas similares, si tiene medicación en casa, la consume sin prescripción médica, lo cual puede derivar posteriormente en enfermedades relacionadas con la medicación, como la resistencia a los medicamentos [11]. Además, también beneficia al gasto económico puesto que muchos de los medicamentos están subvencionados por el Sistema Nacional de Salud, así que dispensar únicamente lo necesario, genera ahorros para el sistema y para el paciente.

Por ello, podemos concluir que las Bases de Gröbner y Graver son muy útiles para dar una solución a determinados problemas de la vida real, y un ejemplo de cómo de importantes son las matemáticas en nuestra actualidad.

Apéndice A

Código Sage

A.1. Ejemplo 1

Paciente que debe tomar 135 comprimidos y se dispensan en envases de 30 o 40 comprimidos.

Comenzamos definiendo el anillo de polinomios.

```
R = PolynomialRing(RationalField(),17,['x1','x2','x3','x4','x5','x6','y7
    ','y2','y6','y5','y4','y3','y1','y8','y9','y10','y11'],order='lex')
x1,x2,x3,x4,x5,x6,y7,y2,y6,y5,y4,y3,y1,y8,y9,y10,y11= R.gens()
```

Definimos el ideal I generado por los polinomios de la función ϕ.

```
xf = y1-x1^(40)
xg = y2-x1^(30)
xh= y3-x1*x2*x6
xk= y4-x1*x4*x6
xj=y5-x2
xi=y6-x4
xl=y7-x6
xm=y8-x2^(40)*x3*x6^(40)
xn=y9-x3
xo=y10-x4^(30)*x5*x6^(30)
xp=y11-x5
I = (xf, xg,xh,xk,xj,xi,xl,xm,xn,xo,xp)*R
I
```

```
Ideal (-x1^40 + y1, -x1^30 + y2, -x1*x2*x6 + y3, -x1*x4*x6 + y4, -x2 + y5, -x4 + y6,
-x6 + y7, -x2^40*x3*x6^40 + y8, -x3 + y9, -x4^30*x5*x6^30 + y10, -x5 + y11) of
Multivariate Polynomial Ring in x1, x2, x3, x4, x5, x6, y7, y2, y6, y5, y4, y3, y1, y8,
y9, y10, y11 over Rational Field
```

Calculamos la base de Gröbner del ideal I.

```
gb = I.groebner_basis()
gb
```

```
Polynomial Sequence with 1545 Polynomials in 17 Variables.
```

Definimos dos funciones de división de polinomios. La primera de ellas realiza la división euclídea entre dos polinomios xf y xg, devolviendo el cociente y el resto. La segunda de ellas divide un polinomio entre una lista de polinomios usando la función anterior.

```
def division(xf, xg):
    if xg == 0:
        return("NaN")
```

```
    else:
        a = 0
        while xf.lt() / xg.lt() in R and xf.lt() / xg.lt() != 0:
            a = a + R(xf.lt() / xg.lt())
            xf = xf - R(xf.lt() / xg.lt()) * xg
    return (a, xf)
```

```
def multidivision(xf, listaf):
    s=[]
    resto=R(xf)
    for i in listaf:
        s.append(division(R(resto),R(i))[0])
        resto=division(R(resto),R(i))[1]
    i=0
    while i<len(listaf) and resto !=0:
        a= division(R(R(resto).lm()), listaf[i].lm())[0]
        r= division(R(R(resto).lm()), listaf[i].lm())[1]
        if r==0:
            a=division(R(resto),listaf[i])[0]
            r= division(R(resto), listaf[i])[1]
            s[i]=s[i]+a
            resto=R(resto-a*listaf[i])
            i=0
        else:
            i+=1
    return s,resto
```

Una vez tenemos la base de Gröbner, solo nos queda dividir el monomio $x_1^{135}x_2^{40}x_3^{30}x_4^{70}$ entre los polinomios de la base de Gröbner para encontrar una solución óptima del sistema.

```
show(multidivision(x1^(135)*x2^(40)*x3*x4^(30)*x5*x6^(70),gb)[1])
```

$y_7^5y_2y_5^5y_4^{30}y_3^{35}y_1y_9y_{11}$

Vemos que se llega a un resto que solo depende de y_i por tanto, hemos encontrado una solución del sistema que buscábamos. Interpretando la solución con las variables que hemos definido anteriormente vemos que se cogerían 2 envases de 40 comprimidos y 2 envases de 30 comprimidos, así sobrarían 5 comprimidos de un envase de 40.

A.2. Ejemplo 2

Paciente que tiene que tomar 104 comprimidos y se dispensan en envases de 30 o 21 comprimidos.

Comenzamos definiendo el anillo de polinomios.

```
R = PolynomialRing(RationalField(),17,['x1','x2','x3','x4','x5','x6','y7
    ','y2','y6','y5','y4','y3','y1','y8','y9','y10','y11'],order='lex')
x1,x2,x3,x4,x5,x6,y7,y2,y6,y5,y4,y3,y1,y8,y9,y10,y11= R.gens()
```

Definimos el ideal I generado por los polinomios de la función ϕ.

```
xf = y1-x1^(30)
xg = y2-x1^(21)
xh= y3-x1*x2*x6
xk= y4-x1*x4*x6
xj=y5-x2
xi=y6-x4
xl=y7-x6
```

```
xm=y8-x2^(30)*x3*x6^(30)
xn=y9-x3
xo=y10-x4^(21)*x5*x6^(21)
xp=y11-x5
I = (xf, xg, xh, xk, xj, xi, xl, xm, xn, xo, xp)*R
I
```

```
Ideal (-x1^30 + y1, -x1^21 + y2, -x1*x2*x6 + y3, -x1*x4*x6 + y4, -x2 + y5, -x4 + y6,
-x6 + y7, -x2^30*x3*x6^30 + y8, -x3 + y9, -x4^21*x5*x6^21 + y10, -x5 + y11) of
Multivariate Polynomial Ring in x1, x2, x3, x4, x5, x6, y7, y2, y6, y5, y4, y3, y1, y8,
y9, y10, y11 over Rational Field
```

Calculamos la base de Gröbner del ideal I.

```
gb = I.groebner_basis()
gb
```

```
Polynomial Sequence with 2917 Polynomials in 17 Variables
```

Definimos dos funciones de división de polinomios. La primera de ellas realiza la división euclídea entre dos polinomios xf y xg, devolviendo el cociente y el resto. La segunda de ellas divide un polinomio entre una lista de polinomios usando la función anterior.

```
def division(xf, xg):
    if xg == 0:
        return("NaN")
    else:
        a = 0
        while xf.lt() / xg.lt() in R and xf.lt() / xg.lt() != 0:
            a = a + R(xf.lt() / xg.lt())
            xf = xf - R(xf.lt() / xg.lt()) * xg
    return (a, xf)
```

```
def multidivision(xf, listaf):
    s=[]
    resto=R(xf)
    for i in listaf:
        s.append(division(R(resto),R(i))[0])
        resto=division(R(resto),R(i))[1]
    i=0
    while i<len(listaf) and resto !=0:
        a= division(R(R(resto).lm()), listaf[i].lm())[0]
        r= division(R(R(resto).lm()), listaf[i].lm())[1]
        if r==0:
            a=division(R(resto),listaf[i])[0]
            r= division(R(resto), listaf[i])[1]
            s[i]=s[i]+a
            resto=R(resto-a*listaf[i])
            i=0
        else:
            i+=1
    return s,resto
```

Dividimos el monomio $x_1^{104}x_2^{30}x_3^{21}x_4^{51}$ entre los polinomios de la base de Gröbner.

```
show(multidivision(x1^(104)*x2^(30)*x3*x4^(21)*x5*x6^(51), gb)[1])
```

$y_7y_2^4y_6y_4^{20}y_8y_{11}$

Vemos que se llega a un resto que solo depende de y_i, por tanto, hemos encontrado una solución óptima del sistema que buscábamos.

Interpretando la solución con las variables definidas, vemos que se cogerían 5 envases de 21 comprimidos, sobrando solo 1 comprimido.

A.3. Ejemplo 3

Un paciente necesita 224 ml de un medicamento que dispone de envases de 60 ml y 40 ml los cuales se caducan a los 7 días, por tanto solo puede aprovechar 56 ml del envase de 60 ml.

Comenzamos definiendo la matriz A.

```
A=Matrix([56,40,-1])
show(A)
```

$$\begin{pmatrix} 56 & 40 & -1 \end{pmatrix}$$

Ahora calculamos el núcleo de esta matriz.

```
kernel = A.right_kernel()
print(kernel)
```

```
Free module of degree 3 and rank 2 over Integer Ring
Echelon basis matrix:
[ 1  0 56]
[ 0  1 40]
```

Definimos los vectores que forman el $ker(A)$.

```
u1=vector([1,0,56])
u2=vector([0,1,40])
```

Definimos el anillo de polinomios para hallar el ideal tórico $I_{\Lambda(A)}$.

```
R = PolynomialRing(RationalField(),6,['y1','y2','y3','x1','x2','x3'],
    order='lex')
y1,y2,y3,x1,x2,x3= R.gens()
```

Calculamos los binomios que forman el ideal tórico $I_{\Lambda(A)}$ a partir de los vectores del núcleo de A.

```
binomio1=y1*y3^(56)-x1*x3^(56)
show(binomio1)
binomio2=y2*y3^(40)-x2*x3^(40)
show(binomio2)
```

$y_1y_3^{56} - x_1x_3^{56}$
$y_2y_3^{40} - x_2x_3^{56}$

Por tanto tenemos que el ideal tórico viene dado por:

```
I = R.ideal(binomio1, binomio2)
show(I)
```

$(y_1y_3^{56} - x_1x_3^{56}, y_2y_3^{40} - x_2x_3^{56})\mathbb{Q}[y_1, y_2, y_3, x_1, x_2, x_3]$

Ahora ya podemos calcular la base de Gröbner del ideal.

```
gb = I.groebner_basis()
show(gb)
```

$[y_1y_3^{56} - x_1x_3^{56}, y_1y_3^{16}x_2x_3^{40} - y_2x_1x_3^{56}, y_1x_2^2x_3^{80} - y_2^2y_3^{24}x_1x_3^{56}, y_2y_3^{40} - x_2x_3^{40}]$

Comprobamos que es una base de Gröbner reducida.

```
def division(xf, xg):
    if xg == 0:
        return("NaN")
    else:
        a = 0
        while xf.lt() / xg.lt() in R and xf.lt() / xg.lt() != 0:
            a = a + R(xf.lt() / xg.lt())
            xf = xf - R(xf.lt() / xg.lt()) * xg
    return (a, xf)
```

```
    def Reducir(xlG):
    i = 0
    while i < len(xlG):
        if R(xlG[i]).degree()== 0:
            xlG.remove(xlG[i])
        i += 1
    i=0
    #lo primero que hacemos es convertir todos los terminos lideres en
       monomios
    while i < len(xlG):
        if R(xlG[i]).lc() != 1:
            xlG[i] = xlG[i] / R(xlG[i]).lc()
        i += 1
    #ahora lo que hacemos es quitar los polinomios cuyo termino lider
       sea multiplo de algun termino lider
    j = 0
    while j < len(xlG):
        k = 0
        while k < len(xlG):
            if j != k:
                [c, r] = division(R(xlG[j]).lt(),R(xlG[k]).lt())
                if r==0:
                    xlG.remove(xlG[j])
                    k=-1
            k += 1
        j += 1
    #ya tenemos que la base es minimal, ahora vamos a hacer que sea
       reducida
    for t in range(len(xlG)):
        k=0
        while k < len(xlG):
            if k != t:
                for coef, mon in zip(R(xlG[t]).coefficients(), R(xlG[t])
                    .monomials()):
                    [d, s] = division(coef*mon,R(xlG[k]).lt())
                    if s==0:
                        xlG[t]=xlG[t]-d*xlG[k]
            k += 1

    return xlG
```

```
gbr=Reducir(gb)
```

```
print(gb==gbr)
```

```
True
```

Calculamos la base de Graver de A.

```
basegraver=gb.subs({y1: 1, y2: 1, y3: 1})
show(basegraver)
```

$[y_1y_3^{56} - 1, y_1y_3^{16} - y_2, y_1 - y_2^2y_3^{24}, y_2y_3^{40} - 1]$

Ahora definimos x que es una solución del sistema.

```
var('y1 y2 y3')
```

```
y=vector([3,2,24])
show(y)
```

$(3, 2, 24)$

Definimos también los vectores que forman la base de Graver.

```
v1=vector([1,0,56])
v2=vector([1,-1,16])
v3=vector([1,-2,-24])
v4=vector([0,1,40])
```

Queremos encontrar una solución x tal que $c \cdot x < c \cdot y = 36$. Por tanto, vamos a ver qué elementos obtenemos al sumar y con todos elementos de Gr_A.

```
sol1=y+v1
sol2=y+v2
sol3=y+v3
sol4=y+v4
show(sol1)
show(sol2)
show(sol3)
show(sol4)
```

$(4, 2, 80)$
$(4, 1, 40)$
$(4, 0, 0)$
$(3, 3, 64)$

Ahora teniendo en cuenta el valor de c vamos a ver si hay alguna solución x que cumpla $c \cdot x < c \cdot y$.

```
c=vector([4,0,1])
```

```
c*y <= c*sol1
```

```
True
```

```
c*y <= c*sol2
```

```
True
```

```
c*y <= c*sol3
```

```
False
```

```
c*y <= c*sol4
```

```
True
```

Tomamos ahora $z = (4, 0, 0)$ que es una solución que mejora la función objetivo y repetimos el proceso.

```
z=vector([4,0,0])
```

```
sol5=z+v1
sol6=z+v2
sol7=z+v3
sol8=z+v4
show(sol5)
show(sol6)
show(sol7)
show(sol8)
```

$(5, 0, 56)$
$(5, -1, 16)$
$(5, -2, -24)$
$(4, 1, 40)$

Comprobamos solo con aquellas soluciones que tienen entradas no negativas.

```
c*z <= c*sol5
```

```
True
```

```
c*z <= c*sol8
```

```
True
```

Por tanto, el vector $(4, 0, 0)$ es una solución que minimiza la función objetivo. Por tanto, la solución óptima es coger 4 envases de 60 ml y así sobrarían 16 ml.

Comprobamos que lo hemos hecho bien usando una función de Sage que calcula la solución óptima directamente.

```
    p = MixedIntegerLinearProgram(maximization=False)
x = p.new_variable(integer=True)

# Restricciones
p.add_constraint(56 * x[1] + 40 * x[2] - x[3] == 224)
p.add_constraint(x[1] >= 0)
p.add_constraint(x[2] >= 0)
p.add_constraint(x[3] >= 0)

# Funcion objetivo: minimizar 4x1 + x3
p.set_objective(4 * x[1] + x[3])

# Resolver
p.solve()

# Obtener las soluciones
sol_x1 = p.get_values(x[1])
sol_x2 = p.get_values(x[2])
sol_x3 = p.get_values(x[3])

# Mostrar los resultados
(sol_x1, sol_x2, sol_x3)
```

```
(4.0, 0.0, 0.0)
```

Bibliografía

[1] M. Hoekstra, *Gröbner bases and Graver bases used in integer programming* , https://fse.studenttheses.ub.rug.nl/11323/1/Masterscriptie.pdf.

[2] W.W. Adams, P. Loustaunau, *An Introduction to Grobner Bases*, Graduate Studies in Mathematics, Volume 3, AMS 1994.

[3] David A. Cox,John Little,Donal O'Shea, *Ideals, Varieties, and Algorithms*, Fourth Edition, Springer.

[4] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[5] Bernd Sturmfels, *Gröbner bases and Convex Polytopes*, Lectures University Series, 2010.

[6] Jaime Gutierrez, Josef Schicho, Martin Weimann, *Computer Algebra and Polynomials*, Springer, 2015.

[7] Jesús Gago Vargas, *Bases de Graver y optimización entera no lineal*, Universidad de Sevilla, 2011, https://grupo.us.es/fqm331/antigua/pdf/20111130_pres_1.pdf.

[8] Wikipedia contributors, *Graver Basis*, Wikipedia, disponible en https://en.wikipedia.org/wiki/Graver_basis.

[9] Alexander Schrijver, *Theory of linear and integer programming*, John Wiley & Sons, https://promathmedia.wordpress.com/wp-content/uploads/2013/10/alexander_schrijver_theory_of_linear_and_integerbookfi-org.pdf.

[10] J.A. De Loera, R. Hemmecke, M.Köppe, *Algebraic and Geometric Ideas in the Theory of Discrete Optimization*, SIAM 2013.

[11] Linde, P.(02 marzo 2024), *Cómo vender antibióticos sin desperdiciar pastillas para frenar la resistencia antimicrobiana*, *El País*, https://elpais.com/sociedad/2024-03-02/como-vender-antibioticos-sin-desperdiciar-pastillas-para-frenar-la-resistencia-antimicrobiana.html