# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

*FACULTAD DE CIENCIAS*

# *Aplicaciones de Bases de Gröbner y Teorías Algebraicas para Programación Entera*



# T E S I S

QUE PARA OBTENER EL TITULO DE :

**M A T E M A T I C A**

P R E S E N T A :

**Itnuit Janovitz Freireich**

**Director de Tesis: Dr. Rodolfo San Agustín Chi**

**Director de Tesis:**
**M. en I. Ma. del Carmen Hernández Ayuso**



**2001**

Universidad Nacional Autónoma de México

**M. EN C. ELENA DE OTEYZA DE OTEYZA**
Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:
Aplicaciones de Bases de Gröbner y teorias algebraicas para Programación Entera

realizado por Itnuit Janovitz Freireich

con número de cuenta 9756931-6, quién cubrió los créditos de la carrera de Matemáticas

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

| | | |
|---|---|---|
| Director de Tesis<br>Propietario | Dr. Rodolfo San Agustín Chi | |
| Director de Tesis<br>Propietario | M.en I. Ma del Carmen Hernández Ayuso | |
| Propietario | Dr. Manuel Falconi Magaña | |
| Suplente | M.en I. Ma de Luz Gasca Soto | |
| Suplente | Mat. José de Jesus Malagón Lopez | |

**Consejo Departamental de Matemáticas**

M. en C. ALEJANDRO BRAVO MOJICA

"A TOUS MES LOUPÉS, MES RATÉS,
MES VRAIS SOLEILS
TOUS LES CHEMINS QUI ME SONT PASSÉS À CÔTÉ.
A TOUS MES BATEAUX MANQUÉS,
MES MAUVAIS SOMMEILS.
A TOUT CEUX QUE JE N'AI PAS ÉTÉ.
A NOS ACTES MANQUÉS."

Jean Jacques Goldman.

A la memoria de mis abuelos,
Rosy, Imre
a quienes recuerdo siempre
con todo mi cariño.

"A TOUS MES LOUPÉS, MES RATÉS,
MES VRAIS SOLEILS
TOUS LES CHEMINS QUI ME SONT PASSÉS À CÔTÉ.
A TOUS MES BATEAUX MANQUÉS,
MES MAUVAIS SOMMEILS.
A TOUT CEUX QUE JE N'AI PAS ÉTÉ.
A NOS ACTES MANQUÉS."

Jean Jacques Goldman.

# Índice General

# Agradecimientos

Antes que nada quisiera expresar mi más profundo agradecimiento a todas las personas que hicieron que este trabajo, que para mí fué como un sueño, fuera posible. A Carmen Hernández Ayuso y Rodolfo San Agustín, gracias por haber aceptado el "paquete", pero sobre todo gracias por haber sido, desde hace ya varios años, quienes me han formado como matemática. Espero poder contar con su amistad siempre. A Lucy Gasca, Manuel Falconi y José Malagón les agradezco toda su ayuda y comentarios en cuanto al trabajo. Finalmente quiero agradecer a Javier Elizondo por haberme hecho descubrir la Geometria Algebraica y en particular los trabajos de Rekha Thomas y Bernd Sturmfels. Gracias Javier por todo tu apoyo y por tu amistad a lo largo de los ultimos años.

Así mismo quiero agradecer a la Fundación Telmex y al PROBETEL por el apoyo económico a lo largo de mi carrera y de este proyecto.

A todos los cuates de la facultad: Omi, Barbas, Era, Rogelio, Mefisto, Paola, Rafas, Benjamín, Adriana, Freya, Germán y a todos mis alumnos, les doy las gracias por aguantar mis grandes locuras y mis pláticas de matemáticas aplicadas y por estar siempre ahí, riendo de mis historias.

He tenido la suerte de poder contar, en todos los momentos de mi vida, con el apoyo y el cariño de mucha gente, lo que me ha permitido alcanzar todas mis metas y mis ideales. A cada una de esas personas que han estado en mi camino les agradezco cada instante, cada sonrisa, cada palabra alentadora.
A mi abuelita, Flory, por todo su amor y alegría
A Maggie y Eva, por hacerme siempre sentir que soy la "Cuchita" consentida

# Prólogo

Creo que desde siempre la geometría ha ejercido en mí una gran fascinación y desde inicios de mi carrera me han atraído las matemáticas aplicadas. Conforme fuí avanzando en mis estudios descubrí la Geometría Algebráica y el Algebra Conmutativa y aunque me sentí, y sigo sintiéndome, profundamente intimidada por la gran dificultad y diversidad de los conceptos y herramientas, mi interés en estas disciplinas ha ido en continuo aumento. Paralelamente a ello comencé a involucrarme con los fundamentos teóricos de la Investigación de Operaciones, particularmente de la Programación Entera, para la cuál el hecho de que los problemas sean $\mathcal{NP}$ – *completos* genera problemáticas muy importantes.

Cuando descubrí que existen aplicaciones de las Bases de Gröbner para Programación Entera me sentí inmensamente feliz al ver que era posible conjuntar las áreas de mi mayor preferencia, las cuáles parecían ajenas, y aportar nuevas perspectivas al tratamiento de los problemas de Programación Entera (PPE) utilizando la potente herramienta que son las Bases de Gröbner. Evidentemente me aboqué a la tarea de estudiar el tema.

En un comienzo estudié los enfoques que existen del problema, basándome en el artículo de Rekha R. Thomas: "Applications to Integer Programming" [**Tho98**] y posteriormente revisando los trabajos realizados por otros investigadores citados en dicho artículo, entre los cuales se encuentran [**CLO98, AL94**].

# Introducción

En los problemas de programación lineal se busca encontrar el valor óptimo (mínimo o máximo) de una función lineal dado un conjunto de restricciones también lineales. Esta clase de problemas surge en muchos ámbitos. En la industria, el ejército, la educación y el área de la salud entre otros, aparecen problemas de planes de producción, contratación de personal, ubicación de centros de atención, asignación, etcétera, los cuales en su interpretación matemática pueden ser formulados mediante modelos de programación lineal.

El problema general de programación lineal (PPL) se plantea como un problema de optimización

$$
\begin{aligned}
Min\, z = \sum_{j=1}^{m} c_j \sigma_j & \\
s.a \qquad A\boldsymbol{\sigma} &= \boldsymbol{b} \\
\boldsymbol{\sigma} &\geq 0
\end{aligned}
$$

con $A = (a_{ij})$; $\boldsymbol{b} = (b_1, \ldots, b_n)$; $a_{ij}, b_i \in \mathbb{Q}$; $c_j \in \mathbb{R}$; $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m) \in \mathbb{R}^m$; $i = 1, \ldots, n$; $j = 1 \ldots, m$.

Sin embargo, en muchas de las situaciones, este planteamiento no es suficiente y se requiere que las variables tengan solamente valores enteros, es decir se necesita trabajar con programación lineal entera.

Los problemas de programación lineal pueden ser resueltos de manera bastante eficiente usando métodos como el *Simplex* y el *Dual-Simplex*, sin embargo en programación entera la resolución de problemas es mucho más compleja ya que el problema general pertenece a la clase de problemas $\mathcal{NP}$ – *completos*,

CAPíTULO 1

# Programación Lineal Entera

## 1. Planteamiento

Los problemas de Programación Lineal son problemas de optimización cuyo rol es fundamental en el área de la Investigación de Operaciones. Un problema de Programación Lineal Entera (PPE) es un problema de Programación Lineal donde los valores de las variables de decisión están restringidos a los números enteros, es decir un problema en donde se busca encontrar el valor óptimo (máximo o mínimo) de una función lineal (función de costo) en $n$ variables de la forma

$$z = c_1\sigma_1 + \cdots + c_n\sigma_n$$

sujeto a (s.a) un conjunto de restricciones lineales dadas por

$$A\boldsymbol{\sigma} = \boldsymbol{b} \qquad \text{donde}$$
$$A = (a_{ij});\ \boldsymbol{b} = (b_1, \dots, b_n);\ a_{ij},\ b_i \in \mathbb{Z};\ c_j \in \mathbb{R};$$
$$\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_m) \in \mathbb{N}^m;\ i = 1, \dots, n;\ j = 1 \dots, m.$$

Decimos que un punto $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_m)$ es *solución factible* del PPE si cumple con el conjunto de restricciones asociado al problema.

Planteado desde el punto de vista geométrico, el problema equivale a encontrar el óptimo de la función lineal, lo que en el espacio $\mathbb{R}^n$ corresponde a un hiperplano, dentro del conjunto de restricciones, que consiste en un politopo (delimitado por la intersección de los hiperplanos generados por las restricciones). En el caso en donde las variables no están restringidas a la integralidad, cualquier punto de la superficie del politopo es solución factible, sin embargo en el caso del PPE los puntos factibles son aquéllos que se encuentran

Aunque el planteamiento es relativamente sencillo y muy semejante al de Programación Lineal, en donde los problemas se pueden resolver de manera relativamente sencilla usando los métodos *Simplex* y *Dual-Simplex*, resolver los PPE es muy complicado ya que los métodos de solución son poco eficientes; de hecho estos problemas pertenecen a una de las clases de problemas de mayor complejidad: los problemas $\mathcal{NP}$ – *completos*.

A continuación veremos las técnicas principales de resolución general con las que se ha trabajado: los métodos de corte y los de ramificación y acotamiento.

## 2. Métodos de corte

Como mencionamos anteriormente la resolución de un problema de Programación Lineal, donde las variables pueden tomar cualquier valor real, se puede realizar de manera efectiva usando el método Simplex o el Dual Simplex (ver [**SK89**] para la descripción detallada de estos métodos). Al enfrentarnos con un problema de Programación Entera la primera pregunta que surge es: ¿Qué pasaría si resuelvo el problema sin la restricción de integralidad y posteriormente refino la solución?

A finales de los años cincuentas y en la década de los sesentas, muchos investigadores trabajaron en el diseño de métodos de corte. La idea general de estos métodos es precisamente resolver el problema de Programación Lineal correspondiente a un problema de PPE pero permitiendo que las variables de decisión tomen cualquier valor, al cuál se llama *problema relajado* del PPE, y a partir de ello realizar "cortes"de la región factible que separen la solución óptima del problema, si no es entera, del conjunto de soluciones enteras factibles. Una vez hecho esto se resuelve el problema relajado dentro de la nueva región factible determinada por el corte y si la solución del nuevo problema sigue siendo no entera se "corta"de nuevo y así consecutivamente, hasta llegar a una solución óptima entera.

Dentro de los métodos de corte para los cuales se ha demostrado la convergencia, el más citado en la literatura es el método de la cortadura dual fraccional, diseñado por Gomory en 1958 [**Gom58**].

**2.1. El corte de Gomory.** El corte de Gomory es también conocido como método de la cortadura dual fraccional y utiliza el método dual simplex lexicográfico dentro de la tabla de Beale, lo que garantiza la convergencia para encontrar las soluciones de cada uno de los problemas relajados asociados con cada corte. A continuacón presentamos un método de solución de problemas usando este corte.

2.1.1. *La Tabla de Beale.* El primer paso de este método es escribir el programa entero despejando cada una de las variables de la siguiente manera:

Maximizar

$$
\begin{array}{llllllllll}
\sigma_0 = & c_0 & + & c_1(-\sigma_1) & + & c_2(-\sigma_2) & + \cdots & + & c_s(-\sigma_s) \\
\text{sujeto a} & & & & & & & & \\
\sigma_1 = & & & -1(-\sigma_1) & & & & & \\
\sigma_2 = & & & & & -1(-\sigma_2) & & & \\
\ldots & \ldots & & \ldots & & \ldots & & & \\
\sigma_s = & & & & & & & & -1(-\sigma_s) \\
\sigma_{s+1} = & -b_1 & + & a_{1,1}(-\sigma_1) & + & a_{2,1}(-\sigma_2) & + \cdots & + & a_{s,1}(-\sigma_s) \\
\ldots & \ldots & & \ldots & & \ldots & & & \\
\sigma_n = & -b_m & + & a_{1,m}(-\sigma_1) & + & a_{2,m}(-\sigma_2) & + \cdots & + & a_{s,m}(-\sigma_s)
\end{array}
$$

donde $\sigma_j \in \mathbb{N}$, para toda $j = 1, \ldots, n$. $\sigma_0$ corresponde a la función objetivo, y $\sigma_{s+1}, \ldots, \sigma_n$ a las $m$ variables de holgura correspondientes a las restricciones del problema.

A partir de este despeje de las variables se construye la Tabla de Beale:

Utilizando esta definición de corte, para resolver un problema entero se deben seguir los siguientes pasos:

- Resolver el problema relajado usando el algoritmo dual simplex en la Tabla de Beale correspondiente.
- Si la solución no es entera introducir un primer corte $x_{*1}$ con respecto a alguna de las variables básicas que viola la integralidad.
- Se repite el algoritmo dual simplex para resolver la infactibilidad que se crea al añadir el corte.
- Si la solución es entera, terminar. Si no, añadir un nuevo corte dentro de la nueva tabla óptima.

2.1.3. *Ejemplo.* Dado el problema:

$$\begin{array}{lrrcl} \text{maximizar} & -4x_1 & -5x_2 & = & x_0 \\ \text{sujeto a} & -x_1 & -4x_2 & \leq & -5 \\ & -3x_1 & -2x_2 & \leq & -7 \\ & x_1, & x_2 & \geq & 0, \text{ enteras.} \end{array}$$

Introducimos las variables de holgura $x_3$ y $x_4$ y construimos la Tabla de Beale:

| #1 | $\bar{c}$ | $(-x_1)$ | $(-x_2)$ |
|---|---|---|---|
| $x_0$ | 0 | 4 | 5 |
| $x_1$ | 0 | −1 | 0 |
| $x_2$ | 0 | 0 | −1 |
| $x_3$ | −5 | −1 | −4 |
| $x_4$ | −7 | $\boxed{-3}$ | −2 |

$(x_1 = x_2 = 0)$

Esta tabla es primal infactible y dual factible por lo que aplicamos el algoritmo dual simplex usando como criterio para el cambio de base el cociente mínimo por columnas, para el renglón con coeficiente de costo

Esta tabla es primal infactible por lo que hacemos un cambio de base, obteniendo la Tabla #4 que es óptima no entera. Añadimos entonces un segundo corte, con respecto a $x_2$ que no es entera, obteniendo $x_6$.

| #4 | $\bar{c}$ | $(-x_5)$ | $(-x_3)$ |
|---|---|---|---|
| $x_0$ | $-76/6$ | $11/6$ | $2/6$ |
| $x_1$ | $14/6$ | $-4/6$ | $2/6$ |
| $x_2$ | $4/6$ | $1/6$ | $-2/6$ |
| $x_3$ | 0 | 0 | $-1$ |
| $x_4$ | $8/6$ | $-10/6$ | $2/6$ |
| $x_5$ | 0 | $-1$ | 0 |
| $x_6$ | $-4/6$ | $-1/6$ | $\boxed{-4/6}$ |

$(x_1 = 14/6, x_2 = 4/6)$

Después de una última operación de pivoteo, obtenemos la tabla #5, que es óptima y cuya solución $x^* = (2, 1)$ es entera, por lo que es la solución óptima.

| #5 | $\bar{c}$ | $(-x_5)$ | $(-x_6)$ |
|---|---|---|---|
| $x_0$ | $-13$ | $7/4$ | $2/4$ |
| $x_1$ | 2 | $-3/4$ | $2/4$ |
| $x_2$ | 1 | $1/4$ | $-2/4$ |
| $x_3$ | 1 | $1/4$ | $-6/4$ |
| $x_4$ | 1 | $-7/4$ | $2/4$ |
| $x_5$ | 0 | $-1$ | 0 |
| $x_6$ | 0 | 0 | $-1$ |

$(x_1 = 2, x_2 = 1)$

Aunque teóricamente muy sencillos, los distintos métodos de corte son en la práctica realmente complejos y no se utilizan muy frecuentemente. Es más efectivo trabajar con algoritmos de ramificación y acotamiento ya que son fácilmente adaptables a problemas específicos, en particular aquellos para los cuales las variables toman solamente ciertos

2. $x_j > [a_j] + 1$

donde $[a_j]$ es el entero menor o igual a $a_j$.

Una vez hecho esto se resuelven los dos subproblemas, estableciendo como cota para cada una de las ramas que generan el valor de la función objetivo óptimo para cada uno de ellos. A continuación se repite el paso de ramificación en las distintas ramas y así sucesivamente. Si se encuentra una solución entera en alguna rama se detiene en ella el proceso de ramificación y se compara el costo de dicha solución con las cotas de las ramas restantes, deteniendo el proceso para aquéllas cuya cota sea inferior al valor factible entero encontrado. La solución óptima, como lo mencionamos antes, es aquélla cuyo costo es mayor que las cotas de las ramas restantes y que los costos de las demás soluciones enteras encontradas. Este método es el que fue propuesto por Dakin en 1965 [**Dak65**].

EJEMPLO 3.1. Retomando el problema visto en la sección anterior, tenemos:

$$\begin{array}{rrrcrl} \text{maximizar} & -4x_1 & -5x_2 & = & z & \\ \text{sujeto a} & -x_1 & -4x_2 & \leq & -5 & (x_3) \\ & -3x_1 & -2x_2 & \leq & -7 & (x_4) \\ & x_1, & x_2 & \geq & 0, & \text{enteras.} \end{array}$$

Al resolver la relajación del problema original obtenemos el nodo 1 del árbol, el cual produce la solución $z_1 = -112/10$, con $x_1 = 18/10$, $x_2 = 8/10$, $x_3 = 0$, $x_4 = 0$ la cual es no entera factible. Ramificamos entonces con respecto a $x_1$, que no es entera, añadiendo las restricciones:

- $x_1 \leq 1$
- $x_1 \geq 2$.

obteniendo así los nodos 2 y 3 respectivamente. El subproblema en el nodo 2 lleva a una solución entera factible $z_2 = -14$, con $x_1 = 1$, $x_2 = 2$, $x_3 = 4$, $x_4 = 0$. En el nodo 3 obtenemos como solución $z_3 = -47/4$, es:

$$\gamma^* = (a_1, \ldots, a_j, \ldots, a_n), \qquad a_j > 0, \text{ no entero}$$

y para el cual la variable $x_j$ puede tomar los valores

$$m_1, m_2, \ldots, m_r \qquad m_i \in \mathbb{N}, \quad i = 1, \ldots, r$$

estudiamos las ramas correspondientes a:

1. $x_j = m_1$
2. $x_j = m_2$

... ...

r. $x_j = m_r$

En este método, como en cada ramificación obligamos a una variable no entera en la solución anterior a que sea entera, el submodelo generado lleva a una solución entera, a una solución donde las variables básicas óptimas no enteras son distintas de $x_j$ o bien es no factible. La solución óptima se determina de la misma manera que en el caso de las variables libres.

EJEMPLO 3.2. Sea el problema

$$\begin{array}{rcccccccl} \text{maximizar} & 5x_1 & +3x_2 & +6x_3 & +6x_4 & +2x_5 & = & z \\ \text{sujeto a} & 5x_1 & +4x_2 & +7x_3 & +6x_4 & +2x_5 & \leq & 15 \\ & x_1, & x_2, & x_3, & x_4, & x_5, & = & 0,1. \end{array}$$

Para este problema obtenemos el siguiente árbol:

los métodos de resolución se vuelven más y más complicados desde el punto de vista computacional. Esto se debe a que los problemas enteros pertenecen a la clase de problemas $\mathcal{NP}$ – *completos*, que es una de las categorías de problemas más difícil de resolver algorítmicamente.

## 4. La complejidad algorítmica de los PPE

Como hemos hecho notar en las secciones anteriores, aunque existen algoritmos convergentes para los cuales podemos garantizar la obtención de una solución óptima en un número finito de pasos, eso no es útil desde el punto de vista operacional ya que el número de iteraciones necesarias para encontrar dicho óptimo puede ser muy grande. Como vimos en los ejemplos presentados, el número de iteraciones que deben realizarse en los algoritmos de Corte y Ramificación y Acotamiento crece de manera no proporcional con respecto al crecimiento del tamaño del problema, lo que nos hace pensar que dichos problemas son de difícil resolución. La teoría de complejidad algorítmica, desarrollada a partir de los trabajos de Cook y Karp [**Coo71, Kee72**] introduce la idea de eficiencia de los algoritmos y permite clasificar los problemas en "fáciles y difíciles".

Se dice que un algoritmo es *eficiente* cuando el número de operaciones necesarias para resolver un problema está acotado por una función polinomial. Los problemas de la clase $\mathcal{P}$ son aquéllos para los cuales existe un algoritmo eficiente que los resuelva. Los problemas de la clase $\mathcal{P}$ son "fáciles". Lo que queremos saber ahora es cómo podemos clasificar los problemas para los cuales no se conocen algoritmos polinomiales para su resolución.

DEFINICIÓN 4.1. Un *problema de decisión* es un problema para el cual los resultados sólo pueden tomar los valores CIERTO o FALSO.

Dado un problema de optimización combinatoria:

$$Min\, z = \sum_{j=1}^{m} c_j \sigma_j$$

Cook demostró que la clase de problemas $\mathcal{NP}$ – *completos* es no vacía, al probar que el problema de "satisfacibilidad" (SAT, del inglés satisfacibility) pertenece a dicha clase de complejidad. A partir de ello es entonces posible saber cuándo un problema es $\mathcal{NP}$ – *completo* al realizar una reducción polinomial con respecto al SAT o a cualquier otro problema $\mathcal{NP}$ – *completo*. En particular, es posible demostrar que el problema de "satisfacibilidad" se reduce polinomialmente al problema general de Programación Entera, por lo que los PPE son problemas $\mathcal{NP}$ – *completos*.

Si se lograra demostrar que existe un algoritmo polinomial que resuelva alguno de los problemas $\mathcal{NP}$ – *completos*, entonces se podría resolver en tiempo polinomial cualquier problema de dicha clase. Sin embargo aún los más eminentes investigadores han sido incapaces de probar que $\mathcal{P} = \mathcal{NP}$, por lo que se considera poco probable que existan algoritmos eficientes para la resolución de los problemas $\mathcal{NP}$–*completos*. La conjetura es entonces $\mathcal{P} \neq \mathcal{NP}$.

A la luz de estos resultados se relegó un poco el estudio del problema general de Programación Entera puesto que se buscaron posibles alternativas para la obtención de resultados. Se crearon entonces nuevas técnicas para el tratamiento de los PPE usando algoritmos de aproximación para poder generar soluciones factibles cuyos valores sean lo mejor posible dentro de intervalos de tiempo razonables.

Se recomienda consultar [**Sch86, Sak84, Man89**] para la definición formal de las clases de complejidad y la demostración de pertenencia a dicha clase del problema de Programación Entera.

En este trabajo se propone un enfoque del problema general usando las propiedades de las Bases de Gröbner . Esto resulta particularmente interesante ya que se trata de una muy importante herramienta del álgebra computacional. Esta última se encuentra en pleno desarrollo y se

CAPÍTULO 2

# Ideales Polinomiales y Bases de Gröbner

En este capítulo presentamos las nociones básicas así como los algoritmos principales usados en Ideales Polinomiales y Bases de Gröbner, mismos que nos permitiran resolver los problemas de programación lineal entera a partir de un planteamiento polinomial.

## 1. Ideales Polinomiales

DEFINICIÓN 1.1. Un *monomio* en una colección de variables $x_1, \ldots, x_n$ es un producto

$$x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$$

donde las $\alpha_i$ son enteros no negativos. Para abreviar, usaremos la notación de multi-índices $x^\alpha$ donde

$$\alpha = (\alpha_1, \ldots, \alpha_n) \qquad x = (x_1, \ldots, x_n)$$

El *grado total* de un monomio $x^\alpha$ es la suma de sus exponentes: $\alpha_1 + \cdots + \alpha_n$ y se denota por $|\alpha|$.

Denotaremos por $\mathbb{T}^n$ al conjunto de los monomios en $n$ variables, también llamados *productos de potencias.*

DEFINICIÓN 1.2. Una combinación lineal de un número finito de monomios con coeficientes en un campo k es un *polinomio* en $x_1, \ldots, x_n$. Un *término* de un polinomio es el producto de un elemento no trivial de k y un monomio.

Un polinomio en $x_1, \ldots, x_n$ con coeficientes en k es entonces de la forma

En efecto, tenemos

$$x^2 = x(x - y^2) + y(xy) \qquad \therefore\ x^2 \in \langle x - y^2, xy \rangle.$$

En el caso de los ideales polinomiales, un resultado fundamental es el siguiente.

TEOREMA 1.7 (Teorema de la Base de Hilbert). En el anillo de polinomios $k[x_1, \ldots, x_n]$ tenemos que:

1. Si $I$ es cualquier ideal de $k[x_1, \ldots, x_n]$, entonces existen polinomios $f_1, \ldots, f_s \in k[x_1, \ldots, x_n]$ tales que $I = \langle f_1, \ldots, f_s \rangle$; es decir $I$ es *finitamente generado.*
2. Si $I_1 \subseteq I_2 \subseteq \cdots \subseteq I_n \subseteq \cdots$ es una cadena ascendente de ideales de $k[x_1, \ldots, x_n]$, entonces existe $N$ tal que $I_N = I_{N+1} = I_{N+2} = \cdots$; es decir, $k[x_1, \ldots, x_n]$ es un *anillo noetheriano.*

La demostración de este teorema puede consultarse en distintos libros de Algebra Conmutativa, así como en [**AL94, CLO96**].

## 2. Orden Monomial

En este trabajo buscamos resolver los problemas de programación entera desde su traducción polinomial y en el Capítulo 3 veremos que esto es equivalente a determinar si un polinomio particular pertenece a la imagen de un mapeo polinomial, que es un caso particular del problema de la *membresía.* Este último se resuelve usando propiedades de las Bases de Gröbner y para ello es necesario poder comparar los polinomios de $k[x_1, \ldots, x_n]$ determinando un *orden* para los monomios que los conforman.

DEFINICIÓN 2.1. Un *orden monomial* en $k[x_1, \ldots, x_n]$ es una relación $>$ en $\mathbb{T}^n$ que satisface:

a) $>$ es una relación *(lineal) total*: se pueden comparar todos los elementos de $\mathbb{T}^n$.

DEFINICIÓN 2.4. El *orden lexicográfico graduado inverso* en $\mathbb{T}^n$ con $x_1 > x_2 > \cdots > x_n$ se define como: dados

$$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n),\ \boldsymbol{\beta} = (\beta_1, \dots, \beta_n) \in \mathbb{N}^n$$

definimos

$$\boldsymbol{x}^{\boldsymbol{\alpha}} <_{degrevlex} \boldsymbol{x}^{\boldsymbol{\beta}} \iff \begin{cases} \sum_{|\alpha|<|\beta|} \quad 6 \\ \sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \beta_i \ \text{ y } \boldsymbol{x}^{\boldsymbol{\alpha}} >_{lex} \boldsymbol{x}^{\boldsymbol{\beta}} \end{cases}$$

Denotaremos *degrevlex* a este orden.

EJEMPLO 2.5. Sean $x_1^3x_2x_3^9,\ x_1^2x_2^4x_3^7 \in k[x_1, x_2, x_3]$. Tenemos

- $x_1^3x_2x_3^9 >_{lex} x_1^2x_2^4x_3^7$
- $x_1^3x_2x_3^9 >_{deglex} x_1^2x_2^4x_3^7$
- $x_1^3x_2x_3^9 <_{degrevlex} x_1^2x_2^4x_3^7$

Sean $x_1^5,\ x_1x_2^2x_3^3x_4 \in k[x_1, x_2, x_3, x_4]$. Tenemos

- $x_1^5 >_{lex} x_1x_2^2x_3^3x_4$
- $x_1^5 <_{deglex} x_1x_2^2x_3^3x_4$
- $x_1^5 <_{degrevlex} x_1x_2^2x_3^3x_4$

Una vez fijado un orden monomial en $k[x_1, \dots, x_n]$, escribiremos todo polinomio
$f \in k[x_1, \dots, x_n]$, con $f \neq 0$ de la forma:

$$f = a_1\boldsymbol{x}_1^{\boldsymbol{\alpha}} + a_2\boldsymbol{x}_2^{\boldsymbol{\alpha}} + \cdots + a_r\boldsymbol{x}_r^{\boldsymbol{\alpha}},$$

donde $0 \neq a_i \in k$, $\boldsymbol{x}_i^{\boldsymbol{\alpha}} \in \mathbb{T}^n$, y de manera que $\boldsymbol{x}_1^{\boldsymbol{\alpha}} > \boldsymbol{x}_2^{\boldsymbol{\alpha}} > \cdots > \boldsymbol{x}_r^{\boldsymbol{\alpha}}$.

DEFINICIÓN 2.6. Dada la notación anterior para los polinomios, definimos:

- $\mathrm{lp}(f) = \boldsymbol{x}_1^{\boldsymbol{\alpha}}$, el *monomio principal* de $f$;
- $\mathrm{lc}(f) = a_1$, el *coeficiente principal* de $f$;
- $\mathrm{lt}(f) = a_1\boldsymbol{x}_1^{\boldsymbol{\alpha}}$, el *término principal* de $f$.

Definimos también $\mathrm{lp}(0) = \mathrm{lc}(0) = \mathrm{lt}(0) = 0$.

DEFINICIÓN 3.2. Dados $f$, $g$, $h \in k[x_1, \ldots, x_n]$, con $g \neq 0$, decimos que $f$ *se reduce a $h$ módulo $g$* en un paso, lo que escribimos:

$$f \xrightarrow{g} h,$$

si y sólo si $\mathrm{lp}(g)$ divide a algún término $X$ distinto de cero que aparece en $f$.

Se cumple entonces que

$$h = f - \frac{X}{lt(g)} g$$

OBSERVACIÓN 3.3. Lo que estamos haciendo aquí es definir la reducción como el proceso en el cual le restamos a $f$ el término $X$ para remplazarlo por términos estrictamente menores, y se puede entonces ver a $h$ como el residuo del primer paso de la división de $f$ entre $g$. Así mismo, podemos continuar este proceso hasta que ningún término de $f$ sea divisible por $\mathrm{lt}(g)$.

EJEMPLO 3.4. Sean
$f = y^2x + 4yx - 3x^2$, $g = 2y + x + 1 \in \mathbb{Q}[x, y]$. Escogemos el orden lexicográfico graduado, con $y > x$. Entonces

$$X = y^2x,$$

$$\begin{aligned} h &= (y^2x + 4xy - 3x^2) - \frac{y^2x}{2y}(2y + x + 1) \\ &= y^2x + 4xy - 3x^2 - \frac{1}{2}yx(2y + x + 1) \\ &= -\frac{1}{2}yx^2 + \frac{7}{2}yx - 3x^2 \end{aligned}$$

$f$ se reduce a $h$ módulo $g$ en un paso. Si continuamos reduciendo módulo $g$, obtenemos:

$$\begin{aligned} f &\xrightarrow{g} -\frac{1}{2}yx^2 + \frac{7}{2}yx - 3x^2 \\ &\xrightarrow{g} \frac{1}{4}x^3 + \frac{7}{2}yx - \frac{11}{4}x^2 \\ &\xrightarrow{g} \frac{1}{4}x^3 - \frac{9}{2}x^2 - \frac{7}{4}x \end{aligned}$$

ALGORITMO DE LA DIVISIÓN MULTIVARIADO 3.8.

***-ENTRADA:*** $f, f_1, \ldots, f_s \in k[x_1, \ldots, x_n]$ *con* $f_i \neq 0\ (1 \leqslant i \leqslant s)$

***-SALIDA:*** $u_1, \ldots, u_s, r$ *tales que* $f = u_1 f_1 + \cdots + u_s f_s + r$, *con* $r$ *reducido con respecto a* $\{f_1, \ldots, f_s\}$ *y*

*max* $\{lp(u_1)lp(f_1), \ldots, lp(u_s)lp(f_s), lp(r)\} = lp(f)$.

***-INICIALIZACION:***

$u_1 := 0,\ u_2 := 0, \ldots, u_s := 0,\ r := 0,\ h := f$

***-SIEMPRE QUE*** $\mathbf{h \neq 0}$

***SI*** *existe alguna* $i$ *tal que* $lp(f_i)$ *divide a* $lp(h)$,

· *escoger el índice menor tal que* $lp(f_i)$ *divide a* $lp(h)$

· *actualizar* $u_i$ *y* $h$*:*

$$u_i := u_i + \frac{lt(h)}{lt(f_i)}$$
$$h := h - \frac{lt(h)}{lt(f_i)} f_i$$

***EN OTRO CASO***

· *actualizar* $r$ *y* $h$*:*

$$r := r + lt(h)$$
$$h := h - lt(h)$$

Cabe notar que este algoritmo supone un orden en el conjunto de polinomios $\{f_1, \ldots, f_s\}$ para poder escoger $i$ como el índice menor tal que $\mathrm{lp}(f_i)$ divide a $\mathrm{lp}(h)$. Una vez fijado el orden, el residuo que obtenemos es único. Se recomienda consultar [**CLO96**] para la justificación del algoritmo.

## 4. Bases de Gröbner

DEFINICIÓN 4.1. Un conjunto de polinomios $G = \{g_1, \ldots, g_t\}$, $g_i \neq 0$ contenidos en un ideal $I$, forman una *base de Gröbner* para $I$ si y sólo si para todo $f \in I$ distinto de 0, existe $i \in \{1, \ldots, t\}$ tal que $\mathrm{lp}(g_i)$ divide a $\mathrm{lp}(f)$.

(ii)$\Longrightarrow$ (iii). Si tenemos ahora por hipótesis que $f \xrightarrow{G}_+ 0$, aplicando el algoritmo de la división, sabiendo que $r = 0$, se tiene:

$$f = h_1g_1 + \ldots + h_tg_t + 0$$
$$f = \sum_{i=1}^{t} h_ig_i \qquad y$$
$$lp(f) = max\ \{lp(h_1)lp(g_1), \ldots, lp(h_t)(g_t)\}$$

(iii)$\Longrightarrow$ (iv). Tenemos que $G \subseteq I$ por lo que $Lt(G) \subseteq Lt(I)$. Sea $f \in I$. Tenemos que $Lt(I) = \langle lt(f) \mid f \in I\rangle$. Por (iii) tenemos

$$f = \sum_{i=1}^{t} h_ig_i$$

Si nos fijamos ahora únicamente en los términos principales entonces se cumple:

$$lt(f) = \sum_{i=1}^{t} lt(h_i)lt(g_i)$$

y $\mathrm{lp}(f) = lp(h_s)lp(g_s)$ para alguna s. Por lo tanto,

$$f \in Lt(I) \Longrightarrow f \in Lt(G) \Longrightarrow Lt(I) \subseteq Lt(G).$$

(iv) $\Longrightarrow$ (i). Sea $f \in I$. Por (iv) sabemos que $f \in Lt(G)$, de donde

$$lt(f) = \sum_{i=1}^{t} h_i\, lt(g_i)$$

para $h_i \in k[x_1, \ldots, x_n]$. Si desarrollamos el lado derecho de la igualdad podemos encontrar $t \in \langle 1, \ldots, t\rangle$ tal que cada término es divisible entre $\mathrm{lp}(g_t)$, por lo que el lado izquierdo de la igualdad, $\mathrm{lt}(f)$, es divisible entre $\mathrm{lp}(g_t)$. Por la definición 4.1 $G$ es entonces una Base de Gröbner . □

Con este teorema se logra caracterizar las Bases de Gröbner de cuatro maneras distintas, pero aún no es posible concluir sobre la existencia de éstas. Para ello es necesario lo siguiente:

construir una Base de Gröbner para él. Es importante notar esto ya que las Bases de Gröbner que se obtienen al modificar el orden o al cambiar los generadores pueden ser distintas.

## 5. El algoritmo de Buchberger

Sabemos ahora qué es una Base de Gröbner y demostramos su existencia para todo ideal polinomial no trivial. La pregunta ahora es: dado un ideal polinomial $I$, ¿cómo construir una Base de Gröbner ? Antes de contestar esta pregunta definimos los siguientes conceptos.

DEFINICIÓN 5.1. Si $\boldsymbol{x}^\alpha$ y $\boldsymbol{x}^\beta$ son los monomios principales de $f$ y $g$ respectivamente, definimos $L = \boldsymbol{x}^\gamma$ como

$$\gamma = (\gamma_1, \dots, \gamma_n), \text{ con } \gamma_i = max\{\alpha_i, \beta_i\} \qquad \forall i = 1, \dots, n.$$

$L$ es el *mínimo común multiplo* de $lp(f)$ y $lp(g)$.

DEFINICIÓN 5.2. Sea $0 \neq f,\ g \in k[x_1, \dots, x_n]$.
Sea $L$ =mcm$(lp(f), lp(g))$. Definimos el *S-polinomio de $f$ y $g$* como:

$$S(f,g) = \frac{L}{lt(f)}f - \frac{L}{lt(g)}g.$$

EJEMPLO 5.3. Sea $f = x^3y^2 - x^2y^3 + x$ y $g = 3x^4y + y^2$ en $\mathbb{R}[x,y]$ con el órden lexicográfico graduado con $x > y$. Entonces $L = x^{max\{3,4\}}y^{max\{2,1\}}$ $L = x^4y^2$. Entonces

$$\begin{aligned} S(f,g) &= \frac{x^4y^2}{x^3y^2}.f - \frac{x^4y^2}{3x^4y}.g \\ &= x.f - \frac{1}{3}y.g \\ &= x^4y^2 - x^3y^3 - \frac{1}{3}3x^4y^2 - \frac{1}{3}y^3 \\ &= -x^3y^3 - \frac{1}{3}y^3. \end{aligned}$$

LEMA 5.4. *Sean $f_1, \dots f_s \in k[x_1, \dots, x_n]$ tales que $lp(f_i) = X$ distinto de cero para toda $i$, $i = 1, \dots, s$. Sea $f = \sum_{i=1}^{s} c_i f_i$ con $c_i \in k$.*

Es importante observar que una base de Gröbner es un conjunto generador del ideal $I$ que no es minimal pero si es el que tiene el número mínimo de elementos para que se cumpla la definición. El $S$-polinomio lo que hace es cancelar los términos principales de $f$ y $g$, lo que simplifica muchos de los cálculos, evita entrar en cíclos dentro de los algoritmos cuando se tienen dos polinomios con los mismos términos principales y permite eliminar ciertos polinomios del conjunto generador. El resultado siguiente muestra la importancia y el carácter "especial" (Special) del S-polinomio.

TEOREMA 5.5 (Buchberger). Sea $G = \{g_1, \ldots, g_t | 0 \neq g_i \in k[x_1, \ldots, x_n]\}$. $G$ es una Base de Gröbner para el ideal $I = \langle g_1, \ldots g_t \rangle$ si y sólo si para toda $i \neq j$,

$$S(g_i, g_j) \xrightarrow{G}_+ 0.$$

DEMOSTRACIÓN. Como $G$ es una Base de Gröbner para el ideal $I = \langle g_1, \ldots, g_t \rangle$, entonces $S(g_i, g_j) \in I$ y por el Teorema 4.3 se cumple que: $S(g_i, g_j) \xrightarrow{G}_+ 0$ para todo $i \neq j$.

Supongamos ahora $S(g_i, g_j) \xrightarrow{G}_+ 0$ para todo $i \neq j$. Sea $f \in I$, entonces como $I = \langle g_1, \ldots, g_t \rangle$ podemos expresar $f$ en términos de las $g_i$, ya que $G$ es un conjunto generador de $I$. Sin embargo, esto de puede hacer de distintas maneras puesto que una base de Gröbner no es un conjunto minimal. Escogemos representarlo como:

$$f = \sum_{i=1}^{t} h_i g_i$$

y sea

$$X = \max_{1 \leq i \leq t} \{lp(h_i), lp(g_i)\}$$

Si $X = lp(f)$ entonces por el Teorema4.3 (iii) $G$ es una Base de Gröbner . Si $X > lp(f)$ entonces sea $S = \{i \mid lp(h_i)lp(g_i) = X\}$. Para todo $i \in S$, podemos escribir $h_i = c_i X_i + p_i$ con $p_i$ un polinomio de términos menores que $h_i$. Sea $g = \sum_{i \in S} c_i X_i g_i$. Se cumple entonces que con $\max_{1\leq v\leq t}\{lp(h'_v)lp(g_v)\} < lp(S(X_ig_i, X_jg_j)) < X$, lo cual es una contradicción ya que supusimos que $lp(f) < X$. □

El teorema de Buchberger (Algorítmo 5.5) nos da una estrategia para obtener las Bases de Gröbner : reducir los S-polinomios y si el residuo es distinto de cero entonces lo añadimos a la lista de polinomios de un conjunto generador del ideal, y repetimos el proceso hasta que todos los S-polinomios se reduzcan a cero. El algoritmo es entonces:

ALGORITMO DE BUCHBERGER 5.6.

***-ENTRADA:*** $F = \{f_1, \dots, f_s\} \subseteq k[x_1, \dots, x_n]$, *con* $f_i \neq 0$ $(1 \leq i \leq s)$.

***-SALIDA:*** $G = \{g_1, \dots, g_t\}$ *una Base de Gröbner para* $\langle f_1, \dots, f_s\rangle$.

***-INICIALIZACION:*** $G := F$, $\mathcal{G} := \{\{f_i, f_j\} \mid f_i \neq f_j \in G\}$.

***-SIEMPRE QUE*** $\mathcal{G} \neq 0$

· *escoger cualquier pareja* $\{f, g\} \in \mathcal{G}$

$\mathcal{G} := \mathcal{G} - \{\{f, g\}\}$

$S(f, g) \xrightarrow{G}_+ h$, *donde h es reducido con respecto a* $G$.

***SI*** h $\neq 0$ ***ENTONCES***

· *actualizar* $\mathcal{G}$ *y* $G$*:*

$\mathcal{G} := \mathcal{G} \cup \{\{u, h\} \mid \text{para todo}\, u \in G\}$

$G := G \cup \{h\}$.

EJEMPLO 5.7. Sean $f_1 = y^2 + yx + x^2$, $f_2 = y + x$, y $f_3 = y \in \mathbb{Q}[x, y]$ ordenados por el orden *lex* con $x < y$

INICIALIZACION: $G := \{f_1, f_2, f_3\}$, $\mathcal{G} := \{\{f_1, f_2\}, \{f_1, f_3\}, \{f_2, f_3\}\}$.

PRIMERA ITERACION: $\mathcal{G} = \{\{f_1, f_2\}, \{f_1, f_3\}, \{f_2, f_3\}\} - \{f_1, f_2\}$, $S(f_1, f_2) = x^2$, y $x^2$ es reducido con respecto a $\{f_1, f_2, f_3\}$. Sea $f_4 = x^2$, entonces actualizamos:

$\mathcal{G} = \{\{f_1, f_3\}, \{f_1, f_4\}, \{f_2, f_3\}, \{f_2, f_4\}, \{f_3, f_4\}\}$.

$G = \{f_1, f_2, f_3, f_4\}$.

de polinomios $\{f, g\}$ de manera indistinta. Queremos entonces agregar ciertas condiciones para los polinomios para así obtener cierta unicidad.

DEFINICIÓN 6.1. Una Base de Gröbner es *mínimal*, si para todo $i$, $lc(g_i) = 1$ y para toda $i \neq j$, $lp(g_i)$ no divide a $lp(g_j)$

LEMA 6.2. *Sea $G = \{g_1, \ldots, g_t\}$ una Base de Gröbner para el ideal $I$. Si $lp(g_2)$ divide a $lp(g_1)$, entonces $\{g_2, \ldots, g_t\}$ es también Base de Gröbner para el ideal $I$.*

DEMOSTRACIÓN. Si $f \in k[x_1, \ldots, x_n]$ es tal que $lp(f)$ es divisible entre $lp(g_1)$, entonces como por hipótesis $lp(g_1)$ es divisible entre $lp(g_2)$, se cumple que $lp(g_2)$ divide a $lp(f)$. Por lo tanto, por la Definición 4.1, $\{g_2. \ldots, g_t\}$ es Base de Gröbner para $I$. □

A partir de la definición y el lema anterior tenemos el siguiente resultado:

COROLARIO 6.3. *Sea $G = \{g_1, \ldots, g_t\}$ una Base de Gröbner para el ideal $I$. Para obtener una Base de Gröbner minimal a partir de $G$, se eliminan todas las $g_i$ para las cuales existe $g_j\, j \neq i$ tal que $lp(g_i) \mid lp(g_j)$, y se divide cada $g_s$ restante entre su coeficiente principal.*

EJEMPLO 6.4. En el Ejemplo 5.7 tenemos que en la Base de Gröbner $G = \{f_1, \ldots, f_5\}$ se tiene $lp(f_1) = y^2 = y.y$ por lo que se cumple $lp(f_1) \mid lp(f_2) = lp(f_3)$. Podemos entonces quitar $f_1$ de la base. También podemos eliminar a $f_2$ o $f_3$, ya que $lp(f_2) \mid lp(f_3)$ y vice versa. Esto muestra que la base minimal que se obtiene no es única; en este caso puede contener a $f_2$ o a $f_3$. Además, $lp(f_4) = x^2 = x \cdot x$ lo que indica que $lp(f_4) \mid lp(f_5)$, por lo que podemos quitar $f_4$ de la base. Como los coeficientes principales de todos los polinomios son 1, entonces tenemos que

$$G_1 = \{f_2, f_5\} \quad y \quad G_2 = \{f_3, f_5\}$$

son bases minimales para I.

COROLARIO 6.7. *Sea $G = \{g_1, \dots, g_t\}$ una Base de Gröbner minimal para el ideal I. Realizamos el siguiente proceso de reducción:*

$g_1 \xrightarrow{H_1}_+ h_1$, *con $h_1$ reducido con respecto a $H_1 = \{g_2, \dots, g_t\}$*

$g_2 \xrightarrow{H_2}_+ h_2$, *con $h_2$ reducido con respecto a $H_2 = \{h_1, g_3, \dots, g_t\}$*

$g_3 \xrightarrow{H_3}_+ h_3$, *con $h_3$ reducido con respecto a $H_3 = \{h_1, h_2, g_4, \dots, g_t\}$*

$\vdots$

$g_t \xrightarrow{H_t}_+ h_t$, *con $h_t$ reducido con respecto a $H_t = \{h_1, h_2, \dots, h_{t-1}\}$*

*Entonces $H = \{h_1, \dots, h_t\}$ es una Base de Gröbner reducida para el ideal $I$.*

DEMOSTRACIÓN. Como $G$ es una Base de Gröbner minimal, entonces se cumple $lp(h_i) = lp(g_i)$ para toda $i = 1, \dots, t$. Por lo tanto, por la definición de bases de Gröbner, $H$ es una Base de Gröbner . Esta es además minimal ya que $G$ lo es. Como el proceso de división de $g_i$ por $h_1, \dots, h_{i-1}, g_{i+1}, \dots, g_t$ se realiza eliminando por reducción términos de $g_i$ con los monomios principales de $h_1, \dots, h_{i-1}$ y de $g_{i+1}, , g_t$ y donde $lp(h_j) = lp(g_j)$ para toda $j$, entonces $h_i$ es reducido con respecto a $H - \{h_i\}$, por lo que a partir de la definición dada, $H$ es una Base de Gröbner reducida para I. □

TEOREMA 6.8 (Buchberger). Dado un orden monomial y un ideal $I$ existe una única Base de Gröbner reducida asociada a $I$.

DEMOSTRACIÓN. Por el corolario anterior tenemos que todo ideal $I$ tiene una Base de Gröbner reducida. Queremos probar que ésta es única. Sea $G = \{g_1, \dots, g_t\}$ y $H = \{h_1, \dots, h_t\}$ Bases de Gröbner reducidas para $I$. Por el lema 6.5, se cumple que ambas tienen el mismo número de elementos, y podemos entonces asumir que $lt(h_i) = lt(g_i)$ para toda $i$. Supongamos que $g_i \neq h_i$ para alguna $i \in [1, t]$. Entonces, dado que $g_i - h_i \in I$ existe $j$ tal que $lp(h_j)$ divide a $lp(g_i - h_i)$. Como $lp(g_i - h_i) < lp(h_i)$, ya que $lt(g_i) = lt(h_i)$, esto implica que $i \neq j$. Se tiene por lo tanto que $lp(h_j) = lp(g_j)$ divide a algún término de $g_i$ o de $h_i$

DEMOSTRACIÓN. Tenemos que $f \equiv g \ (mod\, I) \Longrightarrow f - g \in I \Longrightarrow$ existe $q \in I$ tal que $f = q + g$. Se cumple entonces

$$N_G(f) = N_G(q+g) = N_G(q) + N_G(g)$$

pero $N_G(q) = 0$ ya que $q \in I$, entonces $N_G(f) = N_G(g)$.

Por otro lado, si $N_G(f) = N_G(g)$, entonces

$$f - g = (f - N_G(f)) - (g - N_G(g)) \in I$$

Esto es por el Algoritmo de la División ya que sabemos que $N_G(f)$ es el residuo de la división de $f$ entre $G$ por lo que $f - N_G(f) \in I$ y lo mismo ocurre para $g$. $\square$

Tenemos ahora las herramientas necesarias para construir, dados un ideal polinomial y un orden monomial, una Base de Gröbner reducida (única) que nos va a permitir resolver distintos problemas en el ideal $I$. Algunas de las aplicaciones de las Bases de Gröbner son:

- Dado un polinomio $f \in k[x_1, \ldots, x_n]$ determinar cuándo $f \in I$. Este es el problema de la membresía.
- Determinar cuándo dos ideales $I$, $J \in k[x_1, \ldots, x_n]$ son iguales.
- Encontrar bases para un $k$-espacio vectorial $k[x_1, \ldots, x_n]/I$.

En nuestro caso, utilizaremos las Bases de Gröbner para resolver problemas de Programación Entera.

CAPíTULO 3

# Bases de Gröbner y Programación Entera

En este capítulo presentaremos el uso de las Bases de Gröbner y de ciertos mapeos polinomiales para resolver problemas de programación entera. Lo que queremos es traducir un problema de programación entera a uno equivalente de ideales polinomiales, usar las herramientas de las Bases de Gröbner para resolver el problema y finalmente traducir la solución polinomial a la solución del problema original.

## 1. Planteamiento Polinomial

Hasta ahora sabemos cómo trabajar con Bases de Gröbner en ideales polinomiales y conocemos la estructura de los problemas de Programación Entera, lo que nos hace falta es traducir el lenguage de la Programación Entera, que es el del álgebra lineal, a polinomios.

El problema general de programación entera en su forma estandar está dado, como vimos en el Capítulo 2, por:

$$\begin{aligned} Min\, z = c(\boldsymbol{\sigma}) = \sum_{j=1}^{m} c_j\sigma_j \\ s.a \qquad A\boldsymbol{\sigma} = \boldsymbol{b} \\ \boldsymbol{\sigma} \in \mathbb{N}^m \end{aligned}$$

con $A = (a_{ij})$; $\boldsymbol{b} = (b_1, \dots, b_n)$; $a_{ij}$, $b_i \in \mathbb{Z}$; $c_j \in \mathbb{R}$; $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_m) \in \mathbb{N}^m$; $\quad i = 1, \dots, n$; $j = 1 \dots, m$.

**1.1. Problemas con coeficientes enteros no negativos.** Primero veremos el caso particular de los problemas cuyos coeficientes $a_{ij}$ y $b_i$ son cuyo producto es

$$(x_1^3x_2^4)^{\sigma_1}(x_1^2x_2)^{\sigma_2}(x_1x_2)^{\sigma_3}x_1^{\sigma_4} = x_1^{10}x_2^5$$

y el mapeo polinomial correspondiente que asocia las variables $y_1, y_2, y_3, y_4$ con las variables $x_i$ es entonces:

$$\begin{array}{ccc}
\mathbb{Q}[y_1, y_2, y_3, y_4] & \xrightarrow{\phi} & \mathbb{Q}[x_1, x_2] \\
y_1 & \mapsto & x_1^3x_2^4 \\
y_2 & \mapsto & x_1^2x_2 \\
y_3 & \mapsto & x_1x_2 \\
y_4 & \mapsto & x_1
\end{array}$$

A partir de estos resultados obtenemos entonces que el problema de programación entera planteado originalmente es equivalente a optimizar la función objetivo cumpliendo con la ecuación:

$$\phi(y_1^{\sigma_1}y_2^{\sigma_2}\cdots y_m^{\sigma_m}) = x_1^{b_1}x_2^{b_2}\cdots x_n^{b_n}$$

Nuestro problema involucra ahora a polinomios en las variables $x_1, \dots, x_n, y_1, \dots, y_m$ y al mapeo polinomial $\phi$.

**1.2. Problemas con coeficientes enteros.** Ahora queremos plantear el caso general donde los coeficientes $a_{ij}$ y $b_i$ son enteros no forzosamente no negativos. Como vimos en la Definición 1.1, los exponentes de las variables de un polinomio deben ser positivos, por lo que no podemos trabajar en $k[x_1, \dots, x_n]$ directamente como en el caso anterior, ya que se podrían obtener coeficientes negativos para las variables $x_i$. Debemos entonces encontrar la manera de modificar los coeficientes $a_{ij}$ y $b_i$ para obtener otros coeficientes no negativos para los cuales podamos introducir polinomios, y que lleven a resultados equivalentes.

Con este objeto, se define $\alpha_j \geq \underset{\{i \,|\, a_{ij}<0\}}{max} \{|a_{ij}|\}$ para cada $j = 1, \dots, m$. Entonces tenemos que existen enteros no negativos $a'_{ij}$ tales que

$$(a_{1j}, a_{2j}, \dots, a_{nj}) = (a'_{1j}, a'_{2j}, \dots, a'_{nj}) + \alpha_j(-1, -1, \dots, -1)$$

y podemos introducir las variables $y_j$ determinadas por el homomorfismo:

$$\begin{array}{rcl} k[y_1,\ldots,y_m] & \xrightarrow{\phi} & k[x_1,x_2,\ldots,x_n,w]/I \\ y_j & \mapsto & x_1^{a'_{1j}}x_2^{a'_{2j}}\cdots x_n^{a'_{nj}}w^{\sigma_j}+I \end{array}$$

EJEMPLO 1.4. Dado el conjunto de restricciones:

$$\begin{cases} 2\sigma_1+\sigma_2-3\sigma_3-\sigma_4=4 \\ -3\sigma_1+2\sigma_2-2\sigma_3-\sigma_4=-3 \\ \sigma_i\in\mathbb{N},\ i=1,2,3,4. \end{cases}$$

Escogemos $\alpha_j = \underset{\{i\,|\,a_{ij}<0\}}{max}\{|a_{ij}|\}$, entonces

$$\begin{aligned} \alpha_1 &= max\{|3|,0\}=3 \\ \alpha_2 &= max\{0\}=0 \\ \alpha_3 &= max\{|-3|,|-2|,0\}=3 \\ \alpha_4 &= max\{|-1|,0\}=1 \end{aligned}$$

y tenemos entonces:

$$\begin{array}{ll} a'_{11}=a_{11}+\alpha_1=2+3=5 & a'_{21}=a_{21}+\alpha_1=-3+3=0 \\ a'_{12}=1 \quad a'_{22}=2 & \\ a'_{13}=0 \quad a'_{23}=1 & \\ a'_{14}=2 \quad a'_{24}=0 & \end{array}$$

Tenemos también $\beta = max\{|-3|,0\}=3$, por lo que $b'_1=7$, $b'_2=0$. Introducimos ahora las variables $x_1, x_2$ y tenemos $I=\langle x_1x_2w-1\rangle$ con lo que la ecuación de las restricciones es:

$$(x_1^5x_2^0w^3)^{\sigma_1}(x_1^1x_2^2w^0)^{\sigma_2}(x_1^0x_2^1w^3)^{\sigma_3}(x_1^2x_2^0w^1)^{\sigma_4}+I=x_1^7x_2^0w^3+I,$$

El mapeo polinomial que asocia las variables $y$ con $x_1, x_2$ y $w$ en el anillo afín es:

$$\begin{array}{rcl} \mathbb{Q}[y_1, y_2, y_3, y_4] & \xrightarrow{\phi} & \mathbb{Q}[x_1, x_2, w]/I \\ y_1 & \mapsto & x_1^3 x_2^4 + I \\ y_2 & \mapsto & x_2^3 w^2 + I \\ y_3 & \mapsto & x_1^2 w^1 + I \\ y_4 & \mapsto & x_2 w + I \end{array}$$

Una vez más, el problema planteado originalmente, con coeficientes $a_{ij}$ y $b_j$ enteros consiste, desde el punto de vista polinomial, en optimizar la función objetivo sujeta a:

$$\phi(y_1^{\sigma_1} y_2^{\sigma_2} \cdots y_m^{\sigma_m}) = x_1^{b'_1} x_2^{b'_2} \cdots x_n^{b'_n} w^\beta + I$$

Tenemos ahora el planteamiento polinomial del problema general de Programación Entera. Lo que tenemos que encontrar es el conjunto de soluciones factibles del problema, es decir queremos encontrar $\sigma = (\sigma_1, \ldots, \sigma_m) \in \mathbb{N}^m$ tal que se cumple la ecuación anterior.

Para ello, necesitamos estudiar el mapeo $\phi$ y trabajar con Bases de Gröbner para determinar cuándo un monomio $y^\sigma$ es preimagen bajo $\phi$ de $x^{b'} w^\beta + I$, para así conocer los puntos factibles y poder entonces optimizar la función objetivo.

## 2. Eliminación

Como vimos en el Capítulo 1, al trabajar con Bases de Gröbner es necesario fijar un orden monomial. Es importante por lo tanto encontrar ordenes adaptados al tipo de problemas que se quieren resolver, tomando en cuenta la estructura de los ideales, para llegar a la solución buscada. Como vimos en la sección anterior, para los problemas que queremos resolver vamos a trabajar con los conjuntos de variables $\{y_1, \ldots, y_m\}$ y $\{x_1, \ldots, x_n\}$. Suponiendo que están definidos los ordenes $>_x$ y $>_y$ de los monomios en las variables $x$ y $y$
respectivamente definimos el siguiente orden para los monomios donde aparecen variables $x$ y $y$:

(iii) Se establece un buen orden ya que $1 < y < x < xy$.

Por otro lado, si $Y$ es un monomio en las variables $y$ y $Z$ es un monomio en $k[y_1, \ldots, y_m, x_1, \ldots, x_n]$ con alguna de las $x_i$ elevada a una potencia positiva $\alpha_i$, entonces para cualquier orden sobre las variables $x$ se cumple que $x_i^{\alpha_i} > 1$ ya que un orden monomial establece un buen orden, y entonces tenemos

$$Z \geq_x x_i^{\alpha_i} >_x Y \qquad \Longrightarrow \qquad Z > Y.$$

□

La ventaja de los ordenes por eliminación aparece al introducir Bases de Gröbner para ideales en $k[y_1, \ldots, y_m, x_1, \ldots, x_n]$, (y de igual manera para $k[y_1, \ldots, y_m, x_1, \ldots, x_n, w]$ en cuyo caso las variables $x$ y $w$ son mayores que las $y$), ya que permite la distinción entre las variables. El resultado principal de la teoría de ordenes de eliminación es el siguiente.

TEOREMA 2.4. Sea $I$ un ideal no trivial de $k[y_1, \ldots, y_m, x_1, \ldots, x_n]$ y sea $<$ un orden por eliminación con las variables $x$ mayores que las $y$. Sea $G = \{g_1, \ldots, g_t\}$ una Base de Gröbner para el ideal $I$. Entonces $G \cap k[y_1, \ldots, y_m]$ es Base de Gröbner para el ideal $I \cap k[y_1, \ldots, y_m]$. El ideal $I \cap k[y_1, \ldots, y_m]$ se llama *ideal de eliminación* ya que estamos "eliminando" las variables $x$ del ideal $I$.

DEMOSTRACIÓN. Como $G \subset I$ entonces se cumple $G \cap k[y_1, \ldots, y_m] \subset I \cap k[y_1, \ldots, y_m]$. Sea $f \in I \cap k[y_1, \ldots, y_m]$. Como $G$ es Base de Gröbner para $I$ entonces por la Definición 4.1 existe $i$ tal que $lp(g_i)$ divide a $lp(f)$. Como $f$ solo tiene variables $y$, entonces $lp(g_i)$ solo tiene variables $y$, es decir $g_i \in G \cap k[y_1, \ldots, y_m]$. Tenemos entonces que para todo polinomio $f \in I \cap k[y_1, \ldots, y_m]$ existe $g_i \in G \cap k[y_1, \ldots, y_m]$ tal que $lp(g_i) \mid lp(f)$, lo que, por la Definición 4.1, $G \cap k[y_1, \ldots, y_m]$ es una Base de Gröbner para $I \cap k[y_1, \ldots, y_m]$. □

LEMA 3.1. *Sean $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ elementos de un anillo conmutativo $R$. Entonces el elemento $a_1 a_2 \cdots a_n - b_1 b_2 \cdots b_n$ pertenece al ideal $\langle a_1 - b_1, a_2 - b_2, \ldots, a_n - b_n \rangle$.*

DEMOSTRACIÓN. Tenemos que

$$\begin{aligned}
&a_1 a_2 \cdots a_n - b_1 b_2 \cdots b_n \\
&\quad = a_1(a_2 \cdots a_n - b_2 \cdots b_n) + b_2 \cdots b_n(a_1 - b_1) \\
&\quad = a_1[a_2(a_3 \cdots a_n - b_3 \cdots b_n) + b_3 \cdots b_n(a_2 - b_2)] + b_2 \cdots b_n(a_1 - b_1) \\
&\quad = a_1 a_2(a_3 \cdots a_n - b_3 \cdots b_n) + a_1 b_3 \cdots b_n(a_2 - b_2) + b_2 \cdots b_n(a_1 - b_1) \\
&\quad = \cdots \\
&\quad = a_1 a_2 \cdots a_{n-1}(a_n - b_n) + a_1 a_2 \cdots a_{n-2} b_n(a_{n-1} - b_{n-1}) + \cdots \\
&\qquad \cdots + a_1 b_3 \cdots b_n(a_2 - b_2) + b_2 \cdots b_n(a_1 - b_1)
\end{aligned}$$

Por lo que efectivamente

$$a_1 a_2 \cdots a_n - b_1 b_2 \cdots b_n \in \langle a_1 - b_1, a_2 - b_2, \ldots, a_n - b_n \rangle$$

□

El siguiente teorema permite encontrar los elementos de la imagen de $\phi$ a partir de un ideal de $k[y_1, \ldots, y_m, x_1, \ldots, x_n]$ y su Base de Gröbner reducida con respecto al orden de eliminación con las $x$ mayores que las $y$.

TEOREMA 3.2. Construyamos el ideal $K = \langle y_1 - f_1, \ldots, y_m - f_m \rangle$ en $k[y_1, \ldots, y_m, x_1, \ldots, x_n]$ y sea $G$ una Base de Gröbner de $K$ con respecto al orden por eliminación con las variables $x$ mayores que las $y$. Entonces $f \in k[x_1, \ldots, x_n]$ está en $im(\phi)$ si y sólo si existe $h \in k[y_1, \ldots, y_m]$ tal que $f \xrightarrow{G}_+ h$. En dado caso, se tiene que $f = \phi(h)$.

DEMOSTRACIÓN. Sea $f \in k[x_1, \ldots, x_n]$ es decir $f(x_1, \ldots, x_n)$ en $im(\phi)$. Entonces existe $g \in k[y_1, \ldots, y_m]$, $g(y_1, \ldots, y_m)$, tal que $f(x_1, \ldots, x_n) = \phi(g) = g(f_1, \ldots, f_m)$. Tomamos el polinomio

Aplicando $\phi$ a ambos lados de la ecuación, tenemos:

$$\begin{aligned}\phi(f(x) - h(y)) &= \phi(\sum_{i=1}^{m} g_i(y_i - f_i)) \\ f(x) - \phi(h) &= \sum_{i=1}^{m} g_i(f_i - f_i) \\ f(x) &= \phi(h) \quad \text{por lo que } f \in im(\phi)\end{aligned}$$

□

A partir de este teorema tenemos entonces un método para saber si un polinomio $f \in k[x_1, \dots, x_n]$ está en la imagen de $\phi$. Primero, dado el mapeo $\phi$ construimos $K$ y $G$ una Base de Gröbner de $K$, con respecto a un orden de eliminación. A continuación reducimos el polinomio $f$ módulo $G$ para obtener su forma normal $h$ y si $h \in k[y_1, \dots, y_m]$ entonces $f \in im(\phi)$.

EJEMPLO 3.3. Consideremos el mapeo polinomial del Ejemplo 1.1

$$\begin{array}{ccc} \mathbb{Q}[y_1, y_2, y_3, y_4] & \xrightarrow{\phi} & \mathbb{Q}[x_1, x_2] \\ y_1 & \mapsto & x_1^3x_2^4 \\ y_2 & \mapsto & x_1^2x_2 \\ y_3 & \mapsto & x_1x_2 \\ y_4 & \mapsto & x_1 \end{array}$$

y queremos determinar si $x_1^{10}x_2^5$ está en la imagen de $\phi$. Construimos una Base de Gröbner para el ideal

$$K = \langle y_1 - x_1^3x_2^4,\ y_2 - x_1^2x_2,\ y_3 - x_1x_2,\ y_4 - x_1 \rangle,$$

que está en $\mathbb{Q}[y_1, y_2, y_3, y_4, x_1, x_2]$, con respecto al orden lexicográfico con $x_1 > x_2 > y_1 > y_2 > y_3 > y_4$. Usando el algoritmo de Buchberger, obtenemos:

$G = \{g_1, \dots, g_5\}$, donde $g_1 = x_1 - y_4$, $g_2 = x_2y_4 - y_3$,
$g_3 = x_2y_3^3 - y_1$, $g_4 = y_2 - y_3y_4$, $g_5 = y_1y_4 - y_3^4$.
$G$ es la Base de Gröbner reducida de $K$ con respecto al orden elegido.

y se cumple entonces que

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

como vimos en el lema anterior, se cumple, usando el Lema 3.1, que $g(f_1,\dots,f_m)-g(y_1,\dots,y_m)\in K$ por lo que

$$f-g\in(K+I)=K'$$

y usando la Proposición 7.2 del Capítulo 2, se cumple que $h=N_G(f)=N_G(g)$ está en $k[y_1,\dots,y_m]$.

Inversamente, si $f\xrightarrow{G}_+ h$, con $h\in k[y_1,\dots,y_m]$ entonces se cumple que $f-h\in K'$ por lo que

$$f(x)-h(y)=\sum_{i=1}^{m} g_i(y_i-f_i)+I \quad g_i\in k[y_1,\dots,y_m,x_1,\dots,x_n]$$

Aplicando $\phi$ a ambos lados de la ecuación, tenemos:

$$\begin{aligned} \phi(f(x)-h(y)) &= \phi(\sum_{i=1}^{m} g_i(y_i-f_i))+\phi(I)\\ f(x)-\phi(h) &= \sum_{i=1}^{m} g_i(f_i-f_i)+I\\ f(x)+I &= \phi(h) \quad \text{por lo que } f+I\in im(\phi) \end{aligned}$$

□

EJEMPLO 3.5. Retomando el Ejemplo 1.5, queremos determinar si $x_2^6w+I$ está en la imagen de $\phi$. Construimos una Base de Gröbner para el ideal

$$K'=\langle y_1-x_1^3x_2^4,y_2-x_2^3w^2,y_3-x_1^2w,y_4-x_2w,x_1x_2w-1\rangle$$

con respecto al orden lexicográfico con $x_1>x_2>w>y_1>y_2>y_3>y_4$, obteniendo: $G=\{g_1,\dots,g_9\}$, con

A continuación presentaremos una propiedad importante del núcleo del mapeo $\phi$ que nos permite caracterizar sus elementos.

TEOREMA 3.7. Dado el ideal $K'$ definido como en el teorema anterior se tiene que si $h \in k[y_1, \dots, y_m]$ es tal que $\phi(h) = 0$ entonces $h \in K' \cap k[y_1, \dots, y_m]$.

DEMOSTRACIÓN. Sea $h \in k[y_1, \dots, y_m]$ tal que $\phi(h) = 0$. Entonces $h(f_1, \dots, f_m) \in I$. Si $h(y_1, \dots, y_m) = \sum_{j=1}^{s} h_j y_1^{\alpha_{1j}} \cdots y_m^{\alpha_{mj}}$, entonces tenemos

$$
\begin{aligned}
h(y_1, \dots, y_m) &= h(y_1, \dots, y_m) - h(f_1, \dots, f_m) + h(f_1, \dots, f_m) \\
&= h_1(y_1^{\alpha_{11}} \cdots y_m^{\alpha_{m1}}) + \cdots + h_s(y_1^{\alpha_{1s}} \cdots y_m^{\alpha_{ms}}) \\
&\quad - (h_1(f_1^{\alpha_{11}} \cdots f_m^{\alpha_{m1}}) + \cdots + h_s(f_1^{\alpha_{1s}} \cdots f_m^{\alpha_{ms}})) + h(f_1, \dots, f_m) \\
&= \sum_{j=1}^{s} h_j(y_1^{\alpha_{1j}} \cdots y_m^{\alpha_{mj}} - f_1^{\alpha_{1j}} \cdots f_m^{\alpha_{mj}}) + h(f_1, \dots, f_m)
\end{aligned}
$$

y por el Lema 3.1 $\sum_{j=1}^{s} h_j(y_1^{\alpha_{1j}} \cdots y_m^{\alpha_{mj}} - f_1^{\alpha_{1j}} \cdots f_m^{\alpha_{mj}}) \in K$, por lo que $h(y_1, \dots, y_m) \in (K + I) = K'$. □

## 4. Soluciones Factibles

A partir de lo que hemos visto en las secciones anteriores, contamos ya con las herramientas suficientes para encontrar soluciones factibles, si las hay, dado un problema de programación lineal entera.

**4.1. Problemas con coeficientes enteros no negativos.** Dado el planteamiento polinomial del problema general de programación entera , suponiendo que los coeficientes $a_{ij}$ y $b_i$ son no negativos, y conociendo los resultados del Teorema 3.2, podemos ahora caracterizar los puntos de la región factible del problema.

LEMA 4.1. *Si $x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ pertenece a $im(\phi)$, entonces es la imagen de un monomio $y_1^{\sigma_1} y_2^{\sigma_2} \cdots y_m^{\sigma_m} \in k[y_1, \dots, y_m]$.*

$x_1^{b_1}x_2^{b_2}\cdots x_n^{b_n}$, y entonces

$$\begin{aligned} x_1^{b_1}x_2^{b_2}\cdots x_n^{b_n} &= \phi(y_1^{\sigma_1}\cdots y_m^{\sigma_m}) \\ &= \phi(y_1^{\sigma_1})\phi(y_2^{\sigma_2})\cdots\phi(y_m^{\sigma_m}) \\ &= \phi(y_1)^{\sigma_1}\phi(y_2)^{\sigma_2}\cdots\phi(y_m)^{\sigma_m} \\ &= (x_1^{a_{11}}x_2^{a_{21}}\cdots x_n^{a_{n1}})^{\sigma_1}(x_1^{a_{12}}x_2^{a_{22}}\cdots x_n^{a_{n2}})^{\sigma_2}\cdots(x_1^{a_{1m}}x_2^{a_{2m}}\cdots x_n^{a_{nj}})^{\sigma_m} \end{aligned}$$

lo que, por el planteamiento que hicimos en la Sección 1.1 de este capítulo, indica que $(\sigma_1,\ldots,\sigma_m)$ cumple con las restricciones del problema, por lo que es solución.

Por otro lado, si $(\sigma_1,\ldots,\sigma_m)$ está en la región factible del problema, entonces, por el mismo planteamiento, se cumple que

$$x_1^{b_1}x_2^{b_2}\cdots x_n^{b_n} = \phi(y_1^{\sigma_1}\cdots y_m^{\sigma_m})$$

por lo que $x_1^{b_1}x_2^{b_2}\cdots x_n^{b_n} \in im(\phi)$ □

OBSERVACIÓN 4.3. Si $h$ es la forma normal de $x_1^{b_1}\cdots x_n^{b_n}$ módulo $G$ y $h \notin k[y_1,\ldots,y_m]$ entonces el sistema de restricciónes no tiene soluciones enteras no negativas, es decir, el problema es no factible.

EJEMPLO 4.4. Retomando el ejemplo 3.3, tenemos que

$$x_1^{10}x_2^5 = \phi(y_3^5y_4^5)$$

lo que indica que $(0,0,5,5)$ es solución factible del problema planteado en el Ejemplo 1.1.

**4.2. Problemas con coeficientes enteros.** En este caso, los puntos de la región factible se obtienen de manera muy similar, pero tomando ahora en cuenta el anillo cociente que definimos en la Sección 1.2 y el mapeo $\phi$ que se genera. La caracterización de los puntos de la región factible está dada por:

LEMA 4.5. *Si $x_1^{b'_1}x_2^{b'_2}\cdots x_n^{b'_n}w^\beta + I$ está en la imagen de $\phi$, entonces es la imagen de un monomio $y_1^{\sigma_1}\cdots y_m^{\sigma_m} \in k[y_1,\ldots,y_m]$ si y sólo si el punto $(\sigma_1,\ldots,\sigma_m)$ pertenece a la región factible del problema.*

están en la región factible determinada por el sistema de ecuaciones

$$\begin{cases} 3\sigma_1 - 2\sigma_2 + \sigma_3 - \sigma_4 = -1 \\ \qquad 4\sigma_1 + \sigma_2 - \sigma_3 = 5 \\ \sigma_i \in \mathbb{N}, \ i = 1, 2, 3, 4. \end{cases}$$

## 5. Optimización de la Función Objetivo

Ahora que conocemos cuáles son los puntos de la región factible del problema planteado, la pregunta importante es: ¿Cómo minimizamos la función objetivo sobre la región factible? Lo que haremos es construir un orden adaptado al problema que nos proporcione la solución.

DEFINICIÓN 5.1. Decimos que un orden monomial $<_c$ en $k[y_1, \ldots, y_m, x_1, \ldots, x_n]$ es *un orden adaptado a un problema de programación lineal entera*, si cumple que

1. Es un orden por eliminación.
2. Es compatible con la función objetivo $c$ y el mapeo $\phi$, es decir

$$\begin{aligned} Si \ \ & \phi(y_1^{\sigma_1} \cdots y_m^{\sigma_m}) = \phi(y_1^{\sigma'_1} \cdots y_m^{\sigma'_m}) \\ y \ \ & c(\sigma_1, \ldots, \sigma_m) > c(\sigma'_1, \ldots, \sigma'_m) \end{aligned}$$

$$entonces \ \ y_1^{\sigma_1} \cdots y_m^{\sigma_m} >_c y_1^{\sigma'_1} \cdots y_m^{\sigma'_m}$$

Los órdenes monomiales que cumplen con estas propiedades son aquéllos que proporcionan las soluciones a costo mínimo del sistema de restricciones, como lo muestra el siguiente teorema.

TEOREMA 5.2. Dados un problema de programación lineal entera en su forma estándar, un orden monomial $<_c$ adaptado al problema, el planteamiento polinomial del problema y el mapeo $\phi$ correspondiente, construimos
$K = \langle y_j - x_1^{a_{1j}} x_2^{a_{2j}} \cdots x_n^{a_{nj}} \,|\, j = 1, \ldots, m\rangle$ y $G$ Base de Gröbner de $K$ con respecto a $<_c$. Si $f = x_1^{b'_1} x_2^{b'_2} \cdots x_n^{b'_n} w^{\beta} \in im(\phi)$ entonces la forma

Dado $K'$ construimos una Base de Gröbner para él, estableciendo lo siguiente:

i las variables $x$ son mayores que las $y$

ii $x_1 > x_2 > w$

iii $y_1^{\sigma_1} \cdots y_m^{\sigma_m} < y_1^{\sigma'_1} \cdots y_m^{\sigma'_m}$ si y sólo si
$1000\sigma_1 + \sigma_2 + \sigma_3 + 100\sigma_4 < 1000\sigma'_1 + \sigma'_2 + \sigma'_3 + 100\sigma'_4$

iv en caso de "empate"con respecto a la funcón de costo, se ordenan las $y$ con el orden lexicográfico

La Base de Gröbner reducida para $K'$ es entonces

$$G = \{w - y_2^2 y_4^3,\ y_4 - y_2 y_3,\ x_1 - y_1 y_2^6 y_3^{10},\ x_2 - y_1 y_2^6 y_3^9,\ y_1 y_2^7 y_3^{11} - 1\}$$

Tenemos que

$$x_2^6 w \xrightarrow{G}_+ y_1 y_2^3 y_3^2$$

lo que nos da la solución óptima $(1, 3, 2, 0)$.

# Bibliografía


[AL94] William W. Adams and Philippe Loustanau. *An Introduction to Grobner bases.* American Mathematical Society, Providence, Rhode Island, 1994.

[Bal69] E. Balas. The intersection cut- a new cutting plane for integer programming. *Management Sciences Research Report, Carnegie-Mellon University*, 187, 1969.

[CLO96] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties and Algorithms: an Introduction to Computational Algebraic Geometry and Commutative Algebra.* Undergraduate texts in mathematics. Springer, New York, New York, 1996.

[CLO98] David Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry*, volume 185 of *Graduate Texts in Mathematics.* Springer, New York, New York, 1998.

[Coo71] S. Cook. The complexity of theorem proving procedures. *Third Annual ACM Symposium on Theory of Computing*, pages p.151–158, 1971.

[Dak65] R. Dakin. A tree search algorithm for mixed integer programming problems. *Computer Journal*, 8:p.250–255, 1965.

[Dan59] G. Dantzig. Note on solving linear problems in integers. *Naval Research Logistics Quaterly*, 6:p.75–76, 1959.

[GH63] R. Gomory and A. Hoffman. On the convergence of an integer programming process. *Naval Research Logistics Quaterly*, 10:p.121–123, 1963.

[Gom58] R. Gomory. An algorithm for integer solutions to linear programs. *Princeton IBM Mathematical Research Report*, November, 1958.

[Kee72] R. Karp, R. Miller (ed.), and J. Thatcher (ed.). *Complexity of Computer Computations.* Plenum Press, New York, New York, 1972. Reducibilities Among Combinatorial Problems, p.85-103.

[Man89] Udi Manber. *Introduction to Algorithms. A Creative Approach.* Addison-Wesley, 1989.

[Sak84] Michel Sakarovitch. *Optimization Combinatoire: Methodes Mathematiques et ALgorithmiques. Programmation Discrete.* Hermann, Paris, 1984.