

**INSTITUTO POLITECNICO NACIONAL**  
**ESCUELA SUPERIOR DE FISICA Y MATEMATICAS**

APLICACION DE LAS BASES DE GRÖBNER PARA LA  
RESOLUCION DE PROBLEMAS DE PROGRAMACION ENTERA

**T E S I S**  
QUE PARA OBTENER EL TITULO DE:  
**I N G E N I E R O M A T E M A T I C O**

P R E S E N T A:

**MIGUEL ANGEL POLO VUELVAS**

ASESOR: DR. ADRIAN ALCANTAR TORRES.

CON APOYO DE PROYECTO CONACyT No. 45256.

Abril de 2004.

NO PUEDE EXISTIR UN LENGUAJE MÁS UNIVERSAL Y SIMPLE, MÁS CARENTE DE ERRORES Y OSCURIDADES Y POR LO TANTO MÁS APTO PARA EXPRESAR LAS RELACIONES INVARIABLES DE LAS COSAS NATURALES [...]. [LAS MATEMÁTICAS] PARECEN CONSTITUIR UNA FACULTAD DE LA MENTE HUMANA DESTINADA A COMPENSAR LA BREVEDAD DE LA VIDA Y LA IMPERFECCIÓN DE LOS SENTIDOS.

Joseph Fourier,  
*Théorie analytique de la chaleur:*  
Discurso preliminar (1822)

# Agradecimientos

Toda mi vida, mi carácter, mis cualidades, mis defectos, . . . , todo lo que soy, se lo debo a mis padres. Este trabajo es para ustedes.

Agradezco infinitamente al profesor Adrián Alcántar Torres por la enorme paciencia que ha tenido para apoyarme en la consecución de esta meta.

A todos los profesores con los que compartí esos ratos de aprendizaje, en especial, al profesor Víctor Hugo Ibarra Mercado, quien con su ejemplo y forma de ser, ha logrado motivar en mí el deseo aún más profundo de ser simplemente el mejor.

No puedo dejar de lado los divertidos momentos compartidos con mis amigos; los que fueron de ocio, los que fueron de tensión, los que fueron de estudio, los que fueron de desvelo y los que fueron de baile; esos pedazos de tiempo, los llevo siempre en mi corazón, así como sus nombres: Gaby, Janet, Iris, Eva, Juan José, Noé, Adal, Maggy, las chicas del coro, los de la banda; gracias por estar aquí.

# Prefacio

La optimización se ha convertido en una necesidad para un amplio sector de la sociedad. Problemas económicos, biológicos, físicos, químicos, incluso sociales, no podrían ser resueltos o manipulados, si no existiera una herramienta adecuada para tal efecto.

La búsqueda de tal herramienta, ha llevado a muchos científicos e ingenieros a desarrollar instrumentos, métodos u objetos matemáticos que ayuden a ampliar los conocimientos sobre el tema y a resolver problemas que antes eran irresolubles.

Así mismo, la comunicación, divulgación y exposición de estos temas, y de la ciencia en general, ha contribuido enormemente para que cada vez más personas nos intereseamos por descubrir, aprender, experimentar e inventar.

El presente trabajo, es una investigación documental realizada a merced del interés que despierta en mí la Optimización Lineal; ha sido realizado con la finalidad de dar a conocer una herramienta matemática, y su respectivo apoyo computacional, que facilita y— a mi consideración— agiliza la solución de problemas de Programación Entera. De igual modo, expongo las bases teóricas, así como las características principales de funcionamiento de este método matemático.

Todo esto considerando que en la carrera de Ingeniería Matemática, la Investigación de Operaciones constituye la piedra angular para aplicar todo lo aprendido; y tener un instrumento que nos apoye en rapidez y factibilidad de los procesos de análisis y solución de problemas, es primordial para, de igual forma, optimizar el tiempo dedicado a estos problemas.

La estructura de esta tesis es como sigue:

**Capítulo 1.** Se expone lo que es la optimización lineal, su instrumento de desarrollo (el método símplex), el apoyo teórico, el concepto de dualidad y los problemas más frecuentes que se presentan en la aproximación de programas lineales a soluciones enteras.

- Capítulo 2.** Trata de cómo el algoritmo de la división puede aplicarse a polinomios de varias variables con la aplicación de los órdenes monomiales.
- Capítulo 3.** Aquí se presenta el instrumento teórico más importante de este trabajo, las Bases de Gröbner y el algoritmo de Buchberger.
- Capítulo 4.** El desarrollo de la solución de problemas de Programación Entera aplicando la traslación de un sistema de ecuaciones a un problema de polinomios, es el punto medular de esta tesis y se presenta en este capítulo.
- Capítulo 5.** Es un resumen de los comandos y estructuras básicas que se encuentran en el programa computacional *Macaulay 2*, usado para manipular los objetos matemáticos que nos permiten resolver problemas de Programación Entera.

# Índice general

<b>1. Antecedentes</b>	<b>2</b>
1.1. Programación lineal . . . . .	2
1.2. El algoritmo símplex . . . . .	3
1.2.1. La forma canónica y las variables básicas . . . . .	4
1.2.2. Mejorando una solución básica factible no óptima . . . . .	6
1.2.3. Teoría detrás del algoritmo símplex . . . . .	10
1.3. Método símplex . . . . .	13
1.3.1. El método . . . . .	14
1.3.2. Teoría detrás de la Fase I . . . . .	16
1.4. Variables acotadas . . . . .	17
1.5. Método símplex revisado . . . . .	19
1.5.1. Motivación. . . . .	19
1.6. Dualidad en programación lineal . . . . .	27
1.7. Desventajas del símplex en la programación entera . . . . .	31
<b>2. Algoritmo de la división</b>	<b>34</b>
2.1. Algoritmo de la división en $R[x]$ . . . . .	34
2.2. Ordenes monomiales . . . . .	36
2.3. Algoritmo de la división en $k[x_1, \dots, x_n]$ . . . . .	38
<b>3. Bases de Gröbner</b>	<b>42</b>
3.1. Preliminares . . . . .	42
3.2. S-Polinomios y el algoritmo de Buchberger . . . . .	45
3.3. Bases de Gröbner reducidas . . . . .	46
<b>4. Programación lineal entera</b>	<b>48</b>
4.1. Programación entera . . . . .	48
4.1.1. Caso particular: todas las constantes positivas . . . . .	49

ÍNDICE GENERAL

VI

4.1.2. Caso general: algunas constantes negativas . . . . .	56
4.1.3. Ejemplos de problemas generales . . . . .	59
4.2. Programación entera y combinatoria . . . . .	63
<b>5. El entorno de <i>Macaulay 2</i></b>	<b>73</b>
5.1. Panorama histórico . . . . .	73
5.2. Estructuras, comandos y rutinas básicas . . . . .	74

# Introducción

La programación lineal, gracias a su elevado nivel de eficiencia computacional, es la base para el desarrollo de algoritmos de solución de otros tipos más complejos de modelos de Investigación de Operaciones, incluyendo la programación entera.

Sin embargo, y considerando los métodos y algoritmos que se han aplicado para resolver problemas de programación entera, se ha catalogado a tales problemas como *NP-Complejos*.

Además, debido a que sólo existen métodos particulares para resolver ciertos tipos de problemas de programación entera, no hay un algoritmo general apto y funcional para cualquier clase de problemas de programación entera.

Es por ello, que surge la necesidad de abordar e investigar otros campos y áreas de la matemática con el propósito de encontrar dicho algoritmo general, o en su defecto, avanzar en el desarrollo de instrumentos matemáticos cada vez más eficientes.

Así pues, la aplicación de las bases de Gröbner para la resolución de problemas de programación entera, constituye una herramienta con ciertas ventajas sobre los métodos particulares de programación entera, entre las cuales está el poder aplicar el método a problemas más generales y la rapidez con que estos son resueltos.



# Capítulo 1

## Antecedentes

En este capítulo se dará una explicación detallada sobre lo que es la programación u optimización lineal, el método símplex y el concepto de dualidad, tomándose en su conjunto como punto de partida en cuanto a técnica de resolución de problemas de programación entera, además de las limitaciones que presenta el trabajar estos problemas cuando se intenta aproximar una solución entera basándose en una solución lineal.

### 1.1. Programación lineal

La programación lineal es una técnica de modelado matemático, diseñada para optimizar el empleo de recursos limitados. Se aplica exitosamente en el ejército, la agricultura, la industria, la transportación, la economía, los sistemas de salud, e incluso, en las ciencias conductuales y sociales. La utilidad de la técnica se incrementa mediante la disponibilidad de programas de computadora altamente eficientes.

El método gráfico, que no desarrollaremos aquí, nos permite ver que la solución óptima de programación lineal siempre está asociada con un punto de esquina, conocido matemáticamente como punto extremo del “espacio solución”. Este resultado es la idea clave para el desarrollo del *método símplex* para resolver cualquier modelo de programación lineal.

## 1.2. El algoritmo símplex

La transición del punto extremo geométrico de la solución de un problema de programación lineal al método símplex radica en identificar algebraicamente los puntos extremos. Para esto, primero convertimos el modelo a la forma estándar de programación lineal, utilizando variables de holgura o de superávit, para convertir las restricciones de desigualdad en ecuaciones.

Cabe señalar, que al agregar variables de holgura, el problema adquiere mayor dificultad para resolverse en cuanto a tiempo y manejo de datos adicionales, sin embargo, ese “costo” es aceptable, ya que al convertirse en ecuaciones y conformar así un sistema de ecuaciones lineales, se puede usar la herramienta del álgebra lineal, que manejamos eficientemente, para resolverlo.

El interés en la forma estándar de un modelo de programación lineal, radica en las soluciones básicas de las ecuaciones lineales simultáneas. Esta solución básica algebraica define completamente todos los puntos extremos del espacio de la solución. El método símplex está diseñado para localizar de manera eficiente, la óptima entre estas soluciones básicas.

La forma estándar tiene las siguientes propiedades:

1. Todas las restricciones (con excepción de las restricciones de no negatividad sobre las variables) son ecuaciones con el lado derecho no negativo.
2. Todas las variables son no negativas.
3. La función objetivo puede ser del tipo de maximización o de minimización.

Así, la forma estándar de un problema de programación lineal puede verse como:

(Minimizar, Maximizar):  $c_1x_1 + \cdots + c_nx_n$ , sujeto a:

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
 a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
 &\vdots \\
 a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \\
 x_i &\geq 0, \quad i = 1, \dots, n.
 \end{aligned}
 \tag{1.1}$$

A fin de poder explicar el método símplex desarrollado por Dantzig, reescribiremos la forma estándar en términos de un sistema de ecuaciones lineales de la siguiente forma:

$$\begin{aligned}
 c_1x_1 + c_2x_2 + \cdots + c_nx_n &= z \quad (Min) \\
 a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
 a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
 &\vdots \\
 a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \\
 x_i &\geq 0, \quad i = 1, \dots, n.
 \end{aligned} \tag{1.2}$$

### 1.2.1. La forma canónica y las variables básicas

El método símplex encuentra una solución óptima (o determina que no existe) mediante una secuencia de pivoteos en el sistema de ecuaciones original en la forma estándar (1.2). Por ejemplo, consideremos el problema de minimizar  $z$  para  $x_j \geq 0$  donde

$$\begin{aligned}
 2x_1 + 2x_2 + 2x_3 + x_4 + 4x_5 &= z \\
 4x_1 + 2x_2 + 13x_3 + \mathbf{3x_4} + x_5 &= 17 \\
 x_1 + x_2 + 5x_3 + x_4 + x_5 &= 7
 \end{aligned} \tag{1.3}$$

Un pivoteo consiste en tomar algún elemento no cero (llamado *pivote*) en el arreglo tal como  $\mathbf{3x_4}$  y usarlo para eliminar  $x_4$  de las ecuaciones restantes dividiendo primero esta ecuación por 3, obteniendo:

$$\begin{aligned}
 2x_1/3 + 4x_2/3 - 7x_3/3 + 11x_5/3 &= z - 17/3 \\
 4x_1/3 + 2x_2/3 + 13x_3/3 + x_4 + x_5/3 &= 17/3 \\
 -x_1/3 + \mathbf{x_2/3} + 2x_3 + 2x_5/3 &= 4/3
 \end{aligned} \tag{1.4}$$

Si ahora pivoteamos tomando  $\mathbf{x_2/3}$  como pivote obtenemos:

$$\begin{aligned}
 2x_1 - 5x_3 + x_5 &= z - 11 \\
 2x_1 + 3x_3 + x_4 - x_5 &= 3 \\
 -x_1 + x_2 + 2x_3 + 2x_5 &= 4
 \end{aligned} \tag{1.5}$$

Vemos que el sistema original (1.3) es equivalente a (1.5) porque tienen el mismo conjunto solución. Reescribiendo (1.5) obtenemos:

$$\begin{aligned}
 (-z) + 2x_1 - 5x_3 + x_5 &= -11 \\
 2x_1 + 3x_3 + x_4 - x_5 &= 3 \\
 -x_1 + x_2 + 2x_3 + 2x_5 &= 4
 \end{aligned} \tag{1.6}$$

Los pasos detallados para pivotar en un término  $a_{rs}x_s$ , donde  $a_{rs} \neq 0$ , son los siguientes:

1. Reemplace la  $r$ -ésima ecuación por ella misma multiplicada por  $(1/a_{rs})$ .
2. Para cada  $i = 1, \dots, m$  excepto  $i = r$ , reemplace la  $i$ -ésima ecuación por la suma de la  $i$ -ésima ecuación y la nueva  $r$ -ésima ecuación (obtenida en el paso anterior) multiplicada por  $(-a_{is})$ .

**Definición 1.2.1 (Forma canónica)** *Un sistema de  $m$  ecuaciones con  $n$  variables  $x_j$  se dice que está en forma canónica con respecto a un conjunto ordenado de variables  $(x_{j_1}, x_{j_2}, \dots, x_{j_m})$  si y sólo si  $x_{j_i}$  tiene un coeficiente unitario en la ecuación  $i$  y un coeficiente cero en todas las otras.*

El sistema (1.6) está en *forma canónica* con respecto a las variables  $(-z), x_4, x_2$ , que son llamadas las *variables básicas*, y son expresadas en términos de las *variables no básicas*. En la práctica,  $z$  se refiere a la *variable objetivo*. La *solución básica* factible se obtiene cuando los valores de las variables no básicas son cero. En (1.6), al hacer esto obtenemos:

$$z = 11, x_B = (x_4, x_2) = (3, 4), x_N = (x_1, x_3, x_5) = (0, 0, 0). \quad (1.7)$$

**Definición 1.2.2** *Una solución básica es degenerada si el valor de una o más de las variables básicas es cero.*

El Algoritmo Símplex siempre es iniciado con un sistema de ecuaciones en forma canónica con respecto a algún conjunto ordenado de variables básicas. Por ejemplo, supongamos que tenemos el sistema canónico (1.8) siguiente con variables básicas  $(-z), x_1, x_2, \dots, x_m$ . Este sistema de  $m + 1$  ecuaciones y  $n + 1$  variables es equivalente a la forma estándar (1.2). Nuestro problema es encontrar valores de  $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$ , y  $z$  mínimo satisfaciendo:

$$\begin{array}{rcccc} -z & & + \bar{c}_{m+1}x_{m+1} & + \dots + \bar{c}_jx_j & + \dots + \bar{c}_nx_n & = & -\bar{z}_0 \\ x_1 & & + \bar{a}_{1,m+1}x_{m+1} & + \dots + \bar{a}_{1j}x_j & + \dots + \bar{a}_{1n}x_n & = & \bar{b}_1 \\ x_2 & & + \bar{a}_{2,m+1}x_{m+1} & + \dots + \bar{a}_{2j}x_j & + \dots + \bar{a}_{2n}x_n & = & \bar{b}_2 \\ \vdots & & \vdots & & \vdots & & \vdots \\ x_m & + \bar{a}_{m,m+1}x_{m+1} & + \dots + \bar{a}_{mj}x_j & + \dots + \bar{a}_{mn}x_n & = & \bar{b}_m \end{array} \quad (1.8)$$

donde  $\bar{a}_{ij}, \bar{c}_j, \bar{b}_i, \bar{z}_0$  son constantes.

En notación matricial, el mismo sistema puede ser escrito compactamente como:

$$\begin{pmatrix} 1 & 0 & \bar{c} \\ 0 & I & \bar{A} \end{pmatrix} \begin{pmatrix} -z \\ x_B \\ x_N \end{pmatrix} = \begin{pmatrix} -\bar{z}_0 \\ \bar{b} \end{pmatrix} \quad (1.9)$$

donde  $x_B = (x_1, x_2, \dots, x_m)^T$  y  $x_N = (x_{m+1}, x_{m+2}, \dots, x_n)^T$ . En esta forma canónica, la solución básica es:

$$z = \bar{z}_0, \quad x_B = \bar{b}, \quad x_N = 0. \quad (1.10)$$

En el Algoritmo Símplex se requiere que ésta solución básica inicial sea factible, por lo que necesitamos que:

$$\bar{x} = \bar{b} \geq 0. \quad (1.11)$$

### 1.2.2. Mejorando una solución básica factible no óptima

Dada la forma canónica, es fácil obtener la solución básica asociada. También es fácil determinar si la solución básica (1.10) es o no factible; y si es factible, se puede determinar si es o no óptima.

Continuando con nuestro ejemplo (1.3), reescribimos (1.6) resaltando  $3x_3$ :

$$\begin{array}{rcccccccl} (-z) & + & 2x_1 & & - & 5x_3 & & + & x_5 & = & -11 \\ & & + & 2x_1 & & + & \mathbf{3x_3} & + & x_4 & - & x_5 & = & 3 \\ & - & x_1 & + & x_2 & + & 2x_3 & & + & 2x_5 & = & 4 \end{array} \quad (1.12)$$

El término resaltado será usado después para mejorar la solución. La solución básica factible de (1.12) puede ser escrita como:

$$z = 11, \quad x_B = (x_4, x_2) = (3, 4), \quad x_N = (x_1, x_3, x_5) = (0, 0, 0). \quad (1.13)$$

**Definición 1.2.3** *Los coeficientes  $\bar{c}_j$  en la función objetivo del sistema canónico (1.8) son llamados factores relativos de costos o también costos reducidos asociados con un conjunto básico de variables.*

Un factor relativo de costos en la forma canónica (1.12) es negativo,  $\bar{c}_3 = -5$ , que es el coeficiente de  $x_3$ . Si  $x_3$  se incrementa a algún valor positivo manteniendo en cero los valores de las otras variables no básicas y ajustando

las variables básicas para que satisfagan las ecuaciones, es evidente que el valor de  $z$  se reducirá, porque el valor correspondiente de  $z$  está dado por:

$$z = 11 - 5x_3 \quad (1.14)$$

Parece razonable hacer  $x_3$  tan grande como sea posible, ya que entre más grande sea el valor de  $x_3$ , menor será el valor de  $z$ . Sin embargo, en este caso, el valor de  $x_3$  no puede ser incrementado indefinidamente mientras las otras variables no básicas permanecen en cero, porque los valores correspondientes de las variables básicas que satisfacen (1.12) son:

$$\begin{aligned} x_4 &= 3 - 3x_3 \\ x_2 &= 4 - 2x_3 \end{aligned} \quad (1.15)$$

Vemos que si  $x_3$  se incrementa más de  $3/3$ , entonces  $x_4$  será negativo, y que si se incrementa más de  $4/2$  entonces  $x_2$  también será negativo. Obviamente, el valor más grande de  $x_3$  permitido es el menor de ellos, o sea  $x_3 = 1$ , que sustituyéndose en (1.14) y (1.15) retorna una nueva solución factible (básica) con menor costo:

$$z = 6, \quad x_3 = 1, \quad x_2 = 2, \quad x_1 = x_4 = x_5 = 0. \quad (1.16)$$

Esto se ha hecho mediante el pivoteo sobre el término en negritas del sistema (1.12). Así, la nueva forma canónica relativa a  $(-z), x_3, x_2$  está dada por:

$$\begin{aligned} (-z) &+ \frac{16}{3}x_1 && + \frac{5}{3}x_4 - \frac{2}{3}x_5 &= -6 \\ &+ \frac{2}{3}x_1 &+ x_3 &+ \frac{1}{3}x_4 - \frac{1}{3}x_5 &= 1 \\ &- \frac{7}{3}x_1 + x_2 && - \frac{2}{3}x_4 + \frac{8}{3}x_5 &= 2 \end{aligned} \quad (1.17)$$

Ya que el conjunto solución de las formas canónicas antes y después del pivoteo son los mismos, los valores de  $x_2, x_3$  son determinados únicamente cuando  $(x_1, x_4, x_5) = 0$ , también obtenidos vía (1.15) o por inspección de (1.17). El resultado está dado por (1.16).

Esto da una nueva solución básica factible con  $z = 6$ . Sin embargo, el valor de  $z$  puede reducirse, esto es observable ya que  $\bar{c}_5 = -2/3$ . Así que, como antes, el coeficiente  $\bar{c}_5 = -2/3$  junto con el hecho de que la solución básica es no degenerada, indican que la solución todavía no es mínima y que puede obtenerse una mejor solución dejando las otras variables no básicas,

$x_1 = x_4 = 0$  y resolviendo para los nuevos valores de  $x_2, x_3, z$  en términos de  $x_5$ :

$$\begin{aligned} -z &= -6 + \frac{2}{3}x_5 \\ x_3 &= 1 + \frac{1}{3}x_5 \\ x_2 &= 2 - \frac{8}{3}x_5 \end{aligned} \tag{1.18}$$

Enseguida incrementamos  $x_5$  al máximo valor posible mientras  $x_3, x_2$  sean no negativas. Nótese que la segunda relación en (1.18) no acota el incremento de  $x_5$ , pero que la tercera, restringe  $x_5$  a un máximo de  $(\frac{3}{4})$  con el cual el valor de  $x_2$  se reduce a cero. Ahora, el término pivote  $\frac{8}{3}x_5$  en la tercera ecuación de (1.17) es usado para la siguiente eliminación. Ya que el valor de  $x_2$  se ha transformado en cero y  $x_5$  es positivo, el nuevo conjunto de variables básicas es  $x_3, x_5$ . Reduciendo el sistema (1.17) a la forma canónica relativa a  $x_3, x_5, (-z)$  tenemos:

$$\begin{aligned} (-z) &+ \frac{19}{4}x_1 + \frac{1}{4}x_2 + \frac{3}{2}x_4 = -\frac{11}{2} \\ &+ \frac{3}{8}x_1 + \frac{1}{8}x_2 + x_3 + \frac{1}{4}x_4 = \frac{5}{4} \\ &- \frac{7}{8}x_1 + \frac{3}{8}x_2 - \frac{1}{4}x_4 + x_5 = \frac{3}{4} \end{aligned} \tag{1.19}$$

y la solución básica factible:

$$z = \frac{11}{2}, \quad x_3 = \frac{5}{4}, \quad x_5 = \frac{3}{4}, \quad x_1 = x_4 = x_2 = 0. \tag{1.20}$$

Ya que todos los factores de costos relativos para las variables no básicas son ahora positivos, esta solución es mínima. Más aún, es la única solución mínima porque todos los factores de costos relativos son estrictamente positivos. Nótese que se realizaron dos pivoteos en nuestro sistema canónico inicial (1.12) para encontrar este óptimo.

**Algoritmo 1.2.4 (Algoritmo Símplex)** Supóngase que un programa lineal en forma estándar ha sido convertido a la forma canónica factible:

$$\begin{aligned} (-z) + 0x_B + \bar{c}^T x_N &= -\bar{z}_0 \\ Ix_B + Ax_N &= \bar{b} \end{aligned} \tag{1.21}$$

que se mostró antes en (1.8). Entonces la solución básica factible es:

$$x_B = \bar{b} \geq 0, \quad x_N = 0, \quad z = \bar{z}_0$$

Los pasos del algoritmo son los siguientes:

1. *Menor costo reducido.* Tomar

$$\bar{c}_s = \min_j \bar{c}_j \quad (1.22)$$

2. *Prueba de optimalidad.* Si  $\bar{c}_s \geq 0$ , regresar la solución básica factible como óptima y parar.
3. *Variable entrante.* Si  $\bar{c}_s < 0$ , entonces  $s$  es el subíndice de la variable básica entrante.
4. *Prueba para  $z$  no acotada.* Si  $\bar{A}_{os} \leq 0$  regresar la clase de soluciones factibles  $x_B = \bar{b} - A_{os}x_s$ ,  $x_j = 0$ ,  $j$  no básica y  $j \neq s$ , y  $z = \bar{z}_0 + \bar{c}_s x_s$  tal que  $z \rightarrow -\infty$  cuando  $x_s \rightarrow \infty$  y parar. Esto requiere regresar la solución básica factible, la columna entrante subíndice  $s$  y la columna  $\bar{A}_{os}$ .
5. *Variable saliente.* Tomar la variable básica saliente  $x_{j_r}$ , y el valor de  $\bar{x}_s$ , la variable entrante, como:

$$\bar{x}_s = \frac{\bar{b}_r}{\bar{a}_{rs}} = \min_{\{i|\bar{a}_{is}>0\}} \frac{\bar{b}_i}{\bar{a}_{is}} \geq 0. \quad (1.23)$$

En caso de empate, sea  $\mathcal{R}$  el conjunto de filas  $k$  empatadas:

$$\mathcal{R} = \left\{ k \mid \frac{\bar{b}_k}{\bar{a}_{ks}} \leq \frac{\bar{b}_i}{\bar{a}_{is}}, \bar{b}_i \geq 0, \bar{a}_{is} > 0, \bar{a}_{ks} > 0, i = 1, \dots, m \right\} \quad (1.24)$$

*CASO NO DEGENERADO:* Si  $\bar{b}_k > 0$  para todo  $k \in \mathcal{R}$ , la elección de  $r$  entre los empates es arbitraria.

*CASO DEGENERADO:* Si  $\bar{b}_k = 0$  para más de un  $k \in \mathcal{R}$ , puede usarse la regla de elección aleatoria, esto es, elegir  $r$  al azar (con igual probabilidad).

6. Pivotear en  $\bar{a}_{rs}$  para determinar una nueva solución básica factible, hacer  $j_r = s$  y regresar al paso 1. Nótese que el paso pivote se hace ignorando si el valor de  $z$  decrece o no.



### 1.2.3. Teoría detrás del algoritmo símplex

En esta sección discutiremos los detalles técnicos detrás del algoritmo símplex.

**Teorema 1.2.5 (Optimalidad)** *Una solución básica factible es una solución mínima factible con costo total  $\bar{z}_0$  si todos los factores de costos relativos son no negativos:*

$$\bar{c}_j \geq 0 \quad \text{para } j = 1, \dots, n \quad (1.25)$$

**Demostración.** Refiriéndonos a la forma canónica (1.8), es obvio que si los coeficientes de la fórmula de costos modificada son todos positivos o cero, el menor valor de  $\sum \bar{c}_j x_j$  es mayor o igual a cero, sea cual sea la elección de  $x_j$  no negativo. Así,  $z \geq \bar{z}_0$  para todas las elecciones factibles de  $x$ . En el caso particular de la solución básica factible (1.10), sin embargo, tenemos  $z = \bar{z}_0$ ; así que  $\min z = \bar{z}_0$  y la solución es óptima. ■

Esta prueba muestra que para todas las soluciones  $x_j \geq 0$  que satisfacen la forma canónica (1.8), la solución básica tiene el menor valor de  $z$ . También muestra que para todas las soluciones que satisfacen el sistema original (1.2), la solución básica es óptima porque el sistema original (1.2) y (1.8) son equivalentes, esto es, tienen el mismo conjunto de soluciones factibles. El inverso del Teorema 1.2.5 es verdadero sólo si el programa lineal es no degenerado.

**Teorema 1.2.6 (Óptimo único)** *Una solución básica factible es la única solución mínima factible si  $\bar{c}_j > 0$  para todas las variables no básicas.*

Como hemos observado en el ejemplo numérico, la forma canónica provee un criterio sencillo para comprobar la optimalidad de una solución básica factible. Además, si el criterio no se satisface, se genera otra solución por pivoteo que reduce el valor de la función objetivo (excepto para ciertos casos degenerados).

Ahora formalizaremos este procedimiento de mejorar una solución básica factible no óptima. En general, si por lo menos un factor de costo relativo  $\bar{c}_j$  en la forma canónica (1.8) es negativo, es posible, suponiendo que no hay degeneración, construir una nueva solución básica factible con un valor objetivo menor que  $z = \bar{z}_0$ . La solución del menor valor objetivo es obtenida incrementando el valor de una de las variables no básicas  $x_s$  para  $\bar{c}_s < 0$  y ajustando los valores de las variables básicas en consecuencia.

Qué  $x_s$  escoger cuando hay distintos  $\bar{c}_j < 0$  ha sido tema de mucho estudio. Una regla comúnmente usada es tomar el  $j = s$  que da el máximo decrecimiento de  $z$  por unidad de incremento de una variable no básica  $x_j$ . Este criterio es usado comunmente en la práctica porque es típico que lleven menos iteraciones que usando cualquier  $j = s$  tal que  $\bar{c}_j < 0$ . Usando la forma canónica (1.8), construimos una solución en que  $x_s$  tome algún valor positivo; los valores de todas las otras variables no básicas,  $x_j, j \neq s$ , son mantenidos temporalmente en cero; y los valores de  $z$  y las variables básicas  $x_B$ , cuyos índices (denotados por el subíndice B) son  $j_1, \dots, j_m$ , son ajustados para tener cuidado del incremento de  $x_s$ :

$$\begin{aligned} z &= \bar{z}_0 + \bar{c}_s x_s \\ x_B &= \bar{b} - \bar{A}_{os} x_s \end{aligned} \quad (1.26)$$

donde  $\bar{c}_s < 0$ . Ya que  $\bar{c}_s$  es negativo, es claro que podemos hacer  $z$  tan pequeño como sea posible haciendo  $x_s$  lo más grande posible. Sin embargo, tenemos que retener la factibilidad, y así la única cosa que nos impide hacer  $x_s$  infinitamente grande es la posibilidad de que el valor de una de las variables básicas en (1.26) llegara a ser negativo. Es preciso decir que si todos los componentes de  $\bar{A}_{os}$  son no positivos, entonces  $x_s$  puede hacerse arbitrariamente grande sin violar factibilidad. Esto establece:

**Teorema 1.2.7 (Programa lineal no acotado)** *Si en el sistema canónico, para algún  $s$ , todos los coeficientes  $\bar{a}_{is}$  son no positivos y  $\bar{c}_s$  es negativo, entonces puede construirse una clase de soluciones factibles donde el conjunto de valores de  $z$  no tiene cota inferior: es decir,  $z = z_0 + \bar{c}_s x_s$  y  $x_B = \bar{b} - \bar{A}_{os} x_s \geq 0$  donde  $x_s \rightarrow \infty, x_j = 0$  para todo  $j \neq s$  no básico.*

Por otro lado, si al menos un  $\bar{a}_{is}$  es positivo, no será posible incrementar el valor de  $x_s$  indefinidamente, porque de (1.26), siempre que para este  $i$ ,  $x_s > \bar{b}_i / \bar{a}_{is}$ , el valor de  $x_j$  será negativo. Para mantener factibilidad, podemos sólo incrementar  $x_s$  a la menor razón de  $\bar{b}_i / \bar{a}_{is}$  para todos los  $\bar{a}_{is}$  positivos. Esto es:

$$\bar{x}_s = \frac{\bar{b}_r}{\bar{a}_{rs}} = \min_{\{i|\bar{a}_{is}>0\}} \frac{\bar{b}_i}{\bar{a}_{is}} \geq 0, \quad (\bar{a}_{rs} > 0). \quad (1.27)$$

donde es particularmente notable que sólo esos  $i, r$  son considerados para que  $\bar{a}_{is} > 0$  y  $\bar{a}_{rs} > 0$ .

En general, si  $\bar{x}_s = 0$  en (1.27), uno o más  $\bar{b}_i$  son cero (i.e., la solución básica factible es degenerada); más aun, de (1.26), el valor  $z$  no decrecerá durante tal iteración. Si no hay decrecimiento, existe la posibilidad de ciclo. Para

evitarlo, podemos escoger  $r$  al azar de aquellos  $i$  tal que  $\bar{a}_{is} > 0$  y  $\bar{b}_i = 0$ . Por ejemplo, si  $\bar{a}_{1s} > 0$  y  $\bar{a}_{2s} > 0$  pero  $\bar{b}_1 = \bar{b}_2 = 0$ , arrojaremos una moneda para decidir si  $r = 1$  o  $r = 2$ . Bajo esta regla de elección aleatoria puede probarse que el algoritmo casi seguramente terminará en un número finito de iteraciones.

Si la solución básica es no degenerada tenemos:

**Teorema 1.2.8 (Decrecimiento bajo no degeneración)** *Si en el sistema canónico, la solución básica es no degenerada y un factor de costo relativo  $\bar{c}_s$  es negativo para algún  $s$  y para este  $s$  al menos un coeficiente  $\bar{a}_{is}$  es positivo, entonces la solución básica factible no degenerada puede ser modificada en una nueva solución básica factible con un menor costo total  $z$ .*

Específicamente, ahora debemos ver que el reemplazo de  $x_{j_r}$  por  $x_s$  en el conjunto de variables básicas  $x_{j_1}, \dots, x_{j_m}$  resulta en un nuevo conjunto que es básico y una solución básica correspondiente que es factible. Suponiendo no degeneración,  $\bar{b}_r > 0$ . Ya que  $\bar{a}_{rs} > 0$  tenemos  $\bar{x}_s > 0$  por (1.27) y  $z < \bar{z}_0$  por (1.26). Por construcción de  $\bar{x}_s$  por medio de la ecuación (1.27),  $x_{j_r} = 0$  y las variables restantes  $x_B \geq 0$ , que implica que la nueva solución es factible. Para mostrar que la nueva solución es básica, obsérvese que puesto que  $\bar{a}_{rs} > 0$ , podemos usar la  $r$ -ésima ecuación de la forma canónica (1.8) y  $\bar{a}_{rs}$  como elemento pivote para eliminar la variable  $x_s$  de las otras ecuaciones y de la forma minimizada. Sólo es necesaria esta única operación de pivoteo para reducir el sistema a la forma canónica relativa al nuevo conjunto de variables. Este hecho, y la forma en que  $s$  es seleccionada, constituye la clave de la eficacia computacional del Método Símplex. La solución básica asociada con el nuevo conjunto de variables básicas es único.

**Teorema 1.2.9 (Conclusión finita)** *Suponiendo que no existe degeneración en cada iteración, el algoritmo símplex terminará en un número finito de iteraciones.*

**Demostración.** Hay sólo un número finito de formas de escoger un conjunto de  $m$  variables básicas de todas las  $n$  variables. Si el algoritmo continuara indefinidamente, sería sólo repitiendo periódicamente el mismo conjunto de variables básicas obtenido anteriormente en alguna iteración pasada, además de el mismo sistema canónico y el mismo valor de  $z$ . Esta repetición no ocurre puesto que el valor de  $z$  decrece estrictamente con cada iteración bajo no degeneración. ■

### 1.3. Método símplex

El Método Símplex es aplicado a programas lineales en la forma estándar (1.2). Emplea el Algoritmo Símplex presentado en la sección (1.2) en dos fases. En la Fase I, se busca una solución básica factible inicial para continuar con la Fase II o para determinar que no existe la solución factible. Si se encuentra, entonces en la Fase II se busca una solución básica factible óptima o una clase de soluciones factibles con  $z \rightarrow -\infty$ .

Muchos problemas encontrados en la práctica a menudo tienen una forma canónica factible con la cual empezar. Por ejemplo, podemos construir una gran variedad de soluciones básicas factibles de inicio para problemas de transporte. Otros modelos, tales como los modelos económicos, a menudo contienen almacenamientos y actividades de holgura, permitiendo una solución de inicio obvia que emplea estos almacenamientos y actividades. Como solución, si es básica, puede estar lejos de la solución óptima, pero proporciona un inicio fácil. Incluso si no es básica, puede ser modificada a una solución factible en no más de  $k$  pasos de pivoteo, donde  $k \leq n - m$ .

No obstante, hay muchos problemas encontrados en la práctica donde ninguna forma canónica factible obvia esta disponible, y se requiere una aproximación a la fase I. Inicialmente nada se conoce (matemáticamente hablando) acerca del problema. Depende del algoritmo determinar si existen o no:

1. *Redundancias*: Esto ocurre, por ejemplo, si una ecuación que considera flujos de dinero ha sido obtenida de las ecuaciones que consideran el precio de los materiales multiplicadas por las que tienen que ver con la cantidad de los mismos.
2. *Inconsistencias*: Esto podría ocurrir a causa de los errores de entrada, uso de datos incorrectos, o por la especificación de requisitos que no pueden llenarse con los recursos disponibles. Por ejemplo, se puede idear un problema donde los recursos son para acortar el suministro y la pregunta principal es si existe o no una solución factible.

El procedimiento de la fase I, que usa el algoritmo símplex propio para proveer una forma canónica factible de inicio (si existe) para la fase II, tiene varios rasgos importantes:

1. Ninguna suposición se ha hecho con respecto al sistema original; puede

ser redundante, inconsistente, o sin solución en los números no negativos.

2. No se requiere de eliminación para obtener una solución inicial para la forma canónica para la fase I.
3. El resultado de la fase I es una solución básica factible (si existe) en la forma canónica listo para iniciar la fase II.

### 1.3.1. El método

El primer paso del Método Símplex es la introducción en la forma estándar de variables no negativas adicionales resultando que el problema aumentado esté en la forma canónica. Estas son precisamente las variables de holgura o artificiales.

Ahora en este punto se emplea el Algoritmo Símplex. Consiste de una serie de operaciones de pivoteo, referido como fase I, que produce una sucesión de formas canónicas diferentes con la propiedad que la suma de las variables artificiales decrecen con cada iteración. El objetivo es conducir esta suma a cero. Si hacemos esto, hemos encontrado una solución básica factible al sistema original con el cual iniciar la fase II.

**Algoritmo 1.3.1 (Método símplex)** Con este algoritmo, aclararemos los puntos anteriores.

1. *Tomar cada  $b_i$  no negativo.* Modificar el sistema original de ecuaciones (1.2) para que todos los términos constantes  $b_i$  sean no negativos multiplicando la o las ecuaciones donde  $b_i$  es menor que cero, por  $-1$ .
2. *Adicionar variables artificiales.* Para tener una solución factible inicial para la fase I, aumentar el sistema de ecuaciones para incluir un conjunto básico

$$x_a = (x_{n+1}, x_{n+2}, \dots, x_{n+m}) \geq 0$$

de variables artificiales, así, el sistema es:

$$\begin{array}{rcl} -w & + & e^T x_a = 0 \\ Ax & + & Ix_a = b \\ x & & \geq 0 \\ & & x_a \geq 0 \end{array} \quad (1.28)$$

donde  $e = (1, 1, \dots, 1)^T$ .

3. *Fase I.* Usar el Algoritmo Símplex para encontrar una solución de (1.28) que minimice la suma de variables artificiales  $w$ :

$$w = \sum_{j=n+1}^{n+m} x_j \quad (1.29)$$

La ecuación (1.29) es llamada la *forma de infactibilidad*. El sistema canónico factible inicial para la fase I se obtiene seleccionando como variables básicas  $x_a, (-w)$ , y eliminando  $x_a$  de la forma de infactibilidad restando la suma de las últimas  $m$  ecuaciones de (1.28) de la primera ecuación, produciendo:

$$\begin{array}{rcl} -w + d^T x & = & -\bar{w}_0 \\ Ax + Ix_a & = & b \\ x & \geq & 0 \\ & & x_a \geq 0 \end{array} \quad (1.30)$$

donde  $b_i \geq 0$  y

$$\begin{array}{rcl} d & = & -A^T e \\ -\bar{w}_0 & = & -e^T b \end{array} \quad (1.31)$$

4. *Terminar si*  $\text{mín } w > 0$  *al final de la fase I.* No existe solución factible al problema original.
5. *Empezar la fase II si*  $\text{mín } w = 0$ . Iniciar la fase II del método símplex mediante:
- eliminando de otra consideración todas las variables no artificiales no básicas  $x_j$  donde sus coeficientes correspondientes  $\bar{d}_j$  son positivos (no cero) en la ecuación  $w$  final;
  - eliminando de otra consideración todas las variables artificiales no básicas;
  - eliminando la forma lineal  $w$  (modificada por varias eliminaciones);
  - introduciendo la forma lineal  $z$  primero antes de eliminar todas las variables básicas no artificiales y entonces aumentar la  $z$  resultante por términos artificiales básicos con coeficientes cero.

6. *Fase II.* Aplicar el algoritmo símplex a la forma canónica factible obtenida e iterar para encontrar una solución que minimice el valor de  $z$  o genere una clase de soluciones tal que  $z \rightarrow -\infty$ .

### 1.3.2. Teoría detrás de la Fase I

Es claro que si existe una solución factible al sistema original (1.2), entonces esta misma solución también satisface (1.28) con el conjunto de variables artificiales igual a cero, o sea  $w = 0$  en este caso. De (1.29), el valor más pequeño posible para  $w$  es cero ya que  $w$  es la suma de las variables no negativas. Ahora, si existen soluciones factibles, el mínimo valor de  $w$  será  $w = 0$ . Recíprocamente, si una solución es obtenida para (1.28) con  $w = 0$ , es claro que todos los  $x_{n+i} = 0$  y los valores de  $x_j$  para  $j \leq n$  constituyen una solución factible para (1.2). También que si  $\min w > 0$ , entonces no existen soluciones factibles para (1.2). Note que el procedimiento de la fase I no produce un problema no acotado ya que esto implicaría falsamente que  $w$  definida como una suma de variables negativas no tiene un límite inferior.

Cuando el sistema original contiene redundancias y cuando a menudo ocurren soluciones degeneradas, las variables artificiales permanecerán como parte del conjunto básico de variables al final de la fase I. Así, es necesario que durante la fase II sus valores nunca excedan de cero. Queremos dar ahora tres maneras diferentes para poder lograr esto.

1. Eliminar todas las variables no artificiales cuyo factor de costo relativo para  $w$  sea positivo y eliminando todas las variables artificiales no básicas. Para ver esto notamos que la ecuación  $w$  al final de la fase I satisface

$$\sum_{j=1}^{n+m} \bar{d}_j x_j = w - \bar{w}_0, \quad (1.32)$$

donde  $\bar{d}_j \geq 0$  y  $\bar{w} = 0$ , así que la región factible existe. Para factibilidad,  $w$  debe permanecer en cero en la fase II, lo que indica que todo  $x_j$  correspondiente a  $\bar{d}_j > 0$  debe ser cero; así que todos esos  $x_j$  pueden ser conjuntamente igual a cero y ser eliminados de otra consideración en la fase II. Ahora nuestra atención está confinada sólo a las variables que corresponden a  $\bar{d}_j = 0$ . Todas las soluciones factibles que involucran sólo estas variables restantes ahora tienen  $w = 0$  por (1.32), y por consiguiente las variables artificiales restantes que suman a  $w$  son también

cero y serán cero con los pivoteos subsecuentes. En consecuencia, la solución factible del problema modificado aumentado es también una solución factible del problema original.

Una variante de este método es tratar la ecuación  $z$  como otra restricción, con  $z$  no restringida en signo durante la fase I. Esto automáticamente elimina las variables básicas  $x_B$  de la ecuación  $z$  en cada iteración. Esto no requiere manipulación de la estructura de los datos, pero puede incurrir en más operaciones.

2. Una segunda manera para mantener los valores de las variables artificiales básicas en cero durante la fase II, es intentar eliminar (si es posible) todas las variables artificiales aun en el conjunto básico. Esto puede hacerse escogiendo un pivoteo en una fila  $r$  que corresponda a una variable artificial y en cualquier columna  $s \leq n$  tal que  $\bar{a}_{rs} \neq 0$ . Si todo coeficiente en tal fila para  $j = 1, \dots, n$  son cero, la fila puede ser eliminada porque la ecuación correspondiente en el sistema original es redundante, o la fila puede sacarse si es que eso es más conveniente.
3. Una tercera manera es mantener la ecuación  $w$  durante la fase II y tratar la variable  $(-w)$  como otra variable que es restringida a valores no negativos. El sistema es entonces aumentado por la introducción de la ecuación  $z$  después eliminando las variables básicas de él. Puesto que  $w \geq 0$  es siempre verdadero, la condición agregada  $(-w) \geq 0$  implica  $w = 0$  durante la fase II.

**Teorema 1.3.2 (Variables artificiales en la fase II)** *Si hay variables artificiales que forman parte de los conjuntos básicos de variables en las distintas iteraciones de la fase II, entonces sus valores nunca excederán a cero.*

**Demostración.** La demostración sigue de la discusión de la primera forma de mantener las variables artificiales en cero en la fase II. ■

## 1.4. Variables acotadas

Nuestra atención es ahora para resolver un programa lineal con variables acotadas, esto es:

$$\begin{aligned} &\text{Minimizar} && c^T x \\ &\text{sujeto a} && Ax = b, \quad A : m \times n, \\ &&& l \leq x \leq u. \end{aligned} \tag{1.33}$$



Las variables independientes corresponderán a variables no básicas,  $x_N$ , y dependientes serán las variables básicas,  $x_B$ . Sea la forma canónica con respecto a  $x_B$  como antes:

$$\begin{aligned} (-z) + 0x_B + \bar{c}^T x_N &= -\bar{z}_0 \\ Ix_B + \bar{A}x_N &= \bar{b} \end{aligned} \quad (1.34)$$

donde el subíndice  $B$  es el conjunto de índices de variables básicas, y el subíndice  $N$  es el conjunto de índices de las variables no básicas.

**Definición 1.4.1 (Solución básica)** *Suponiendo que por lo menos uno de los límites para cada  $x_j$  es finito, la solución especial obtenida haciendo cada no básica igual a otra, es cota inferior o superior y resolviendo para los valores de las variables básicas será llamada una solución básica.*

**Definición 1.4.2 (Solución degenerada)** *Una solución básica es degenerada si los valores de una o más de las variables básicas en la solución básica está en su límite inferior o límite superior.*

**Teorema 1.4.3** *Sea  $x = (x_B^*, x_N^*)$  una solución básica para (1.34) de acuerdo a la definición anterior, esto es,  $x_B^* = \bar{b} - \bar{A}x_N^*$  y cada componente no básico de  $x^*$ , para  $j \in N$ , debe satisfacer  $x_j^* = l_j$  o  $x_j^* = u_j$ . Entonces  $(x_B^*, x_N^*)$  es una solución mínima con costo total  $z^* = \bar{z}_0 + \bar{c}^T x_N^*$ , si*

$$\begin{aligned} l_B &\leq x_B^* = \bar{b} - \bar{A}x_N^* \leq u_B & (a) \\ \bar{c}_j &\geq 0 \quad \text{para } j \in L & (b) \\ \bar{c}_j &\leq 0 \quad \text{para } j \in U & (c) \end{aligned} \quad (1.35)$$

donde

$$\begin{aligned} L &= \{j \in N \mid x_j^* = l_j\}, \\ U &= \{j \in N \mid x_j^* = u_j\} \end{aligned} \quad (1.36)$$

**Demostración.** Para que la solución básica sea factible debe tenerse (1.35a). El valor de la función objetivo está dado por:

$$z^* = \bar{z}_0 + \bar{c}^T x_N^* = \bar{z}_0 + \sum_{j \in L} \bar{c}_j x_j^* + \sum_{j \in U} \bar{c}_j x_j^*. \quad (1.37)$$

Para cada  $(x, z)$  factible, tenemos

$$z = \bar{z}_0 + \sum_{j \in L} \bar{c}_j x_j + \sum_{j \in U} \bar{c}_j x_j. \quad (1.38)$$

Si  $\bar{c}_j \geq 0$  para  $j \in L$ , entonces lo mejor que podemos hacer es tener los correspondientes  $x_j$  siendo sus límites inferiores,  $x_j^* = l_j$ . Similarmente, si  $\bar{c}_j \leq 0$  para  $j \in u$ , entonces lo que podemos hacer es tener los correspondientes  $x_j$  siendo los límites superiores,  $x_j^* = u_j$ . De esta manera el lado derecho de (1.37) es un límite inferior para  $z$ . Ya que este límite inferior es obtenido para la solución básica factible  $(x_B^*, x_N^*)$ , el mínimo valor de  $z$  en (1.38) es el mínimo valor de  $z$  obtenido para todas las soluciones factibles, básicas o no.

■

## 1.5. Método símplex revisado

El método símplex revisado no es un método diferente pero es una manera diferente de llevar a cabo cada paso operacional del método símplex. El método símplex revisado ofrece varias ventajas sobre el algoritmo símplex.

1. Ahorros considerables en operaciones son posibles si la fracción de coeficientes no cero es menor que  $1 - (2m/n)$ . Esto es típico en la práctica.
2. Se guardan menos datos de una iteración a la siguiente, lo que permite resolver un problema más grande cuando la capacidad de memoria de una computadora está limitada. Además, son posibles ahorros considerables en las operaciones porque no todas las entradas en la tabla necesitan ser actualizadas.
3. Una debilidad del algoritmo símplex en la forma de tabla es que el inverso de la base en forma explícita es parte de la tabla llena. La representación explícita de la inversa puede ser numéricamente inestable. Ya que la inversa es sólo necesitada para resolver ciertos sistemas de ecuaciones, la necesidad de tener una inversa puede evitarse en el algoritmo símplex revisado resolviendo el sistema de ecuaciones numéricamente por métodos estables.

### 1.5.1. Motivación.

Mientras cada iteración del método símplex requiere calcular y memorizar una nueva tabla, puede observarse que sólo la fila del costo modificado y la columna correspondiente a la variable introducida al conjunto básico juegan un papel en el proceso de decisión. Esto es, para llevar a cabo los pasos del algoritmo símplex (ver el algoritmo 1.2.4)

1. Primero vemos los costos reducidos ( $\bar{c}$ ) para determinar si es posible una mejora en la solución, y en ese caso, determinamos qué columna  $s$  poner en la base (ver los pasos 1-3 del algoritmo símplex).
2. Después necesitamos ver la columna pivote  $\bar{A}_{os}$  y el lado derecho  $s$  para determinar la fila pivote  $r$  y la variable que deja la base (pasos 4-5 del algoritmo).

Ya que  $\bar{b}$  puede ser fácilmente actualizado, para cada iteración, todo lo que necesitamos son los costos reducidos y la columna actualizada correspondiente a la variable entrante. Resulta que podemos obtener esta información directamente si tenemos disponible una manera de resolver un sistema de ecuaciones cuya matriz de coeficientes es la matriz de la base actual  $B$  o su transpuesta.

Para el propósito de ilustrar el algoritmo del método símplex revisado en pequeños ejemplos, necesitamos usar representaciones explícitas de  $B^{-1}$  y una manera de actualizarlo. En general como hemos notado, calculando y usando una inversa puede ser un proceso numéricamente inestable y además, para problemas grandes, una representación explícita de  $B^{-1}$  puede ser computacionalmente no eficiente en términos de rapidez y almacenamiento.

Si tenemos a la mano una representación explícita de  $B^{-1}$ , es fácil derivar las cantidades requeridas  $\bar{c}$  y  $\bar{A}_{os}$  de los datos originales. Supóngase que está disponible una solución factible inicial para un programa en la forma estándar (1.2) dado. Supóngase, para conveniencia de la discusión, que las columnas de la matriz de coeficientes  $A$  han sido ordenadas de manera que las columnas de la base (representadas por la matriz  $B = B_t$  para la iteración  $t$ ) son las primeras  $m$  columnas de  $A$  y las columnas no básicas (representadas por la matriz  $N = N_t$ ) son las últimas  $(n - m)$  columnas de  $A$ . Esto es:

$$\begin{aligned} (-z) + c_B^T x_B + c_N^T x_N &= 0 & (a) \\ Bx_B + Nx_N &= b & (b) \end{aligned} \quad (1.39)$$

Algebraicamente, la forma canónica de (1.39) para la iteración  $t$  esta dada por:

$$\begin{aligned} (-z) + (c_N^T - c_B^T B^{-1} N)x_N &= -c_B^T B^{-1} b & (a) \\ Ix_B + B^{-1} Nx_N &= B^{-1} b & (b) \end{aligned} \quad (1.40)$$

Esto puede verse multiplicando (1.39b) por  $B^{-1}$  en la izquierda para obtener (1.40b). Si ahora multiplicamos (1.40b) en la izquierda por  $c_B^T$  y

sustrayendo de (1.39a), obtenemos (1.40a). Pero esta no es la mejor manera de hacer los cálculos.

Para calcular  $\bar{c}$  eficientemente primero calculamos:

$$\pi^T = c_B^T B^{-1}, \quad (1.41)$$

y entonces

$$\bar{c}^T = c_N^T - \pi^T N, \quad (1.42)$$

y luego  $\bar{c}_s = \text{mín } \bar{c}_j$ . Entonces si  $\bar{c}_s < 0$ , calculamos

$$\bar{A}_{os} = B^{-1} A_{os} \quad (1.43)$$

El valor  $\bar{b} = B^{-1}b$  no es calculado explícitamente; en cambio es actualizado en el tabla del símplex revisado cuando se ejecuta un pivoteo.

**Teorema 1.5.1 (Unicidad de  $\pi$ )** *Para cualquier iteración  $t$ , el vector precio (o multiplicador símplex)  $\pi$  es único.*

**Definición 1.5.2** *La operación de multiplicar  $\pi$  por las columnas no básicas  $A_{oj}$  para  $j$  no básicas en la determinación de  $N^T \pi$  es llamada precio de salida de las columnas  $j$ .*

La combinación lineal de las columnas en la base que produce la columna entrante  $A_{os}$  es llamada la *representación de la  $s$ -ésima actividad en términos del conjunto básico de actividades*. Es fácil ver que esta representación es la columna actualizada  $A_{os}$ , vease (1.43), porque en notación matricial (1.43) puede ser reescrito como:

$$B\bar{A}_{os} = A_{os} \quad (1.44)$$

El precio de salida de una columna y la representación de una actividad en términos de una base puede ser terminada muy eficientemente para matrices que tiene muchos elementos cero (llamadas *matrices de cambio*).

Enseguida se muestra cómo encontrar fácilmente  $B^{-1}$  y  $\pi$  en la tabla. Por conveniencia de discusión reordenamos las variables como antes en (1.39) así que las primeras  $m$  columnas de  $A$  son  $B = B^t$ . Supóngase también que agregamos  $m$  variables artificiales  $x_a$  a (1.39b) para obtener:

$$\begin{aligned} (-z) + c_B^T X_B + c_N^T x_N &= 0 & (a) \\ Bx_B + Nx_N + Ix_a &= b & (b) \end{aligned} \quad (1.45)$$

Una vez más multiplicando (1.45b) por  $B^{-1}$ , obtenemos (1.46b) y sustrayendo el resultado de  $c_B^T$  por (1.46b) de (1.45a), obtenemos (1.46a), que juntos dan la forma canónica para la iteración  $t$ :

$$\begin{aligned} (-z) &+ (c_N^T - c_B^T b^{-1} N)x_N - c_B^T B^{-1} x_a = -c_B^T B^{-1} b & (a) \\ Ix_B + & B^{-1} N x_N + B^{-1} x_a = B^{-1} b & (b) \end{aligned} \quad (1.46)$$

De esta manera,  $\pi^T = c_B^T B^{-1}$  y  $B^{-1}$  para alguna iteración  $t$  puede ser leído fuera de la tabla examinando las columnas correspondientes a  $x_a$ .

Además, si queremos cambiar la base pivotando en una nueva columna de  $\bar{A} = B^{-1}N$ , entonces la nueva base inversa, dígase  $\bar{B}^{-1}$ , estaría disponible en las columnas correspondientes a  $x_a$ .

La discusión anterior implica que podemos reducir el tamaño de nuestra tabla conservando sólo la columna  $(-z)$ , las columnas correspondientes a  $x_a$ , y el lado derecho actualizado. Con esto podemos generar el costo reducido  $\bar{c}$  y la columna entrante  $\bar{A}_{os}$ . La tabla compactada se muestra abajo:

$(-z)$	$x_a$	Lado derecho	$x_s$
1	$-\pi^T$	$-\pi^T \bar{b}$	$\bar{c}_s$
	$B^{-1}$	$\bar{b}$	$\bar{A}_{os}$

← Pivote en  $\bar{A}_{rs}$

Cuadro 1.1: Tabla para el Método Símplex Revisado

Así, nunca calculamos explícitamente  $\bar{A}$ . Para actualizar la tabla del símplex revisado, la tabla del cuadro 1.1 es pivotada en la última columna en el elemento  $\bar{a}_{rs}$ . De esta manera, el  $\bar{b}$  para la siguiente iteración es calculado por el paso pivote.

El algoritmo del Método Símplex Revisado se presenta a continuación.

**Algoritmo 1.5.3 (Método Símplex Revisado)** Considérese un programa lineal en forma estándar y que  $x_B = \bar{b}$  es una solución factible y que las columnas con índices  $j_1, \dots, j_m$  para una base factible  $B$  cuya inversa  $B^{-1}$  es conocida.

1. *Multiplicadores símplex.* Determine los multiplicadores símplex:

$$\pi = (B^{-1})^T c_B$$

2. *Costos reducidos.* Determine los costos reducidos:

$$\bar{c}_j = c_j - \pi^T A_{oj}$$

para  $j$  no básico.

3. *Menor costo reducido.* Paso 1 del algoritmo 1.2.4.
4. *Prueba de optimalidad.* Paso 2 de 1.2.4.
5. *Variable entrante.* Paso 3 de 1.2.4.
6. *Determinar  $\bar{A}_{os}$ ,* la representación de  $A_{os}$  en términos de  $B$ .
7. *Prueba para  $z$  no acotada.* Paso 4 de 1.2.4.
8. *Variable saliente.* Paso 5 de 1.2.4.
9. Pivotear en  $\bar{a}_{rs}$  en la matriz  $(\bar{b}, B^{-1}, \bar{A}_{os})$  para obtener  $(\bar{b}$  actualizado,  $B^{-1}$  actualizado,  $e_r$ ), donde  $e_r$  es el vector unitario con 1 en la fila  $r$  y ceros en las demás.
10. Tomar  $j_r = e_r$  y regresar al paso 1 con  $\bar{b}$  actualizado y  $B^{-1}$ .

Para finalizar esta sección, ilustraremos el Método Simplex Revisado usando la forma de tabla. Nuestro problema es encontrar mín  $z$ ,  $x \geq 0$  tal que:

$$\begin{aligned} 2x_1 + x_2 + 2x_3 + x_4 + 4x_5 &= z \\ 4x_1 + 2x_2 + 13x_3 + 3x_4 + x_5 &= 17 \\ x_1 + x_2 + 5x_3 + x_4 + x_5 &= 7 \end{aligned} \quad (1.47)$$

Para facilitar la exposición, desarrollaremos la forma de tabla pero sólo mostraremos las cantidades que son determinadas explícitamente u obtenidas por pivoteos.

La forma con coeficientes separados para (1.47) se muestra en el cuadro 1.2. Los coeficientes que aparecen en el renglón superior son  $d_j = -e^T A_{oj}$ , donde  $e = (1, 1, \dots, 1)^T$ . Los costos reducidos  $\bar{d}_j$  que serán calculados en las iteraciones 0,1 y 2 de la fase I del Método Simplex Revisado están almacenados en la línea  $-w$  de cada iteración en el cuadro 1.3, como lo serán los costos reducidos  $\bar{c}_j$  para las iteraciones 2 y 3 de la fase II en el cuadro 1.4. El cuadro 1.3 muestra los datos guardados al inicio de la iteración  $t$  del Método Simplex Revisado. La columna llamada “variables básicas” muestra

el nombre de las variables en el orden  $(j_1, \dots, j_m)$  en la base. Las columnas *relevantes* de la forma canónica corresponden a las variables artificiales y contienen el negativo de los multiplicadores símplex y la base inversa. Los multiplicadores símplex  $\pi$  son los coeficientes negativos de las artificiales en la fila  $-w$  en la fase I; en la fase II éstos están en la fila  $-z$ . La columna después de la última variable artificial es el lado derecho modificado. Estos datos se obtienen por pivoteo.

Para calcular  $\bar{d}_j$  y  $\bar{c}_j$  directamente del sistema original en el cuadro 1.2, necesitaremos los multiplicadores símplex asociados con la iteración  $t$ . Los números mostrados en cursiva o negrita son generados directamente del sistema original. Recuérdesse que  $\bar{d}_j = d_j - \pi^T A_{oj}$  y  $\bar{c}_j = c_j - \pi^T A_{oj}$  pueden ser calculados de los datos originales. Si la solución no es óptima, determinamos la columna  $s$  entrante. La inversa de la base es usada para calcular  $\bar{A}_{os}$  directamente de los datos originales por la fórmula  $\bar{A}_{os} = B^{-1}A_{os}$ . El pivoteo transforma la columna  $s$  en un vector columna unitario que no se guarda.

Variables básicas			Variables originales					Variables artificiales		Constantes
	<i>-w</i>	<i>-z</i>	<i>x</i> <sub>1</sub>	<i>x</i> <sub>2</sub>	<i>x</i> <sub>3</sub>	<i>x</i> <sub>4</sub>	<i>x</i> <sub>5</sub>	<i>x</i> <sub>6</sub>	<i>x</i> <sub>7</sub>	
<i>-w</i>	1		-5	-3	-18	-4	-2			-24
<i>-z</i>		1	2	1	2	1	4			0
<i>x</i> <sub>6</sub>			4	2	13	3	1	1		17
<i>x</i> <sub>7</sub>			1	1	5	1	1		1	7

Cuadro 1.2: Coeficientes

En la iteración 0, la base inversa de la fase I es la matriz identidad (ver cuadro 1.3), y ahora las entradas mostradas en la tabla son las mismas que las correspondientes entradas de los datos originales. Ya que el costo reducido  $\bar{d}_3 = -18$  es el menor, ponemos el correspondiente  $x_3$  en la base en la próxima iteración. Ya que  $B^{-1} = I$  para esta iteración,  $\bar{A}_{o3} = A_{o3}$  puede obtenerse directamente de los datos originales e introducida en la columna correspondiente del cuadro 1.3 para la iteración 0. La columna actualizada  $\bar{A}_{o3}$  y  $\bar{b}$  nos permite encontrar el pivote, que es  $\bar{A}_{13}$ . Pivoteando en  $\bar{A}_{13}$  mandamos a  $x_6$  fuera de la base, reemplazándola por  $x_3$ , y generando las columnas artificiales y  $\bar{b}$  actualizado de la siguiente iteración,  $t = 1$ .

En la iteración 1, la base inversa es:

$$B^{-1} = \begin{pmatrix} \frac{1}{13} & 0 \\ -\frac{5}{13} & 1 \end{pmatrix}$$

en las columnas correspondientes a  $x_6$  y  $x_7$  en el cuadro 1.3. Los multiplicadores símplex son  $\pi_1 = -18/13$  y  $\pi_2 = 0$ , que son los negativos de las entradas en la primera fila en las columnas correspondientes a  $x_6$  y  $x_7$ . Así, los  $\bar{d}_j$  pueden ser calculados usando (1.42) como  $d_j - \pi^T A_{oj}$  para la iteración 1 y así en adelante.

Iteración 0 (Fase I)									
Variables básicas	OBJ	Variables originales					Variables artificiales		Lado derecho
	$-w$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	
		$A_{o3} = B^{-1}A_{o3}$					$-\pi_1$	$-\pi_2$	
$-w$	1	-5	-3	-18	-4	-2	0	0	-24
$x_6$				<b>13</b>			1		17
$x_7$				5				1	7
							$B^{-1}$		
Iteración 1 (Fase I) $\bar{d}_j = d_j - \pi^T A_{oj}$									
		$B^{-1}A_{o5} = \bar{A}_{o5}$					$-\pi_1$	$-\pi_2$	
$-w$	1	$\frac{7}{13}$	$-\frac{3}{13}$	0	$\frac{2}{13}$	$-\frac{8}{13}$	$\frac{18}{13}$		$-\frac{6}{13}$
$x_3$					$\frac{1}{13}$	$\frac{1}{13}$			$\frac{17}{13}$
$x_7$					$\frac{8}{13}$	$-\frac{5}{13}$		1	$\frac{6}{13}$
							$B^{-1}$		
Iteración 2 (Fase I óptima) $\bar{d}_j = d_j - \pi^T A_{oj} \geq 0$									
$-w$	1	0	0	0	0	0	1	1	0
$x_3$							$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{5}{4}$
$x_5$							$-\frac{5}{8}$	$\frac{13}{8}$	$\frac{3}{4}$
							$B^{-1}$		

Cuadro 1.3: Fase I del Método Símplex Revisado



Iteración 2 (Inicio de fase II: introducción de  $z$ )

Variables básicas	OBJ	Variables originales					Variables artificiales		Lado derecho
	$-z$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	
$-z$	1	2	1	2	1	4	0	0	0
$x_3$							$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{5}{4}$
$x_5$							$-\frac{5}{8}$	$\frac{13}{8}$	$\frac{3}{4}$

$B^{-1}$

Iteración 2 (Fase II:  $z$  actualizado)  $\bar{c}_j = c_j - \pi^T A_{oj}$   
 $\bar{A}_{o2} = B^{-1} A_{o2}$   $-\pi = -c_B^T B^{-1}$

$-z$	1	$\frac{19}{4}$	$-\frac{3}{4}$		$\frac{3}{2}$		$\frac{9}{4}$	$-\frac{25}{4}$	$-\frac{11}{2}$
$x_3$			$\frac{1}{8}$				$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{5}{4}$
$x_5$			$\frac{3}{8}$				$-\frac{5}{8}$	$\frac{13}{8}$	$\frac{3}{4}$

$B^{-1}$

Iteración 3 (Fase II óptima)  $\bar{c}_j = c_j - \pi^T A_{oj} \geq 0$

$-z$	1	3	0	0	1	2	1	$\frac{19}{2}$	-4
$x_3$							$\frac{1}{3}$	$-\frac{2}{3}$	1
$x_2$							$-\frac{5}{3}$	$\frac{13}{3}$	2

$B^{-1}$

Cuadro 1.4: Fase II del Método Símplex Revisado

## 1.6. Dualidad en programación lineal

Es un hecho bien conocido, que un problema de programación lineal lo podemos ver de dos “enfoques” distintos.

Como sabemos, hemos considerado el sistema (1.2) la forma estándar de un problema inicial, a éste le llamamos el *PRIMAL*. Ahora bien, podemos ver el otro lado del problema al hacer la transpuesta del sistema inicial (primal) y cambiar de sentido el problema, esto es, si se trata de un problema inicial de maximización, corresponderá a un problema de minimización y viceversa. A este problema final se le conoce como el *DUAL*.

Mostramos el sentido general de estos puntos en una forma matricial:

Sea un problema primal:

$$(Max)\bar{c}\bar{x}^T \quad \text{sujeto a :}$$

$$A\bar{x}^T \leq \bar{b}^T$$

Entonces su correspondiente problema dual está dado por:

$$(Min)\bar{y}\bar{b}^T \quad \text{sujeto a :}$$

$$\bar{y}A \geq \bar{c}$$

Se presentan algunas propiedades de estas formas, así como sus fundamentos teóricos.

**Teorema 1.6.1** *Si  $\bar{X}$  y  $\bar{Y}$  son soluciones factibles a un par de problemas, uno primal ( $P_1$ ) y su correspondiente dual ( $D_1$ ), entonces*

$$Z = c\bar{X}^T \leq \bar{Y}b^T = G$$

**Demostración.** Como  $\bar{X}$  es factible en ( $P_1$ ), cumple con  $A\bar{X}^T \leq b^T$  y  $\bar{X} \geq 0$ . Como  $\bar{Y}$  es factible en ( $D_1$ ), cumple con  $\bar{Y}A \geq c$  y  $\bar{Y} \geq 0$ . Premultiplicando  $A\bar{X}^T \leq b^T$  por  $\bar{Y} \geq 0$  y a  $\bar{Y}A \geq c$  por  $\bar{X}^T \geq 0$  se tiene

$$\bar{Y}A\bar{X}^T \leq \bar{Y}b^T$$

$$\bar{Y}A\bar{X}^T \geq c\bar{X}^T$$

De estas dos últimas expresiones se concluye que

$$Z = c\bar{X}^T \leq \bar{Y}A\bar{X}^T \leq \bar{Y}b^T = G$$

y el teorema queda demostrado. ■

El teorema 1.6.1 únicamente dice que para cualquier par de soluciones factibles  $(\bar{X}, \bar{Y})$  de  $(P_1)$  y  $(D_1)$ , la función objetivo primal es siempre menor o igual a la función objetivo dual.

**Lema 1.6.2** *Sea  $A$  una matriz,  $b$  un vector. El sistema  $Ax^T \leq b^T$  tiene solución para algún  $x$  si y sólo si  $yb^T \geq 0$  para todo  $y \geq 0$  con  $yA = 0$ .*

**Teorema 1.6.3 (Teorema de dualidad)** *Sea  $A$  una matriz  $m \times n$ ,  $b \in \mathbf{R}^m$ ,  $c \in \mathbf{R}^n$  vectores entonces:*

$$\text{máx}\{cx^T \mid Ax^T \leq b^T\} = \text{mín}\{yb^T \mid y \geq 0; yA = c\}$$

*bajo el supuesto de que los dos conjuntos son finitos y no vacíos.*

**Demostración.** El vector  $x$  y el vector  $y$  existen por el supuesto de que ambos conjuntos son no vacíos. Entonces, si  $Ax^T \leq b^T$ ;  $y \geq 0$  y  $yA = c$ , se tiene que

$$cx^T = yAx^T \leq yb^T$$

por lo que si los óptimos que estamos comparando son finitos siempre se tiene que:

$$\text{máx}\{cx^T \mid Ax^T \leq b^T\} = \text{mín}\{yb^T \mid y \geq 0; yA = c\}$$

■

**Teorema 1.6.4** *Dado el siguiente par de problemas primal y dual*

<i>Primal</i>	<i>Dual</i>
$Min Z = cX^T$	$Max G = Yb^T$

*sujeto a*

$AX^T \geq b^T$	$YA \leq c$
$X \geq 0$ (P)	$Y \geq 0$ (D)

*una condición necesaria y suficiente para que  $\bar{X}$  y  $\bar{Y}$  sean óptimas respectivamente de (P) y de (D) es*

$$\bar{Y}(A\bar{X}^T - b^T) = 0$$

*y*

$$\bar{X}(\bar{Y}A - c) = 0$$

**Demostración.** Como  $\bar{X}$  es factible para (P) y  $\bar{Y}$  para (D) se tiene

$$\begin{aligned}\bar{Y}A &\leq c & A\bar{X}^T &\geq b^T \\ \bar{Y} &\geq 0 & \bar{X} &\geq 0\end{aligned}$$

o lo que es lo mismo

$$A\bar{X}^T - b^T \geq 0$$

y

$$\bar{Y}A - c \leq 0$$

Multiplicando a la primera desigualdad por  $\bar{Y} \geq 0$  y a la segunda por  $\bar{X}^T \geq 0$  no se afecta el sentido de la desigualdad, obteniéndose

$$\alpha = \bar{Y}(A\bar{X}^T - b^T) \geq 0 \quad \beta = (c - \bar{Y}A)\bar{X}^T \geq 0$$

Sumando  $\alpha$  y  $\beta$  se tiene

$$\begin{aligned}\alpha + \beta &= \bar{Y}(A\bar{X}^T - b^T) + (c - \bar{Y}A)\bar{X}^T \geq 0 \\ &= \bar{Y}A\bar{X}^T - \bar{Y}b^T + c\bar{X}^T - \bar{Y}A\bar{X}^T \geq 0 \\ &= c\bar{X}^T - \bar{Y}b^T \geq 0\end{aligned}$$

o sea

$$Z = c\bar{X}^T \geq \bar{Y}b^T = G$$

Pero del teorema (1.6.1) se tiene que

$$Z = c\bar{X}^T < \bar{Y}b^T = G$$

por lo que

$$Z = c\bar{X}^T = \bar{Y}b^T = G$$

concluyendo que

$$\alpha + \beta = c\bar{X}^T - \bar{Y}b^T = 0$$

Como se supuso,  $\alpha \geq 0$  y  $\beta \geq 0$  se tiene que  $\alpha = 0$  y  $\beta = 0$ .

El teorema queda demostrado. Este teorema recibe el nombre de *Teorema de Holgura Complementaria Débil*. ■

**Teorema 1.6.5** *Dado un par de problemas, uno primal y su correspondiente dual, se tienen las siguientes implicaciones:*

$$\begin{aligned} a) \quad & \bar{Y} > 0 \Rightarrow A\bar{X}^T = b^T \\ b) \quad & A\bar{X}^T > b^T \Rightarrow \bar{Y} = 0 \\ c) \quad & \bar{X} > 0 \Rightarrow \bar{Y}A = c \\ d) \quad & \bar{Y}A > c \Rightarrow \bar{X} = 0 \end{aligned}$$

La demostración se obtiene directamente de la aplicación del teorema (1.6.4).

Las relaciones (a), (b), (c) y (d) de este teorema deben ser ciertas para cualquier par de soluciones óptimas  $\bar{X}$  y  $\bar{Y}$  de un problema primal y su correspondiente dual, respectivamente. Puede darse el caso de que  $\bar{Y} = 0$  y  $A\bar{X}^T = b^T$ , o que  $\bar{X} = 0$  y  $\bar{Y}A = c$ . El siguiente teorema nos permite evitar lo anterior, es decir que no puede ocurrir simultáneamente que  $\bar{X} = 0$  y  $\bar{Y}A = c$ , o  $\bar{Y} = 0$  y  $A\bar{X}^T = b^T$ .

**Teorema 1.6.6** *Sea  $A$  una matriz,  $b, c$  vectores. Consideremos*

$$\text{máx}\{cx^T \mid Ax^T \leq b^T\} = \text{mín}\{yb^T \mid y \geq 0; yA = c\} \quad (1.48)$$

*Supongamos que los dos óptimos son finitos. Sean  $x_0$  y  $y_0$  soluciones factibles. Entonces los siguientes enunciados son equivalentes:*

1.  $x_0, y_0$  son soluciones óptimas de (1.48).
2.  $cx_0^T = y_0b^T$ .
3. Si una componente de  $y_0$  es positiva, la correspondiente desigualdad en  $Ax^T \leq b^T$  es satisfecha como igualdad por  $x_0$ , esto es,  $y_0(b^T - Ax_0^T) = 0$ .

**Demostración.**

(1)  $\Rightarrow$  (2) Si  $x_0, y_0$  son soluciones óptimas de (1.48), aplicando el teorema de dualidad sabemos que las soluciones coinciden en el óptimo, esto es:

$$cx_0^T = y_0b^T$$

(2)  $\Rightarrow$  (3) Sabemos de antemano que se cumple la desigualdad

$$cx_0^T = y_0Ax_0^T \leq y_0b^T$$

entonces si se tiene  $cx_0^T = y_0b^T$ , como los extremos coinciden, se tiene:

$$y_0Ax_0^T = y_0b^T$$

lo cual es equivalente a escribir:

$$y_0(b^T - Ax_0^T) = 0$$

(3)  $\Rightarrow$  (1) Es directo, si tomamos  $y_0(b^T - Ax_0^T) = 0$  esto implica que  $y_0b^T = y_0Ax_0^T = cx_0^T$  lo cual ocurre sólo en el caso en que  $x_0, y_0$  son soluciones óptimas de (1.48). ■

Ahora, después de dar un recordatorio general del método símplex y la optimización lineal, en la siguiente sección mencionaremos los problemas más frecuentes que se presentan al intentar obtener una solución óptima entera mediante aproximaciones de soluciones lineales que resultan de la aplicación del método símplex a problemas de programación entera.

## 1.7. Desventajas del símplex en la programación entera

Uno de los problemas más frecuentes de aproximar una solución entera mediante una solución lineal es el hecho de que el resultado de redondear los valores de la solución óptima lineal o de tomar el punto entero de la región factible que esté más cerca de este óptimo lineal, no constituye el óptimo entero, o visto de otro modo, la solución entera óptima, se encuentra “muy lejos” de la solución lineal óptima del mismo problema, tal es el caso del problema (4.3) que se presenta en la sección 4.1, cuyo óptimo lineal se obtiene con  $A = \frac{11}{2}, B = 3$ . Podemos ver que el punto entero en la región factible más próximo a esta solución lineal corresponde a  $A = 5, B = 3$ , con un valor de la función objetivo de 100. Como se podrá ver, el punto (4, 4), está en la región factible y constituye el óptimo entero del problema (4.3).

Otro de los problemas relacionado con la aproximación lineal de soluciones enteras, es cuando redondeamos la solución lineal óptima, y tal redondeo, nos resulta en un punto entero que se encuentra fuera de la región factible, que por consiguiente, no es solución. Incluso existen problemas en los que la región factible no incluye ningún punto entero, y en este caso, cualquier método o algoritmo para programación entera no podrá obtener una solución. Por

ejemplo, en  $\mathbf{R}^2$  consideremos la región definida por:

$$\begin{aligned} A + B &\leq 1 \\ 3A - B &\geq 1 \\ 2A - B &\leq 1 \end{aligned}$$

y  $A, B \geq 0$  (ver figura 1.1).

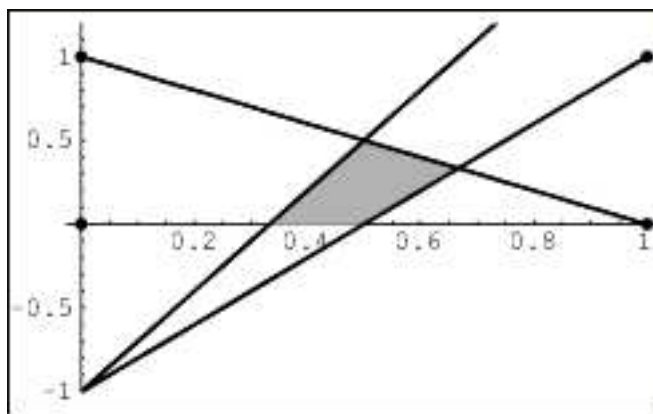


Figura 1.1: Región que no contiene puntos enteros

Un ejemplo donde la aproximación entera nos da un punto fuera de la región factible, es el siguiente:

$$\text{Max: } z = 3A + 4B$$

s.a.:

$$\begin{aligned} -A + 4B &\leq 12 \\ 3A + B &\leq 15 \\ A, B &\in \mathbf{R}^+ \end{aligned}$$

La solución lineal óptima de este problema, es cuando  $A = \frac{48}{13}$  y  $B = \frac{51}{13}$ , con un valor en la función objetivo de  $z = \frac{348}{13}$ . Si quisiéramos aproximar la solución entera de este problema por medio del redondeo, tendríamos que  $A = 4$  y  $B = 4$ , con lo que no se satisface la segunda restricción y por ende, no es solución factible del problema (ver figura 1.2).

La dificultad principal al tratar de resolver problemas de programación entera, es sin duda, la complejidad de manejo cuando el número de variables

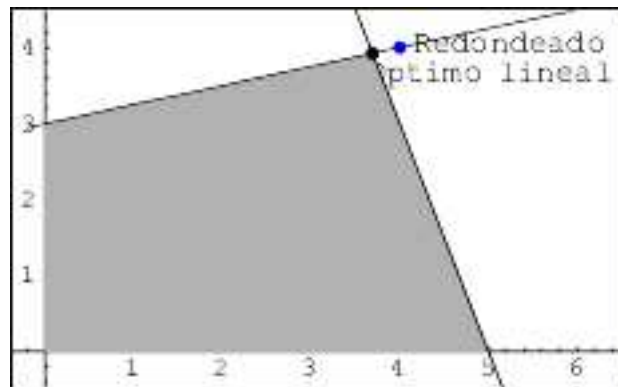


Figura 1.2: Entero óptimo (por redondeo) fuera de la región factible

se incrementa, o dicho geoméricamente, cuando el número de dimensiones aumenta. Tenemos experiencia al tratar problemas con dos variables o tres, incluso, con algunas dificultades, podemos resolver gráficamente estos problemas. Pero cuando los problemas implican cuatro o más variables, es imposible resolverlos por métodos gráficos, y a través del método símplex comienza a ser tedioso.

Así, el siguiente capítulo está enfocado a dar algunos planteamientos teóricos que servirán después para entender la mecánica de la herramienta que ha de usarse a fin de facilitar el tratamiento de los problemas de programación entera.



# Capítulo 2

## Algoritmo de la división

Este capítulo está dedicado a plantear las ideas básicas del algoritmo de la división para polinomios de una variable, así como los requerimientos y la forma de aplicar este algoritmo en polinomios de varias variables.

### 2.1. Algoritmo de la división en $\mathbb{R}[x]$

Antes de definir lo que es un anillo de polinomios y enunciar el algoritmo de la división, daremos la definición de lo que es un anillo.

**Definición 2.1.1** *Un conjunto no vacío  $R$  se dice que es un anillo asociativo si en  $R$  están definidas dos operaciones, denotadas por “+” y “.” respectivamente tales que para cualesquiera  $a, b, c$  de  $R$ :*

1.  $a + b$  está en  $R$ .
2.  $a + b = b + a$ .
3.  $(a + b) + c = a + (b + c)$ .
4. Existe un elemento  $0$  en  $R$  tal que  $a + 0 = a$  (para todo  $a$  en  $R$ ).
5. Existe un elemento  $-a$  en  $R$  tal que  $a + (-a) = 0$ .
6.  $a \cdot b$  está en  $R$ .
7.  $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ .
8.  $a \cdot (b + c) = a \cdot b + a \cdot c$ .

Si la multiplicación de  $R$  es tal que  $a \cdot b = b \cdot a$  para todo  $a, b \in R$  entonces llamamos a  $R$  *anillo conmutativo*. Además, si existe un elemento  $1$  en  $R$  tal que  $a \cdot 1 = 1 \cdot a = a$  para toda  $a \in R$ , diremos que  $R$  es un *anillo con elemento unitario*.

Dado que trabajaremos también con el concepto de campo, enunciaremos su definición.

**Definición 2.1.2** *Un campo  $k$  es un anillo conmutativo con elemento unitario en el que todo elemento distinto de cero tiene un inverso multiplicativo, es decir: existe  $b = \frac{1}{a}$  en  $k$  tal que  $b \cdot a = 1$  para todo  $0 \neq a \in k$ .*

Si  $R$  es un anillo, entonces un polinomio en una variable  $x$  con coeficientes en  $R$  se escribe como

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

donde los coeficientes  $a_i \in R$ .

**Teorema 2.1.3** *Sea  $k$  un campo y sea  $g \neq 0$  un polinomio en  $k[x]$ , donde  $k[x]$  es el anillo de polinomios de una variable sobre el campo  $k$ . Entonces, para todo  $f(x) \in k[x]$ , existen polinomios  $q(x)$  y  $r(x)$  en  $k[x]$  tal que  $f(x)$  se puede escribir como*

$$f(x) = q(x)g(x) + r(x)$$

con  $r(x) = 0$  o  $\deg(r(x)) < \deg(g(x))$ , además  $r(x)$  y  $q(x)$  son únicos.

**Definición 2.1.4** *Un polinomio  $p(x)$  en  $k[x]$  se dice que es irreducible sobre  $k$  si siempre que  $p(x) = a(x)b(x)$  con  $a(x), b(x) \in k[x]$ , entonces uno de los dos,  $a(x)$  o  $b(x)$ , tiene grado cero (es decir, es una constante).*

La irreducibilidad depende del campo; por ejemplo, el polinomio  $x^2 + 1$  es irreducible sobre el campo real, pero no sobre el campo complejo, pues en este último  $x^2 + 1 = (x + i)(x - i)$ , donde  $i^2 = -1$ .

Podemos tener una idea intuitiva de un polinomio en varias variables, pero ahora daremos una construcción precisa de esta noción.

Definimos el *anillo de polinomios en las  $n$  variables  $x_1, \dots, x_n$  sobre  $R$* ,  $R[x_1, \dots, x_n]$ , como sigue: sea  $R_1 = R[x_1]$ ,  $R_2 = R_1[x_2]$  es el anillo de polinomios en  $x_2$  sobre  $R_1$ , y en general  $R_n = R_{n-1}[x_n]$ . A  $R_n$  se le llama el anillo de polinomios en las variables  $x_1, \dots, x_n$  sobre  $R$ . Sus elementos son de la forma  $\sum a_i x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ , con la igualdad y la adición definidas por los

coeficientes y la multiplicación por el uso de la ley distributiva y la regla de exponentes  $(x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n})(x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n}) = x_1^{i_1+j_1} x_2^{i_2+j_2} \cdots x_n^{i_n+j_n}$ .

Se nos presenta ahora la oportunidad de plantear el algoritmo de la división para  $R[x_1, \dots, x_n]$ , sin embargo, nos encontramos con una dificultad al intentar ordenar los términos de los polinomios. En  $R[x]$  esto es sencillo, pues sólo nos fijamos en el exponente de cada término y ordenamos de mayor a menor, pero si intentamos esto para polinomios de varias variables, podemos incurrir en empates de grado y no podríamos determinar que término se ubica antes del otro. Por ejemplo:

$$f(x, y, z) = 2x^3y^2z + 2y^3z^3 - 3xyz + y^2$$

presenta un empate en el primer y segundo término, ya que sus grados respectivos son  $3+2+1=6$  y  $3+3=6$ , y así no estaríamos seguros de presentar dicho polinomio como en la forma anterior o como  $f(x, y, z) = 2y^3z^3 + 2x^3y^2z - 3xyz + y^2$ .

Esta dificultad puede ser resuelta si definimos una forma de ordenamiento específica, en la cual no puedan existir empates.

## 2.2. Ordenes monomiales

La necesidad de especificar un orden monomial sobre los términos de  $k[x_1, \dots, x_n]$  es de gran importancia, y en esta sección estudiaremos como dar un orden a los monomios.

**Definición 2.2.1** *Un orden monomial en  $k[x_1, \dots, x_n]$  es una relación  $>$  que satisface:*

1.  $>$  es un orden total en  $k[x_1, \dots, x_n]$ .
2.  $x^\beta > 1$ , para todo  $x^\beta \in k[x_1, \dots, x_n], x^\beta \neq 1$ .
3. Si  $x^\alpha > x^\beta$ , entonces  $x^\alpha x^\lambda > x^\beta x^\lambda$ , para todo  $x^\lambda \in k[x_1, \dots, x_n]$ .

**Definición 2.2.2** *Sea  $x^a, x^b \in k[x_1, \dots, x_n]$ , donde  $a = (a_1, \dots, a_n)$ ,  $b = (b_1, \dots, b_n)$  y  $x_1 > x_2 > \cdots > x_n$ , definimos el orden lexicográfico en  $k[x_1, \dots, x_n]$  como:*

$$x^a > x^b \Leftrightarrow \{ \text{existe } 1 \leq i \leq n \text{ tal que } a_j = b_j \text{ para } 1 \leq j \leq i-1 \text{ y } a_i > b_i \}$$

Por ejemplo, utilizando el orden lexicográfico con  $x > y$  se tiene:

$$\cdots > x^2 > \cdots > xy^2 > xy > x > y^n > \cdots > y^3 > y^2 > y > 1$$

Ahora, si  $x < y$ , se tiene que:

$$1 < x < x^2 < x^3 < \cdots < x^n < y < xy < x^2y < \cdots < y^2 < \cdots$$

Al orden lexicográfico lo denotamos por *lex*.

**Definición 2.2.3** Sea  $x^a, x^b \in k[x_1, \dots, x_n]$ , donde  $a = (a_1, \dots, a_n)$ ,  $b = (b_1, \dots, b_n)$  y  $x_1 > x_2 > \cdots > x_n$ , definimos el orden lexicográfico graduado en  $k[x_1, \dots, x_n]$  como:

$$x^a > x^b \Leftrightarrow \left\{ \begin{array}{l} \sum_{i=1}^n a_i > \sum_{i=1}^n b_i \\ o \\ \sum_{i=1}^n a_i = \sum_{i=1}^n b_i, x^a > x^b \text{ con respecto al orden } lex \end{array} \right\}$$

En este caso, si utilizamos el orden lexicográfico graduado con  $x > y$  vemos que:

$$\cdots > xy^2 > y^3 > x^2 > xy > y^2 > x > y > 1$$

Teniendo  $x < y$  obtenemos:

$$1 < x < y < x^2 < xy < y^2 < x^3 < x^2y < \cdots$$

Denotaremos al orden lexicográfico graduado por *deglex*.

**Definición 2.2.4** Sea  $x^a, x^b \in k[x_1, \dots, x_n]$ , donde  $a = (a_1, \dots, a_n)$ ,  $b = (b_1, \dots, b_n)$  y  $x_1 > x_2 > \cdots > x_n$ , definimos el orden lexicográfico graduado inverso en  $k[x_1, \dots, x_n]$  como:

$$x^a > x^b \Leftrightarrow \left\{ \begin{array}{l} \sum_{i=1}^n a_i > \sum_{i=1}^n b_i \\ o \\ \sum_{i=1}^n a_i = \sum_{i=1}^n b_i, x^a > x^b \text{ entonces existe } 1 \leq i \leq n \\ \text{con } a_j = b_j \text{ para } i+1 \leq j \leq n \text{ y } a_i > b_i \end{array} \right\}$$

Al orden lexicográfico graduado inverso lo denotamos por *degrevlex*.

Notación: Dado un término ordenado en  $k[x_1, \dots, x_n]$ , entonces para todo  $0 \neq f \in k[x_1, \dots, x_n]$  podemos escribir a  $f$  como:

$$f = a_1x^{\alpha_1} + \dots + a_rx^{\alpha_r}$$

donde  $0 \neq a_i \in k$  y  $x^{\alpha_i} \in k[x_1, \dots, x_n]$  para todo  $i = 1, \dots, r$  y  $x^{\alpha_1} > x^{\alpha_2} > \dots > x^{\alpha_r}$ , definimos:

- $ml(f) = x^{\alpha_1}$ , el monomio líder de  $f$ .
- $cl(f) = a_1$ , el coeficiente líder de  $f$ .
- $tl(f) = a_1x^{\alpha_1}$ , el término líder de  $f$ .

**Ejemplo 1** Sea  $f = 2x^2yz + 3xy^3 - 2x^3$  entonces:

- Con el orden lex:  $ml(f) = x^3$ ,  $cl(f) = -2$ ,  $tl(f) = -2x^3$ .
- Con el orden deglex:  $ml(f) = x^2yz$ ,  $cl(f) = 2$ ,  $tl(f) = 2x^2yz$ .
- Con el orden degrevlex:  $ml(f) = xy^3$ ,  $cl(f) = 3$ ,  $tl(f) = 3xy^3$ .

Como podemos ver, el ordenamiento de los términos depende del orden monomial que se use, con los cuales, no existen empates.

### 2.3. Algoritmo de la división en $k[x_1, \dots, x_n]$

En la sección (2.1) estudiamos el algoritmo de la división para polinomios de una variable, lo cual hicimos como un proceso de reducción. Definiremos un algoritmo de la división en  $k[x_1, \dots, x_n]$  como una generalización de lo visto en la sección (2.1). La idea es básicamente la misma, cuando dividimos un polinomio  $f$  por  $f_1, \dots, f_s$ , queremos eliminar términos de  $f$  por los coeficientes líderes de los polinomios  $f_i$ , y de esta manera continuar hasta que ya no haya términos en  $f$  reducibles. Fijando un orden monomial en  $k[x_1, \dots, x_n]$ , tenemos lo siguiente.

**Definición 2.3.1** Dados  $f, g, h \in k[x_1, \dots, x_n]$ , con  $g \neq 0$ , decimos que  $f$  se reduce a  $h$  módulo  $g$ , y escribimos

$$f \xrightarrow{g} h$$

si y sólo si  $ml(g)$  divide a un término  $X$  que aparece en  $f$  y

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

No necesariamente el polinomio  $h$  de la definición es reducido con respecto a  $g$ , es decir, posiblemente  $f$  se reduzca a  $h_1$  módulo  $g$ ,  $h_1$  se reduzca a  $h_2$  módulo  $g$ , y así un número finito de reducciones.

**Definición 2.3.2** Sean  $f, h$  y  $f_1, \dots, f_s$  polinomios en  $k[x_1, \dots, x_n]$ , con  $f_i \neq 0$  para  $1 \leq i \leq s$ , y  $F = \{f_1, \dots, f_s\}$ . Decimos que  $f$  se reduce a  $h$  módulo  $F$ , y escribimos

$$f \xrightarrow{F} {}_+h$$

si y sólo si existe una sucesión de índices  $i_1, \dots, i_t \in \{1, \dots, s\}$  y una sucesión de polinomios  $h_1, \dots, h_{t-1} \in k[x_1, \dots, x_n]$  tales que

$$f \xrightarrow{f_{i_1}} h_1 \xrightarrow{f_{i_2}} \dots \xrightarrow{f_{i_{t-1}}} h_{t-1} \xrightarrow{f_{i_t}} h$$

**Ejemplo 2** Sea  $f_1 = yx - y, f_2 = y^2 - x \in Q[x, y]$ . Sea el orden monomial deglex con  $y > x$ , y sea  $F = \{f_1, f_2\}$ ,  $f = y^2x$ . Entonces

$$f \xrightarrow{F} {}_+x$$

ya que

$$y^2x \xrightarrow{f_1} y^2 \xrightarrow{f_2} x.$$

**Definición 2.3.3** Un polinomio  $r$  es llamado reducido con respecto a un conjunto de polinomios no cero  $F = \{f_1, \dots, f_s\}$ , si  $r = 0$  o si ningún término de  $r$  es divisible por algún  $ml(f_i)$ ,  $i = 1, \dots, s$ , es decir  $r$  no puede reducirse módulo  $F$ .

**Definición 2.3.4** Si  $f \xrightarrow{F} {}_+r$  y  $r$  es reducido con respecto a  $F$ , entonces llamamos a  $r$  un residuo para  $f$  con respecto a  $F$ .

Este proceso de reducción da pauta para definir un algoritmo de la división que imita el algoritmo visto en  $R[x]$ . Dados  $f, f_1, \dots, f_s \in k[x_1, \dots, x_n]$  con  $f_i \neq 0$ , para  $0 \leq i \leq s$ , este algoritmo regresa cocientes  $u_1, \dots, u_s \in k[x_1, \dots, x_n]$ , y un residuo  $r \in k[x_1, \dots, x_n]$ , tales que

$$f = u_1f_1 + \dots + u_sf_s + r$$

Dicho algoritmo está dado como sigue:

*Entrada* :  $f, f_1, \dots, f_s \in k[x_1, \dots, x_n]$  con  $f_i \neq 0$  ( $1 \leq i \leq s$ ).

*Salida* :  $u_1, \dots, u_s, r$  tales que  $f = u_1 f_1 + \dots + u_s f_s + r$  y  $r$  es reducido con respecto a  $\{f_1, \dots, f_s\}$  y  $ml(f) = \max(ml(u_1)ml(f_1), \dots, ml(u_s)ml(f_s), ml(r))$ .

*Inicialización* :  $u_1 := 0, u_2 := 0, \dots, u_s := 0, r := 0, h := f$ .

*Mientras* :  $h \neq 0$

*Si* : existe  $i$  tal que  $ml(f_i)$  divide a  $ml(h)$ , entonces elija  $i$  mínimo tal que  $ml(f_i)$  divida a  $ml(h)$

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

$$h := h - \frac{tl(h)}{tl(f_i)} f_i$$

*Si no* :

$$r := r + tl(h)$$

$$h := h - tl(h)$$

Notemos que en el algoritmo anterior consideramos un orden monomial para el conjunto de polinomios  $\{f_1, \dots, f_s\}$  cuando elegimos el menor  $i$  tal que  $ml(f_i)$  divide a  $ml(h)$ .

Las similitudes entre el algoritmo de la división en una variable y en varias variables, son las siguientes: Los cocientes  $u_1, \dots, u_s$  en el algoritmo anterior corresponden al cociente  $q$  en el algoritmo visto en la sección (2.1). Obviamente el algoritmo anterior produce  $s$  cocientes porque  $f$  es dividido por  $s$  diferentes polinomios  $f_1, \dots, f_s$ . El residuo  $r$  en ambos casos tiene el mismo significado: ningún término de  $r$  es divisible por el término líder de algún divisor. En el algoritmo de la sección (2.1) el término líder de  $r$  no es divisible por  $tl(g)$ , y sabemos que ningún otro término de  $r$  es divisible por  $tl(g)$  y así se obtiene el residuo. En el algoritmo para  $k[x_1, \dots, x_n]$  empezamos con  $h = f$  y  $r = 0$  y sustraemos términos líderes de  $h$ ; cuando no podamos dividir dicho término por el término líder del  $f_i$  correspondiente, este término líder se adiciona al residuo, y de esta manera construimos el residuo.

Finalmente, hemos de considerar muy importante el hecho de que la forma en que se ordenan los  $f_i$  interfiere en el proceso de reducción y en el residuo final mismo. Para aclarar esto, consideremos el ejemplo 2 con el orden *deglex*

y  $y > x$ , pero ahora consideremos a  $f_1 = y^2 - x$  y  $f_2 = yx - y$  y a  $f = y^2x$  como antes. Entonces, al hacer la reducción obtenemos:

$$f \xrightarrow{F} +x^2$$

ya que

$$y^2x \xrightarrow{f_1} +x^2$$

y  $x^2$  no es divisible por  $f_2$  en este caso.

Entonces, para erradicar este problema de residuos múltiples, hemos de aplicar un orden monomial primero a cada  $f_i$  y después, para ordenar estos polinomios, aplicamos el mismo orden entre los términos líderes de los distintos  $f_i$ . En caso de empate, aplicamos el orden respecto a los segundos términos de los polinomios empatados, si de nuevo hay empates, el orden monomial será respecto a los terceros términos y así sucesivamente. Por ejemplo, en el caso anterior con *deglex* y  $y > x$  tenemos que los  $f_i$  ordenados de manera correcta corresponden a:

$$f_1 = y^2 - x \qquad f_2 = yx - y$$

Esta forma de ordenar, tanto a los términos de cada  $f_i$  como a los mismos  $f_i$ , nos será de gran utilidad en el capítulo siguiente para explicar la mecánica de las bases de Gröbner.



# Capítulo 3

## Bases de Gröbner

En este capítulo daremos los fundamentos teóricos para definir las Bases de Gröbner, que serán nuestra herramienta para trabajar en nuestro estudio sobre la programación entera.

### 3.1. Preliminares

De la sección (2.1) tenemos la definición de lo que es un anillo. Ahora en esta, daremos a conocer algunos otros conceptos que nos serán de utilidad para entender lo que son las Bases de Gröbner.

**Definición 3.1.1** *Un conjunto  $G$  con los elementos  $a, b, c$  constituye un grupo si para estos elementos se ha definido una operación de enlace « $\circ$ », de manera que se verifican:*

1.  $a \circ b \in G$  (cerradura).
2.  $(a \circ b) \circ c = a \circ (b \circ c)$  (propiedad asociativa).
3. Existe  $e \in G$  tal que  $a \circ e = a$  para todo  $a \in G$  (elemento neutro).
4. Para cada elemento  $a \in G$  existe un único elemento  $a^{-1} \in G$  tal que  $a \circ a^{-1} = e$  (elemento inverso).

*Si también se verifica la propiedad conmutativa  $a \circ b = b \circ a$ , entonces se trata de un grupo abeliano.*

**Definición 3.1.2** *Un subgrupo es un subconjunto de un grupo que, a su vez, es un grupo. Los subgrupos tienen siempre en común con el grupo el mismo elemento neutro.*

**Definición 3.1.3** *Un subconjunto no vacío  $U$  de  $R$  se dice que es un ideal de  $R$  si:*

1.  $U$  es un subgrupo de  $R$  bajo la adición.
2. Para todo  $u \in U$  y  $r \in R$ , tanto  $u \cdot r$  como  $r \cdot u$  están en  $U$ .

La notación que se usa es  $I = \langle f_1, \dots, f_m \rangle$ , y decimos que  $I$  es el ideal generado por las  $f_i$ . Veremos ahora qué pasa si tenemos un ideal  $I$  de  $k[x]$  el cual es generado por  $m$  elementos.

**Proposición 3.1.4** *Sea  $f_1, \dots, f_m$  polinomios en  $k[x]$ . Entonces*

1.  $\langle f_1, \dots, f_m \rangle = \langle \text{mcd}(f_1, \dots, f_m) \rangle$
2. Si  $m \geq 3$ , entonces  $\text{mcd}(f_1, \dots, f_m) = \text{mcd}(f_1, \text{mcd}(f_2, \dots, f_m))$ .

Podemos decir ahora que en el anillo  $k[x]$ , todo ideal  $I$  puede ser generado por un elemento, en este caso, por el *mcd* de los generadores de  $I$ . Desde este punto de vista, las bases de Gröbner serán una generalización del *mcd* en el anillo  $k[x_1, \dots, x_n]$  de la siguiente forma: Una base de Gröbner para un ideal  $I$  de  $k[x_1, \dots, x_n]$ , será un conjunto de polinomios  $G$  que generen a  $I$ , además, dado  $f \in k[x_1, \dots, x_n]$ , tenemos que  $f \in I$  si y sólo si el residuo de dividir  $f$  por los polinomios de  $G$  es cero.

Para explicar los teoremas y definiciones que a continuación se presentan, supondremos que existe un orden monomial respecto al cual se enuncian dichas afirmaciones.

**Definición 3.1.5** *Sea  $G = \{g_1, \dots, g_t\} \neq \emptyset$  un conjunto de polinomios contenido en un ideal  $I$ .  $G$  es una base de Gröbner si y sólo si para todo  $0 \neq f \in I$ , existe  $i \in \{1, \dots, t\}$  tal que  $\text{ml}(g_i)$  divide a  $\text{ml}(f)$ .*

**Definición 3.1.6** *Para un conjunto  $S$  de  $k[x_1, \dots, x_n]$  definimos el ideal de términos líderes de  $S$  como:*

$$Tl(S) = \langle \text{tl}(s) \mid s \in S \rangle$$

**Teorema 3.1.7** *Sea  $I$  un ideal no cero de  $k[x_1, \dots, x_n]$ . Las siguientes afirmaciones son equivalentes para un conjunto no cero de polinomios  $G = \{g_1, \dots, g_t\} \subseteq I$ :*

1.  $G$  es una base de Gröbner para  $I$ .
2.  $f \in I$  si y sólo si  $f \xrightarrow{G} 0$
3.  $f \in I$  si y sólo si  $f = \sum_{i=1}^t h_i g_i$  con  $ml(f) = \max_{1 \leq i \leq t} (ml(h_i) \cdot ml(g_i))$ .
4.  $Tl(G) = Tl(I)$ .

**Lema 3.1.8** *Sea  $I$  un ideal generado por un conjunto  $S$  de términos no cero, y sea  $f \in k[x_1, \dots, x_n]$ . Entonces  $f$  está en  $I$  si y sólo si para todo término  $X$  que aparece en  $f$  existe  $Y \in S$  tal que  $Y$  divide a  $X$ . Más aún, existe un subconjunto  $S_0$  finito tal que  $I = \langle S_0 \rangle$ .*

**Demostración.** Si  $f \in I$ , entonces

$$tl(f) = \sum_{i=1}^l h_i X_i$$

donde  $h_i \in k[x_1, \dots, x_n]$  y  $X_i \in S$ , para  $i = 1, \dots, l$ . Si expandimos el lado derecho de la ecuación de arriba, vemos que todo término es divisible por algún término  $X_i \in S$ , y por lo tanto, todo término del lado izquierdo debe también ser divisible por algunos términos  $X_i \in S$ . Recíprocamente, si para todo término  $X$  que aparece en  $f$  existe un término  $Y \in S$  tal que  $Y$  divide a  $X$ , entonces cada  $X \in I = \langle S \rangle$  y por lo tanto  $f \in I$ . Para probar la última parte, vemos que por el teorema de la base de Hilbert,  $I$  tiene un conjunto finito  $S_0$  de generadores. Por la primera parte del lema, cada término que aparece en un polinomio de  $f$  en  $S_0$  es dividido por algún término en  $S$ . El conjunto finito  $S'_0$  de todos los divisores es claramente un conjunto generador para  $I$ . ■

**Corolario 3.1.9** *Todo ideal  $I$  no cero de  $k[x_1, \dots, x_n]$  tiene una base de Gröbner.*

**Demostración.** Por el lema (3.1.8), el ideal  $Tl(I)$  tiene un conjunto finito de generadores, el cual podemos asumir que es de la forma  $\{tl(g_1), \dots, tl(g_t)\}$  con  $g_1, \dots, g_t \in I$ . Si escribimos  $G = \{g_1, \dots, g_t\}$ , entonces tenemos que  $Tl(G) = Tl(I)$  y por la definición (3.1.5)  $G$  es base de Gröbner. ■

**Teorema 3.1.10** Sea  $G = \{g_1, \dots, g_t\}$  un conjunto no cero de polinomios en  $k[x_1, \dots, x_n]$ . Entonces  $G$  es una base de Gröbner si y sólo si para todo  $f \in k[x_1, \dots, x_n]$  el residuo de la división de  $f$  por  $G$  es único.

## 3.2. S-Polinomios y el algoritmo de Buchberger

**Definición 3.2.1** Sea  $f \neq 0, g \in k[x_1, \dots, x_n]$ . Sea  $L = \text{lcm}(ml(f), ml(g))$ . El polinomio:

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

lo llamamos el  $S$ -polinomio de  $f$  y  $g$ .

Notemos que si  $I = \langle g_1, \dots, g_n \rangle$  con  $g_i \in k[x_1, \dots, x_n]; i = 1, \dots, n$ , entonces si  $i \neq j, S(g_i, g_j) \in I$ .

**Lema 3.2.2** Sea  $f_1, \dots, f_s \in k[x_1, \dots, x_n]$  tales que  $ml(f_i) = x \neq 0$  para todo  $i = 1, \dots, s$ . Sea  $f = \sum_{i=1}^s c_i f_i$  con  $c_i \in K$ . Si  $ml(f) < x$  entonces  $f$  es una combinación lineal de  $S(f_i, f_j), 1 \leq i < j \leq s$  con coeficientes en  $K$ .

**Demostración.** Para todo  $i = 1, \dots, s$  escribimos  $f_i = a_i X + \text{términos menores}$ , por hipótesis tenemos que  $\sum_{i=1}^s c_i a_i = 0$  y  $S(f_i, f_j) = \frac{1}{a_i} f_i - \frac{1}{a_j} f_j; i \neq j$ . Ahora escribimos a  $f$  de la forma  $f = c_1 f_1 + \dots + c_s f_s$ . Esto se logra haciendo

$$\begin{aligned} f &= c_1 a_1 \left( \frac{1}{a_1} f_1 - \frac{1}{a_2} f_2 \right) + (c_1 a_1 + c_2 a_2) \left( \frac{1}{a_2} f_2 - \frac{1}{a_3} f_3 \right) + \dots \\ &\quad + (c_1 a_1 + \dots + c_{s-1} a_{s-1}) \left( \frac{1}{a_{s-1}} f_{s-1} - \frac{1}{a_s} f_s \right) + (c_1 a_1 + \dots + c_s a_s) \end{aligned}$$

y así tenemos

$$f = c_1 a_1 S(f_1, f_2) + (c_1 a_1 + c_2 a_2) S(f_2, f_3) + \dots + (c_1 a_1 + \dots + c_{s-1} a_{s-1}) S(f_{s-1}, f_s)$$

■

**Teorema 3.2.3 (Buchberger)** Sea  $G = \{g_1, \dots, g_t\} \neq \emptyset$  donde cada  $g_i$  está en  $k[x_1, \dots, x_n]$ . Entonces  $G$  es una base de Gröbner para el ideal  $I = \langle g_1, \dots, g_t \rangle$  si y sólo si para todo  $i \neq j, S(g_i, g_j) \xrightarrow{G} 0$ .

Observamos que el teorema de Buchberger da una estrategia para calcular Bases de Gröbner, si reducimos los S-polinomios y si el residuo es no cero, entonces adicionamos el residuo al conjunto generador.

**Ejemplo 3** Sea  $f_1 = xy - x, f_2 = x^2 - y \in Q[x, y]$ , con el orden monomial deglex con  $x < y$ . Sea  $F = \{f_1, f_2\}$ . Entonces  $S(f_1, f_2) = xf_1 - yf_2 = y^2 - x^2 \xrightarrow{F} y^2 - y$  y  $f_3 = y^2 - y$  es reducido con respecto a  $F$ . Así que adicionamos  $f_3$  a  $F$  y sea  $F' = \{f_1, f_2, f_3\}$ . Entonces  $S(f_1, f_2) \xrightarrow{F'} 0$ . Ahora  $S(f_1, f_3) = yf_1 - xf_3 = 0 \xrightarrow{F'} 0$  y  $S(f_2, f_3) = y^2f_2 - x^2f_3 = -y^3 + x^2y \xrightarrow{F'} 0$ . Así  $\{f_1, f_2, f_3\}$  es una base de Gröbner.

**Algoritmo 3.2.4 (Algoritmo de Buchberger)** Sea  $I = \langle f_1, \dots, f_s \rangle$  con  $f_i \neq 0; 1 \leq i \leq s$ , entonces una base de Gröbner puede ser construida en un número finito de pasos como sigue:

Input:  $F = f_1, \dots, f_s \in k[x_1, \dots, x_n]; f_i \neq 0 (1 \leq i \leq s)$

Output:  $G = g_1, \dots, g_t$ , una base de Gröbner para  $I$

$G := F$

$\mathcal{G} := \{f_i, f_j | f_i \neq f_j \in G\}$

Mientras  $\mathcal{G} \neq \emptyset$

    buscar algún  $\{f, g\} \in \mathcal{G}$

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

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

    Si  $h \neq 0$  Entonces

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

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

### 3.3. Bases de Gröbner reducidas

En la sección pasada vimos como calcular bases de Gröbner. Sin embargo, estas bases de Gröbner no son necesariamente únicas. En esta sección daremos las condiciones necesarias para tener la unicidad de las bases de Gröbner.

**Definición 3.3.1** Una base de Gröbner  $G = \{g_1, \dots, g_t\}$  es llamada minimal con respecto a un orden monomial, si y sólo si para todo  $i, cl(g_i) = 1$  y para todo  $i \neq j, ml(g_i)$  no divide a  $ml(g_j)$ .

**Definición 3.3.2** Una base de Gröbner  $G = \{g_1, \dots, g_t\}$  es llamada reducida con respecto a un orden monomial, si para todo  $i$   $cl(g_i) = 1$  y  $g_i$  es reducido con respecto a  $G - g_i$ . Esto es, para todo  $i$ , ningún término en  $g_i$  es dividido por algún  $ml(g_j)$  para todo  $j \neq i$ .

**Teorema 3.3.3** Sea  $G = \{g_1, \dots, g_t\}$  una base de Gröbner minimal para el ideal  $I$ . Considere el siguiente proceso de reducción:

$$\begin{aligned} g_1 &\xrightarrow{H_1} + h_1, \text{ donde } h_1 \text{ es reducido con respecto a } H_1 = \{g_2, \dots, g_t\} \\ g_2 &\xrightarrow{H_2} + h_2, \text{ donde } h_2 \text{ es reducido con respecto a } H_2 = \{h_1, g_3, \dots, g_t\} \\ g_3 &\xrightarrow{H_3} + h_3, \text{ donde } h_3 \text{ es reducido con respecto a } H_3 = \{h_1, h_2, g_4, \dots, g_t\} \\ &\vdots \\ g_t &\xrightarrow{H_t} + h_t, \text{ donde } h_t \text{ es reducido con respecto a } H_t = \{h_1, h_2, \dots, h_{t-1}\} \end{aligned}$$

Entonces  $H = \{h_1, \dots, h_t\}$  es una base de Gröbner reducida para  $I$ .

**Demostración.** Note que, como  $G$  es una base de Gröbner minimal, entonces  $ml(g_i) = ml(h_i)$  para cada  $i = 1, \dots, t$ , así,  $H$  es una base de Gröbner minimal para  $I$ . Como la división de  $g_i$  por los elementos del conjunto  $H_i$  es hecha por eliminación de términos de  $g_i$  usando a  $ml(h_1), \dots, ml(h_{i-1}), ml(g_{i+1}), \dots, ml(g_t)$  y como  $ml(g_i) = ml(h_i)$  entonces  $H$  es una base de Gröbner reducida. ■

**Lema 3.3.4** Dado un orden monomial, todo ideal  $I$  no cero tiene una única base de Gröbner reducida con respecto a ese orden monomial.

**Demostración.** Sea  $H = \{h_1, \dots, h_t\}$  y  $G = \{g_1, \dots, g_t\}$  bases de Gröbner reducidas para  $I$ . Tenemos que  $ml(g_i) = ml(h_i)$  por la demostración del teorema anterior. Supongamos que  $h_i \neq g_i$  para algún  $i \in \{1, \dots, t\}$ , entonces  $g_i - h_i \in I$ , por lo tanto existe  $j \neq i$  tal que  $ml(h_j)$  divide a  $ml(g_i - h_i)$  que implica que  $ml(h_j) = ml(g_j)$  divide a algún término  $X$  que aparece en  $g_i$  o en  $h_i$ , lo cual es una contradicción al hecho de que  $G$  y  $H$  son bases de Gröbner reducidas. Así,  $g_i = h_i$  para cualquier  $i \in \{1, \dots, t\}$ . Por lo tanto  $H = G$ . ■

Ahora bien, ya presentada la teoría necesaria, soporte de las aplicaciones que a continuación se exponen, procederemos a desarrollar el tema que nos concierne en este trabajo, explicando las características generales, el proceso y la base matemática en que se sustentan los distintos métodos.

Presentamos también, otra aplicación del álgebra conmutativa y las bases de Gröbner como una referencia del poder y belleza de estos conceptos.

# Capítulo 4

## Programación lineal entera

La finalidad de este capítulo es exponer, de la manera más clara posible, el tema de esta tesis: la forma en que se pueden aplicar las bases de Gröbner en la resolución de problemas de programación entera. Además, indicar la teoría que sustenta al método y presentar algunos ejemplos resueltos de distintos tipos de problemas de programación entera.

De igual manera, se describe otra aplicación de las bases de Gröbner en problemas de enumeración, los cuales tienen cierto parecido con los de programación entera.

### 4.1. Programación entera

Algunos problemas de programación entera pueden resolverse mediante distintos métodos, que han sido desarrollados para cada tipo de problema.

Aún no existe el método o algoritmo para el caso general de cualquier problema de programación entera. Sin embargo, y gracias a los trabajos realizados por P. Conti y C. Traverso, se ha podido avanzar en la consecución de este algoritmo general.

Es posible cambiar un problema de programación entera por un problema sobre polinomios de varias variables, mediante una traslación. Podemos trabajar en anillos polinómicos mediante el uso de las bases de Gröbner, y reinterpretar ciertos resultados particularmente importantes, o de atención específica - por decirlo de otra forma - para dar una solución a nuestro problema inicial de programación entera.





La imagen de  $\varphi$  en la Proposición (4.1.1) es precisamente el conjunto de polinomios en  $k[z_1, \dots, z_m]$  que pueden ser expresados como polinomios en las  $f_j = \prod_{i=1}^m z_i^{a_{ij}}$ , también podemos escribir la imagen como  $k[f_1, \dots, f_n]$ , el subanillo de  $k[z_1, \dots, z_m]$  generado por las  $f_j$ .

**Proposición 4.1.2** *Suponga que  $f_1, \dots, f_n \in k[z_1, \dots, z_m]$  están dados. Fije un orden monomial en  $k[z_1, \dots, z_m, \omega_1, \dots, \omega_n]$  con la propiedad de eliminación: cualquier monomio que contenga uno de los  $z_i$  es mayor que cualquier monomio que contenga sólo los  $\omega_j$ . Sea  $G$  una base de Gröbner para el ideal*

$$I = \langle f_1 - \omega_1, \dots, f_n - \omega_n \rangle \subset k[z_1, \dots, z_m, \omega_1, \dots, \omega_n]$$

y para cada  $f \in k[z_1, \dots, z_m]$ , sea  $\bar{f}^G$  el residuo en la división de  $f$  por  $G$ . Entonces

- Un polinomio  $f$  satisface  $f \in k[f_1, \dots, f_n]$  si y sólo si  $g = \bar{f}^G \in k[\omega_1, \dots, \omega_n]$ .
- Si  $f \in k[f_1, \dots, f_n]$  y  $g = \bar{f}^G \in k[\omega_1, \dots, \omega_n]$  como en la parte a, entonces  $f = g(f_1, \dots, f_n)$ , dando una expresión para  $f$  como un polinomio en las  $f_j$ .
- Si cada  $f_j$  y  $f$  son monomios y  $f \in k[f_1, \dots, f_n]$ , entonces  $g$  también es un monomio.

La parte c) nos permite enunciar el siguiente lema:

**Lema 4.1.3** *Si  $z_1^{b_1} \cdots z_m^{b_m}$  está en la imagen de  $\varphi$ , entonces es la imagen del monomio  $\omega_1^{A_1} \cdots \omega_n^{A_n} \in k[\omega_1, \dots, \omega_n]$ .*

**Demostración.** Sea  $I = \langle f_1 - \omega_1, \dots, f_n - \omega_n \rangle$  el ideal generado para (4.1). Sea  $G$  una base de Gröbner para  $I$  con respecto a un orden de eliminación con las variables  $z_i$  mayores que las  $w_j$ , entonces:

$$z_1^{b_1} \cdots z_m^{b_m} \in \text{im}(\varphi) \iff z_1^{b_1} \cdots z_m^{b_m} \xrightarrow{G} {}_+ h \text{ con } h \in k[\omega_1, \dots, \omega_n]$$

Más aún, si  $z_1^{b_1} \cdots z_m^{b_m} \xrightarrow{G} {}_+ h$  con  $h \in k[\omega_1, \dots, \omega_n]$  entonces  $z_1^{b_1} \cdots z_m^{b_m} = \varphi(h)$ .

Notemos primero que los polinomios que generan  $I$  son todos diferencias de dos monomios. Por eso, durante el algoritmo de Buchberger (3.2.4) para calcular  $G$ , sólo son generados polinomios que son diferencias de dos

monomios. Así, el S-polinomio de dos polinomios que son ambos diferencias de dos monomios, es de igual modo una diferencia de dos monomios, y el primer paso en la reducción de un polinomio que es una diferencia de dos monomios por otro polinomio de la misma forma produce un polinomio que es también una diferencia de dos monomios. Por eso los polinomios en  $G$  son todos diferencias de dos monomios. Ahora si  $z_1^{b_1} \cdots z_m^{b_m}$  está en la imagen de  $\varphi$ , entonces se reduce a un polinomio  $h \in k[\omega_1, \dots, \omega_n]$ . Pero el primer paso de la reducción de un monomio por un polinomio que es una diferencia de dos monomios, produce un monomio. Por eso  $h$  es un monomio. ■

La demostración de (4.1.3) nos da un método para determinar cuando tiene solución el sistema (4.1), y para encontrar una solución (el algoritmo se presenta más adelante):

1. Calcular una base de Gröbner para  $I = \langle f_1 - w_1, \dots, f_n - w - n \rangle$  con respecto a un orden de eliminación con las  $z_i$  mayores que las  $w_j$ ;
2. Encontrar el residuo  $h$  de la división del monomio  $z_1^{b_1} \cdots z_m^{b_m}$  por  $G$ ;
3. Si  $h \notin k[\omega_1, \dots, \omega_n]$ , entonces el sistema (4.1) no tiene solución entera. Si  $h = w_1^{A_1} \cdots w_n^{A_n}$ , entonces  $(A_1, A_2, \dots, A_n)$  es una solución del sistema (4.1).

Para aclarar un poco esto, resolveremos el siguiente ejemplo utilizando estas ideas:

$$\text{Max: } 11A + 15B$$

s.a.:

$$\begin{aligned} 4A + 5B &\leq 37 \\ 2A + 3B &\leq 20 \\ A, B &\in \mathbf{Z}_{\geq 0} \end{aligned} \tag{4.3}$$

En la figura 4.1 podemos ver la caracterización gráfica de los puntos enteros que cumplen con las restricciones de este problema.

Ahora consideremos este problema en su forma estándar, con variables de holgura  $C$  en la primera ecuación y  $D$  en la segunda.

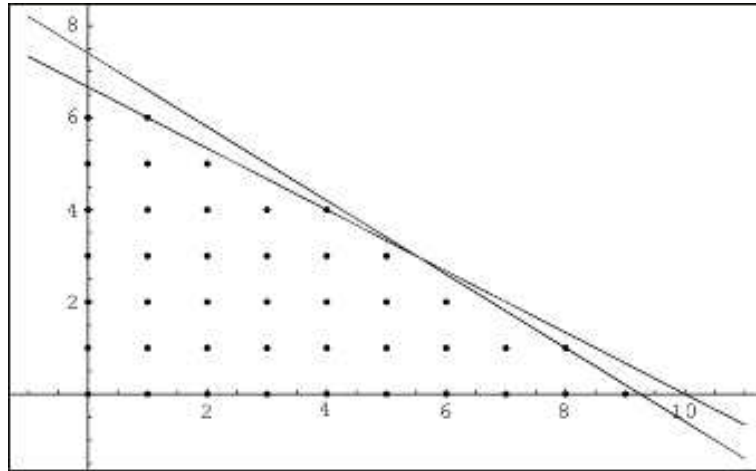


Figura 4.1: Puntos enteros factibles del problema (4.3)

Max:  $11A + 15B$

s.a.:

$$\begin{aligned} 4A + 5B + C &= 37 \\ 2A + 3B + D &= 20 \end{aligned} \tag{4.4}$$

$$A, B, C, D \in \mathbf{Z}_{\geq 0}$$

Tenemos por definición en la Proposición (4.1.1)

$$\begin{aligned} \varphi : k[\omega_1, \omega_2, \omega_3, \omega_4] &\rightarrow k[z_1, z_2] \\ \omega_1 &\mapsto z_1^4 z_2^2 \\ \omega_2 &\mapsto z_1^5 z_2^3 \\ \omega_3 &\mapsto z_1 \\ \omega_4 &\mapsto z_2 \end{aligned} \tag{4.5}$$

Los puntos enteros en la región factible de este arreglo de el problema, son los  $(A, B, C, D)$  tal que

$$\varphi(\omega_1^A \omega_2^B \omega_3^C \omega_4^D) = z_1^{37} z_2^{20} = f$$

Y entonces podemos expresar las  $f_j$  de la siguiente manera:

$$\begin{aligned} f_1 &= z_1^4 z_2^2 & f_2 &= z_1 \\ f_2 &= z_1^5 z_2^3 & f_4 &= z_2 \end{aligned}$$

Según la Proposición (4.1.2) tendríamos definido el ideal como:

$$I = \langle z_1^4 z_2^2 - \omega_1, z_1^5 z_2^3 - \omega_2, z_1 - \omega_3, z_2 - \omega_4 \rangle$$

Usando el paquete para computadora *Macaulay 2* obtenemos los generadores de la base de Gröbner minimal para el ideal  $I$ :

$$\begin{aligned} g_1 &= z_1 - \omega_3, \\ g_2 &= z_2 - \omega_4, \\ g_3 &= \omega_4^2 \omega_3^4 - \omega_1, \\ g_4 &= \omega_4 \omega_3^3 \omega_2 - \omega_1^2, \\ g_5 &= \omega_4 \omega_3 \omega_1 - \omega_2, \\ g_6 &= \omega_4 \omega_1^4 - \omega_3 \omega_2^3, \\ g_7 &= \omega_3^2 \omega_2^2 - \omega_1^3, \end{aligned}$$

Procedemos a la reducción de  $f$  por  $G$  y el resultado, denotado en (4.1.2) por  $\bar{f}^G$  es:

$$\bar{f}^G = \omega_1^4 \omega_2^4 \omega_3$$

Este monomio corresponde a la solución del problema de programación entera (4.4) donde  $(A = 4, B = 4, C = 1, D = 0)$ . En un sentido, esto es un accidente, ya que el orden lexicográfico que usamos para el cálculo de la base de Gröbner y el residuo no toma en cuenta la función objetivo  $l$  explícitamente. Para encontrar la solución de un problema de programación entera minimizando una función  $l(A_1, \dots, A_n)$  dada, usualmente necesitamos usar un orden monomial diseñado específicamente para el problema.

**Definición 4.1.4** *Un orden monomial en  $k[z_1, \dots, z_m, \omega_1, \dots, \omega_n]$  se dice que es adaptado para un problema de programación entera (4.1) si tiene las siguientes dos propiedades:*

1. *(Eliminación) Cualquier monomio que contenga una de las  $z_i$  es mayor que cualquier monomio que sólo contenga  $\omega_j$ .*
2. *(Compatibilidad con  $l$ ) Sea  $A = (A_1, \dots, A_n)$  y  $A' = (A'_1, \dots, A'_n)$ . si los monomios  $\omega^A, \omega^{A'}$  satisfacen  $\varphi(\omega^A) = \varphi(\omega^{A'})$  y  $l(A_1, \dots, A_n) > l(A'_1, \dots, A'_n)$  entonces  $\omega^A > \omega^{A'}$ .*

**Teorema 4.1.5** *Considere un problema de programación entera en la forma estándar (4.1). Asúmanse todos los  $a_{ij}, b_i \geq 0$  y sea  $f_j = \prod_{i=1}^m z_i^{a_{ij}}$  como antes. Sea  $G$  una base de Gröbner para:*

$$I = \langle f_1 - \omega_1, \dots, f_n - \omega_n \rangle \subset k[z_1, \dots, z_m, \omega_1, \dots, \omega_n]$$

*con respecto a cualquier orden monomial adaptado. Entonces si  $f = z_1^{b_1} \dots z_m^{b_m}$  está en  $k[f_1, \dots, f_n]$ , el residuo  $\bar{f}^G \in k[\omega_1, \dots, \omega_n]$  dará una solución de (4.1) minimizando  $l$ .*

**Demostración.** Sea  $G$  una base de Gröbner para  $I$  con respecto a cualquier orden monomial adaptado. Suponga que  $\omega^A = \bar{f}^G$  así que  $\varphi(\omega^A) = f$ , pero que  $A = (A_1, \dots, A_n)$  no es un mínimo de  $l$ . Esto es, suponga que existe algún  $A' = (A'_1, \dots, A'_n) \neq A$  tal que  $\varphi(\omega^{A'}) = f$  y  $l(A'_1, \dots, A'_n) < l(A_1, \dots, A_n)$ . Considere la diferencia  $h = \omega^A - \omega^{A'}$ . Tenemos  $\varphi(h) = f - f = 0$ . Esto implica que  $h \in I$ . Pero entonces  $h$  debe reducirse a cero bajo la base de Gröbner  $G$  para  $I$ . Sin embargo, ya que  $>$  es un orden adaptado, el último término de  $h$  debe ser  $\omega^A$ , y este monomio es reducido con respecto a  $G$  ya que es un residuo. Esta contradicción muestra que  $A$  debe dar un mínimo de  $l$ . ■

**Algoritmo 4.1.6** El teorema (4.1.5) da origen a un algoritmo de bases de Gröbner para resolver problemas de programación entera con  $a_{ij}, b_i \geq 0$ :

Input:  $A, b$  de (4.1), un orden monomial adaptado  $>$

Output: una solución de (4.1), si existe alguna

$$f_j := \prod_{i=1}^m z_i^{a_{ij}}$$

$$I := \langle f_1 - \omega_1, \dots, f_n - \omega_n \rangle$$

$G :=$  Base de Gröbner de  $I$  con respecto a  $>$

$$f := \prod_{i=1}^m z_i^{b_i}$$

$$g := \bar{f}^G$$

Si  $g \in k[\omega_1, \dots, \omega_n]$  Entonces

este vector de exponentes da una solución

En otro caso

no hay solución

Los órdenes monomiales satisfacen ambas propiedades de eliminación y compatibilidad de (4.1.4) y pueden ser especificados en las maneras siguientes:

Primero, suponga que todos los  $c_j \geq 0$ . Entonces es posible definir un orden ponderado  $>_l$  en las variables  $\omega$  usando la función lineal  $l$  como sigue:

$$\omega_1^{A_1} \cdots \omega_n^{A_n} >_l \omega_1^{A'_1} \cdots \omega_n^{A'_n}$$

si  $l(A_1, \dots, A_n) > l(A'_1, \dots, A'_n)$  y elimina los empates usando cualquier otro orden monomial fijo en  $k[\omega_1, \dots, \omega_n]$ . Entonces incorporamos este orden monomial en el orden sobre  $k[z_1, \dots, z_m, \omega_1, \dots, \omega_n]$  con las variables  $z$  mayores que todas las variables  $\omega$  para asegurar que se tiene la propiedad de eliminación de (4.1.4).

Si algún  $c_j < 0$  el método produce un ordenamiento total en monomios en  $k[z_1, \dots, z_m, \omega_1, \dots, \omega_n]$  que es compatible con la multiplicación y que satisface la propiedad de eliminación; pero no será un buen orden. Así, para aplicar la teoría de bases de Gröbner con respecto a órdenes monomiales, necesitamos más práctica. Comenzaremos con la siguiente observación:

En  $k[z_1, \dots, z_m, \omega_1, \dots, \omega_n]$ , definimos un grado para cada variable poniendo  $\deg(z_i) = 1$  para todo  $i = 1, \dots, m$  y  $\deg(\omega_j) = d_j = \sum_{i=1}^m a_{ij}$  para todo  $j = 1, \dots, n$ . Cada  $d_j$  debe ser estrictamente positivo. Decimos que un polinomio  $f \in k[z_1, \dots, z_m, \omega_1, \dots, \omega_n]$  es homogéneo con respecto a estos grados si todos los monomios  $z^\alpha \omega^\beta$  que aparecen en  $f$  tienen el mismo grado total  $|\alpha| + \sum_j d_j \beta_j$ .

**Lema 4.1.7** *Con respecto a los grados  $d_j$  en  $\omega_j$ , se tienen las siguientes declaraciones:*

1. *El ideal  $I = \langle f_1 - \omega_1, \dots, f_n - \omega_n \rangle$  es homogéneo.*
2. *Cada base de Gröbner reducida para el ideal  $I$  consiste de polinomios homogéneos.*

**Demostración.** La parte 1 se sigue de que los generadores dados son homogéneos para estos grados, ya que  $f_j = \prod_{i=1}^m z_i^{a_{ij}}$ , los dos términos en  $f_j - \omega_j$  tienen el mismo grado.

La parte 2 se sigue en la misma forma como para los ideales que son homogéneos en el sentido usual. ■

Ya que  $d_j > 0$  para todo  $j$ , dados los  $c_j$  de  $l$  y  $\mu \geq 0$  suficientemente grande, todas la entradas del vector

$$(c_1, \dots, c_n) + \mu(d_1, \dots, d_n)$$

serán positivas. Sea  $\mu$  cualquier número fijo para el cual esto es verdadero. Considere los  $(m + n)$  componentes de los vectores de peso  $u_1, u_2$ :

$$\begin{aligned} u_1 &= (1, \dots, 1, 0, \dots, 0) \\ u_2 &= (0, \dots, 0, c_1, \dots, c_n) + \mu(0, \dots, 0, d_1, \dots, d_n). \end{aligned}$$

Entonces todas las entradas de  $u_2$  son no negativas, y podemos definir un orden ponderado  $>_{u_1, u_2, \sigma}$  comparando primero los pesos de  $u_1$ , comparamos los de  $u_2$  si los de  $u_1$  son iguales, y finalmente, eliminamos los empates con cualquier otro orden monomial  $>_\sigma$ .

Por ejemplo, nuestro problema (4.4) puede ser resuelto usando el segundo método. Tomamos  $u_1 = (1, 1, 0, 0, 0, 0)$ , y siendo  $\mu = 2$ , vemos que:

$$u_2 = (0, 0, -11, -15, 0, 0) + 2(0, 0, 6, 8, 1, 1) = (0, 0, 1, 1, 2, 2)$$

tiene todas la entradas no negativas. Finalmente, se eliminan los empates con  $>_\sigma = \text{degrevlex}$  sobre todas las variables ordenadas  $z_1 > z_2 > \omega_1 > \omega_2 > \omega_3 > \omega_4$ . Esto concluye las especificaciones de órdenes adaptados y el tratado referente al caso particular donde todas las constantes son positivas.

#### 4.1.2. Caso general: algunas constantes negativas

Ahora, queremos discutir problemas de programación entera generales donde algunos de los  $a_{ij}, b_i$  pueden ser negativos. No hay diferencia conceptual en este caso, la interpretación geométrica del problema de programación entera es exactamente la misma, sólo cambia la posición de la región factible del espacio lineal afín acotado. Sin embargo, sí hay una diferencia en la traslación algebraica. A saber, no podemos ver los  $a_{ij}, b_i$  negativos directamente como exponentes, ya que no es válido en un polinomio ordinario. Una forma de resolver este problema es considerar en cambio lo que son llamados *polinomios de Laurent* en las variables  $z_i$ , que son expresiones polinomiales en los  $z_i$  y  $z_i^{-1}$ . Para tratar con estos objetos más generales sin introducir un nuevo conjunto de  $m$  variables, usaremos la segunda representación de el anillo de polinomios de Laurent:

$$k[z_1^{\pm 1}, \dots, z_m^{\pm 1}] \cong k[z_1, \dots, z_m, t] / \langle tz_1 \cdots z_m - 1 \rangle.$$

En términos intuitivos, este isomorfismo trabaja introduciendo una nueva variable simple  $t$  que satisface  $tz_1 \cdots z_m - 1 = 0$ , así que formalmente  $t$  es el producto de los inversos de los  $z_i$ :  $t = z_1^{-1} \cdots z_m^{-1}$ . Entonces cada uno de los  $\prod_{i=1}^m z_i^{a_{ij}}$  involucrados en la traslación algebraica del problema de programación entera puede ser reescrito en la forma  $t^{e_j} \prod_{i=1}^m z_i^{a'_{ij}}$ , donde ahora todos los  $a'_{ij} \geq 0$ , podemos tomar  $e_j \geq 0$  como el valor absoluto del más negativo de los  $a_{ij}$  que aparecen, y  $a'_{ij} = a_{ij} + e_j$  para cada  $i$ . Similarmente,  $\prod_{i=1}^m z_i^{b_i}$  puede ser reescrito en la forma  $t^e \prod_{i=1}^m z_i^{b'_i}$  con  $e \geq 0$  y  $b'_i \geq 0$  para todo  $i$ . Así que la ecuación (4.2) se hace una ecuación entre expresiones polinomiales en  $t, z_1, \dots, z_m$ :

$$\prod_{j=1}^n (t^{e_j} \prod_{i=1}^m z_i^{a'_{ij}})^{A_j} = t^e \prod_{i=1}^m z_i^{b'_i}$$

módulo la relación  $tz_1 \cdots z_m - 1 = 0$ . Tenemos una analogía directa con la proposición (4.1.1):

**Proposición 4.1.8** *Defínase un mapeo*

$$\varphi : k[\omega_1, \dots, \omega_n] \rightarrow k[z_1^{\pm 1}, \dots, z_m^{\pm 1}]$$

como sigue

$$\varphi(\omega_j) = t^{e_j} \prod_{i=1}^m z_i^{a'_{ij}} \text{ mod } \langle tz_1 \cdots z_m - 1 \rangle$$

para cada  $j = 1, \dots, n$  y extendiéndose a  $g(\omega_1, \dots, \omega_n) \in k[\omega_1, \dots, \omega_n]$  como antes. Entonces  $(A_1, \dots, A_n)$  es un punto entero en la región factible si y sólo si  $\varphi(\omega_1^{A_1} \omega_2^{A_2} \cdots \omega_n^{A_n})$  y  $t^e z_1^{b'_1} \cdots z_m^{b'_m}$  representan el mismo elemento en  $k[z_1^{\pm 1}, \dots, z_m^{\pm 1}]$  (esto es, su diferencia es divisible por  $tz_1 \cdots z_m - 1$ ).

Similarmente, la proposición (4.1.2) tiene su contraparte en esta situación más general. Escribiremos  $S$  para la imagen de  $\varphi$  en  $k[z_1^{\pm 1}, \dots, z_m^{\pm 1}]$ . Entonces tenemos la siguiente versión para los miembros del subanillo.

**Proposición 4.1.9** *Suponga que  $f_1, \dots, f_n \in k[z_1, \dots, z_m, t]$  están dados. Fije un orden monomial en  $k[z_1, \dots, z_m, t, \omega_1, \dots, \omega_n]$  con la propiedad de eliminación: cualquier monomio que contenga una de las  $z_i$  o  $t$  es mayor que cualquier monomio que contenga sólo los  $\omega_j$ . Finalmente, sea  $G$  una base de Gröbner para el ideal:*

$$J = \langle tz_1 \cdots z_m - 1, f_1 - \omega_1, \dots, f_n - \omega_n \rangle$$



en  $k[z_1, \dots, z_m, t, \omega_1, \dots, \omega_n]$  y para cada  $f \in k[z_1, \dots, z_m, t]$ , sea  $\bar{f}^G$  el residuo en la división de  $f$  por  $G$ . Entonces:

- a.  $f$  representa un elemento en  $S$  si y sólo si  $g = \bar{f}^G \in k[\omega_1, \dots, \omega_n]$ .
- b. Si  $f$  representa un elemento en  $S$  y  $g = \bar{f}^G \in k[\omega_1, \dots, \omega_n]$  como en la parte a, entonces  $f = g(f_1, \dots, f_n)$ , dando una expresión para  $f$  como un polinomio en los  $f_j$ .
- c. Si cada  $f_j$  y son monomios y  $f$  representa un elemento en  $S$ , entonces  $g$  es también un monomio.

Ilustraremos el caso general descrito previamente por medio de un ejemplo. Considere el siguiente problema en forma estándar:

$$\text{Min: } A + 1000B + C + 100D$$

s.a.:

$$\begin{aligned} 3A - 2B + C &= -1 \\ 4A + B - C - D &= 5 \\ A, B, C, D &\in \mathbf{Z}_{\geq 0} \end{aligned} \tag{4.6}$$

Con la relación  $tz_1z_2 - 1 = 0$ , nuestro ideal  $J$  en este caso es:

$$J = \langle tz_1z_2 - 1, z_1^3z_2^4 - \omega_1, t^2z_2^3 - \omega_2, tz_1^2 - \omega_3, tz_1 - \omega_4 \rangle.$$

Si usamos un orden de eliminación poniendo  $t, z_1, z_2$  antes de las variables  $\omega$ , y usando un orden ponderado compatible con  $l$  en las  $\omega_j$  (eliminando empates con *degrevlex*), entonces obtenemos una base de Gröbner  $G$  para  $J$  que consiste de los siguientes polinomios:

$$\begin{aligned} g_1 &= \omega_2\omega_3^2 - \omega_4 \\ g_2 &= \omega_1\omega_4^7 - \omega_3^3 \\ g_3 &= \omega_1\omega_2\omega_4^6 - \omega_3 \\ g_4 &= \omega_1\omega_2^2\omega_3\omega_4^5 - 1 \\ g_5 &= z_2 - \omega_1\omega_2^2\omega_3\omega_4^4 \\ g_6 &= z_1 - \omega_1\omega_2\omega_4^5 \\ g_7 &= t - \omega_2\omega_3\omega_4. \end{aligned}$$

Del lado derecho de las ecuaciones, consideramos  $f = tz_2^6$ . El cálculo del residuo nos da:

$$\bar{f}^G = \omega_1 \omega_2^2 \omega_4.$$

Ya que este es un problema muy pequeño, es fácil checar a mano que la solución correspondiente ( $A = 1, B = 2, C = 0, D = 1$ ) realmente minimiza  $l(A, B, C, D) = A + 1000B + C + 100D$  sujeto a las restricciones dadas.

Ya conocida la forma de trabajar con las bases de Gröbner en problemas de programación entera, es importantísimo recalcar, a modo de observación, que la correcta resolución de problemas enteros, así como la operatividad de los métodos desarrollados, depende en sumo de la estructura del problema. Requiere de gran atención la forma en que se definen variables, restricciones y el tipo de problema mismo.

Así pues, podemos afirmar que el método de aplicar bases de Gröbner a problemas de programación entera, funciona de manera eficaz para distintos tipos de problemas, sólo si estos tienen una estructura bien definida como forma estándar de un problema de programación entera como en (4.1).

### 4.1.3. Ejemplos de problemas generales

Se presentan a continuación un problema del tipo *knapsack* (mochila) o “0-1” y uno del tipo de transporte y asignación, incluyendo la manera en que se plantearon sus respectivas formas estándar y cómo se resolvieron.

$$\begin{aligned} \text{Max: } & 3y_1 + 2y_2 - 5y_3 - 2y_4 + 3y_5 \\ \text{s.a.:} & \\ & y_1 + y_2 + y_3 + 2y_4 + y_5 \leq 4 \\ & 7y_1 + 3y_3 - 4y_4 + 3y_5 \leq 8 \\ & 11y_1 - 6y_2 + 3y_4 - 3y_5 \geq 3 \end{aligned} \tag{4.7}$$

$$y_1, y_2, y_3, y_4, y_5 = (0, 1)$$

Primero que nada, debemos convertir las desigualdades en ecuaciones, esto lo hacemos agregando variables artificiales:

$$\begin{aligned} y_1 + y_2 + y_3 + 2y_4 + y_5 + y_6 & = 4 \\ 7y_1 + 3y_3 - 4y_4 + 3y_5 + y_7 & = 8 \\ 11y_1 - 6y_2 + 3y_4 - 3y_5 - y_8 & = 3 \end{aligned}$$





	Denver (1)	Miami (2)
Los Angeles(1)	\$80	\$215
Detroit(2)	\$100	\$108
Nueva Orleáns(3)	\$102	\$68

La forma estándar del problema es:

$$\text{min: } 80x_{11} + 215x_{12} + 100x_{21} + 108x_{22} + 102x_{31} + 68x_{32}$$

s.a.:

$$\begin{array}{rccccr}
 x_{11} & + & x_{12} & & & = & 1000 \\
 & & & x_{21} & + & x_{22} & = & 1500 \\
 & & & & & & x_{31} & + & x_{32} & = & 1200 \\
 x_{11} & & & + & x_{21} & & + & x_{31} & & = & 2300 \\
 & & x_{12} & & + & x_{22} & & + & x_{32} & = & 1400
 \end{array}$$

Estas restricciones son todas ecuaciones debido a que la oferta total es igual a la demanda total.

Entonces, al aplicar el procedimiento de exponenciación y definir las funciones  $f_i$  tenemos que el ideal es:

$$I = \langle z_1z_4 - x_{11}, z_1z_5 - x_{12}, z_2z_4 - x_{21}, z_2z_5 - x_{22}, z_3z_4 - x_{31}, z_3z_5 - x_{32} \rangle$$

La base de Gröbner consta de 18 generadores y tampoco se muestran. Al hacer la reducción de  $f = z_1^{1000}z_2^{1500}z_3^{1200}z_4^{2300}z_5^{1400}$  por la base de Gröbner para el ideal  $I$  obtenemos:

$$x_{11}^{1000}x_{21}^{1300}x_{22}^{200}x_{32}^{1200}$$

que nos indica que deben enviarse 1000 unidades de Los Angeles a Denver, 1300 de Detroit a Denver, 200 de Detroit a Miami y 1200 de Nueva Orleáns a Miami para minimizar el costo de transportación, el cual, en este caso es de \$313 200.

Como hemos visto, es posible resolver distintos tipos de problemas de programación entera, con la insistencia de que la estructura del problema debe adoptar la forma estándar de manera estricta para poder resolverlo usando bases de Gröbner.

## 4.2. Programación entera y combinatoria

En esta sección estudiaremos una hermosa aplicación del álgebra conmutativa y las ideas desarrolladas en la sección (4.1) en problemas de enumeración combinatoria. Nuestro ejemplo inicial se discute aquí y sus generalizaciones de largo alcance son desarrolladas usando herramientas algebraicas más avanzadas. También hay conexiones entre las técnicas que desarrollaremos aquí, teoría de invariantes, la teoría de variedades tóricas y la geometría de poliedros. Los prerrequisitos para esta sección son la teoría de bases de Gröbner para ideales polinomiales, familiaridad con anillos de cocientes y hechos básicos sobre funciones de Hilbert.

Mucho de esta sección será dedicado a la consideración del siguiente problema clásico de conteo. Recordemos que un cuadro mágico es una matriz entera  $n \times n$  con la propiedad que la suma de las entradas en cada fila y cada columna es la misma. Un cuadro mágico  $4 \times 4$  famoso aparece en el muy conocido grabado *Melancholia* de Albert Dürer:

$$\begin{array}{cccc} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{array}$$

La suma de cada fila y cada columna es 34. Aunque la condición extra que las  $m_{ij}$  sean enteros distintos es a menudo incluido, no lo haremos parte de la definición. Así mismo, muchos ejemplos familiares de cuadros mágicos tienen suma diagonal igual a las sumas de las filas y columnas y otras propiedades interesantes; tampoco lo requerimos. Nuestro problema es este:

**Problema 1** *Dados enteros positivos  $s, n$ , cuántos cuadros mágicos  $n \times n$  diferentes existen con  $m_{ij} \geq 0$  para todo  $i, j$  y con suma  $s$  de filas y columnas.*

Hay cuestiones relativas de estadística y de diseño de experimentos de práctica así como interés puramente matemático. En algunos casos pequeños, la respuesta al problema es fácilmente deducible.

Nuestra meta principal en esta sección será desarrollar un camino general para atacar este y otros problemas similares de conteo donde los objetos a contar pueden ser identificados con los puntos enteros en una región poliédrica en  $\mathbf{R}^N$  para algún  $N$ , así que estamos en la misma situación como en los problemas de programación entera de la sección (4.1). Tomaremos sin embargo, un poco *ad hoc* esta similitud, y la usaremos sólo como maquinaria

muy general que necesitaremos para contestar nuestra pregunta para valores pequeños de  $n$ .

Para ver cómo el problema 1 encaja en este contexto, notese que el conjunto completo de cuadros mágicos  $M$  con  $n \times n$  enteros no negativos es el conjunto de soluciones en  $\mathbf{Z}_{\geq 0}^{n \times n}$  de un sistema de ecuaciones lineales con coeficientes enteros. Por ejemplo en el caso  $3 \times 3$ , las condiciones que todas las sumas de las filas y columnas sean iguales pueden ser expresadas como 5 ecuaciones independientes en las entradas de la matriz. Escribiendo:

$$\vec{m} = (m_{11}, m_{12}, m_{13}, m_{21}, m_{22}, m_{23}, m_{31}, m_{32}, m_{33})^T$$

la matriz  $M = (m_{ij})$  es un cuadrado mágico si y sólo si

$$A_3 \vec{m} = 0 \tag{4.9}$$

donde  $A_3$  es la matriz

$$A_3 = \begin{pmatrix} 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 \\ 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \end{pmatrix} \tag{4.10}$$

y  $m_{ij} \geq 0$  para todo  $i, j$ . Similarmente, los cuadros mágicos  $n \times n$  pueden ser vistos como las soluciones de un sistema similar  $A_n \vec{m} = 0$  para una matriz entera  $A_n$  con  $n^2$  columnas.

Hay tres diferencias importantes entre nuestra situación aquí y los problemas de optimización considerados en la sección (4.1). Primero, no hay funciones lineales a optimizar. En cambio, estamos interesados principalmente en el entendimiento de la estructura del conjunto completo de puntos enteros en la región factible. Segundo, a diferencia de las regiones consideradas en los ejemplos en la sección (4.1), la región factible en este caso es no acotada, y hay infinidad de puntos enteros. Finalmente, tenemos un sistema homogéneo de ecuaciones, así los puntos de interés son elementos del kernel de la matriz  $A_n$ . En lo siguiente, escribiremos

$$K_n = \ker(A_n) \cap \mathbf{Z}_{\geq 0}^{n \times n}$$

para el conjunto de todos los cuadros mágicos  $n \times n$  de enteros no negativos. Comenzaremos con algunas observaciones simples.

**Proposición 4.2.1** Para cada  $n$ :

1.  $K_n$  es cerrado bajo la suma de vectores en  $\mathbf{Z}^{n \times n}$ , y contiene el vector cero.
2. El conjunto  $C_n$  de soluciones de  $A_n \vec{m} = 0$  que satisface  $m \in \mathbf{R}_{\geq 0}^{n \times n}$  forma un cono poliédrico convexo en  $\mathbf{R}^{n \times n}$ , con vértice en el origen.

**Demostración.** La afirmación 1 se sigue por linealidad. Para b,  $C_n$  es poliedro ya que las ecuaciones que lo definen son las ecuaciones lineales  $A_n \vec{m} = 0$  y las desigualdades lineales  $m_{ij} \geq 0 \in \mathbf{R}$ . Es un cono ya que cualquier múltiplo real positivo de un punto en  $C_n$  está también en  $C_n$ . Finalmente, es convexo puesto que si  $\vec{m}$  y  $\vec{m}'$  son dos puntos en  $C_n$ , cualquier combinación lineal  $x = r\vec{m} + (1-r)\vec{m}'$  con  $r \in [0, 1]$  también satisfacen las ecuaciones  $A_n x = 0$  y tienen entradas no negativas, y está en  $C_n$ . ■

Un conjunto  $M$  con una operación binaria se dice ser un *monoide* si la operación es asociativa y posee un elemento identidad en  $M$ . Por ejemplo  $\mathbf{Z}_{\geq 0}^{n \times n}$  es un monoide bajo adición de vectores. En este lenguaje, parte una de las proposiciones que dicen que  $K_n$  es un *submonoide* de  $\mathbf{Z}_{\geq 0}^{n \times n}$ .

Para entender la estructura del submonoide  $K_n$ , buscamos encontrar un conjunto mínimo de generadores aditivos que sirvan como estructura básica para todos los elementos de  $K_n$ . La noción apropiada es dada por la siguiente definición.

**Definición 4.2.2** Sea  $K$  cualquier submonoide del monoide aditivo  $\mathbf{Z}_{\geq 0}^N$ . Un subconjunto finito  $H \subset K$  se dice que es una base de Hilbert para  $K$  si esta satisface las siguientes dos condiciones:

1. Para cada  $k \in K$  existen  $h_i \in H$  y enteros no negativos  $c_i$  tal que  $k = \sum_{i=1}^q c_i h_i$  y
2.  $H$  es mínimo con respecto a inclusión.

Es un hecho general que las bases de Hilbert existen y son únicas para todos los submonoides  $K \subset \mathbf{Z}_{\geq 0}^N$ . En vez de dar una prueba de existencia, presentaremos un algoritmo de bases de Gröbner para encontrar la base de Hilbert para el submonoide  $K = \ker(A)$  en  $\mathbf{Z}_{\geq 0}^N$  para cualquier matriz entera con  $N$  columnas. Como en la sección (4.1), trasladamos nuestro problema de el contexto de puntos enteros a polinomios de Laurent. Dada una matriz entera  $A = (a_{ij})$  con  $N$  columnas y  $m$  filas como se dijo, introducimos un



indeterminado  $z_i$  para cada fila,  $i = 1, \dots, m$ , y consideramos el anillo de polinomios de Laurent:

$$k[z_1^{\pm 1}, \dots, z_m^{\pm 1}] \cong k[z_1, \dots, z_m, t] / \langle tz_1 \cdots z_m - 1 \rangle$$

Definimos un mapeo

$$\psi : k[v_1, \dots, v_N, \omega_1, \dots, \omega_N] \rightarrow k[z_1^{\pm 1}, \dots, z_m^{\pm 1}][\omega_1, \dots, \omega_N] \quad (4.11)$$

como sigue. Primero tomamos

$$\psi(v_j) = \omega_j \cdot \prod_{i=1}^m z_i^{a_{ij}} \quad (4.12)$$

y  $\psi(\omega_j) = \omega_j$  para cada  $j = 1, \dots, N$ , entonces hacemos una extensión a polinomios en  $k[v_1, \dots, v_N, \omega_1, \dots, \omega_N]$  para hacer  $\psi$  un homomorfismo sobre anillos.

**Proposición 4.2.3** *Un vector  $\alpha^T \in \ker(A)$  si y sólo si  $\psi(v^\alpha - \omega^\alpha) = 0$ , esto es, si y sólo si  $v^\alpha - \omega^\alpha$  está en el kernel de el homomorfismo  $\psi$ .*

Podemos escribir  $J = \ker(\psi)$  como:

$$J = I \cap k[v_1, \dots, v_N, \omega_1, \dots, \omega_N],$$

donde

$$I = \langle \omega_j \cdot \prod_{i=1}^m z_i^{a_{ij}} - v_j : j = 1, \dots, N \rangle$$

en el anillo  $k[z_1^{\pm 1}, \dots, z_m^{\pm 1}][v_1, \dots, v_N, \omega_1, \dots, \omega_N]$ . El siguiente teorema de Sturmfels ofrece un camino para encontrar bases de Hilbert.

**Teorema 4.2.4** *Sea  $G$  una base de Gröbner para  $I$  con respecto a algún orden de eliminación  $>$  para el cual todas las  $z_i, t > v_j$ , y todos los  $v_j > \omega_k$ . Sea  $S$  el subconjunto de  $G$  conformado de elementos de la forma  $v^\alpha - \omega^\alpha$  para algún  $\alpha \in \mathbf{Z}_{\geq 0}^N$ . Entonces*

$$H = \{\alpha : v^\alpha - \omega^\alpha \in S\}$$

es la base de Hilbert de  $K$ .

La idea de esta demostración es similar a la del teorema (4.1.5).

Tenemos un ejemplo para ilustrar el teorema (4.2.4). Considere el submonoide de  $\mathbf{Z}_{\geq 0}^4$  dado como  $K = \ker(A) \cap \mathbf{Z}_{\geq 0}^4$ , para

$$A = \begin{pmatrix} 1 & 2 & -1 & 0 \\ 1 & 1 & -1 & -2 \end{pmatrix}$$

Para encontrar una base de Hilbert para  $K$ , consideramos el ideal  $I$  generado por:

$$\omega_1 z_1 z_2 - v_1, \quad \omega_2 z_1^2 z_2 - v_2, \quad \omega_3 t - v_3, \quad \omega_4 z_1^2 t^2 - v_4$$

y  $z_1 z_2 t - 1$ . Calculando una base de Gröbner  $G$  con respecto a un orden de eliminación como en (4.2.4), encontramos sólo una de la forma deseada:

$$v_1 v_3 - \omega_1 \omega_3$$

De aquí que la base de Hilbert para  $K$  consiste de un simple elemento:  $H = \{(1, 0, 1, 0)\}$ . No es difícil verificar de la forma de la matriz  $A$  que todo elemento en  $K$  es un entero múltiplo de este vector. Note que el tamaño de la base de Hilbert no es el mismo que la dimensión del kernel de la matriz  $A$  como un mapeo lineal en  $\mathbf{R}^4$ . En general, no hay conexión entre el tamaño de la base de Hilbert para  $K = \ker(A) \cap \mathbf{Z}_{\geq 0}^N$  y la dimensión de  $\ker(A)$ ; el número de elementos en la base de Hilbert puede ser tanto mayor que, igual a o menor que la dimensión del kernel, dependiendo de  $A$ .

Usaremos el teorema (4.2.4) para continuar nuestro trabajo en el problema de enumeración de cuadros mágicos. Si aplicamos el método del teorema para encontrar la base de Hilbert para el  $\ker(A_3) \cap \mathbf{Z}_{\geq 0}^{3*3}$  (ver la ecuación (4.10) anterior) entonces necesitamos calcular una base de Gröbner para el ideal  $I$  generado por:

$$\begin{array}{ll} v_1 - \omega_1 z_1 z_2 z_4 z_5 & v_2 - \omega_2 z_1^2 z_2^2 z_3^2 z_5 t \\ v_3 - \omega_3 z_1^2 z_2^2 z_3^2 z_4 t & v_4 - \omega_4 z_2 z_4^2 z_5^2 t \\ v_5 - \omega_5 z_2 z_3 z_5 t & v_6 - \omega_6 z_2 z_3 z_4 t \\ v_7 - \omega_7 z_1 z_4^2 z_5^2 t & v_8 - \omega_8 z_1 z_3 z_5 t \\ v_9 - \omega_9 z_1 z_3 z_4 t & \end{array}$$

y  $z_1 \cdots z_5 t - 1$  en el anillo:

$$k[z_1, \dots, z_5, t, v_1, \dots, v_9, \omega_1, \dots, \omega_9].$$

Usando un orden de eliminación como se describió en el teorema (4.2.4) con el sistema computacional *Macaulay 2*, obtenemos una base de Gröbner bastante grande. Sin embargo, si identificamos el subconjunto  $S$  como en el teorema, hay sólo seis polinomios que corresponden a los elementos de la base de Hilbert:

$$\begin{aligned} v_3v_5v_7 - \omega_3\omega_5\omega_7 & & v_3v_4v_8 - \omega_3\omega_4\omega_8 \\ v_2v_6v_7 - \omega_2\omega_6\omega_7 & & v_2v_4v_9 - \omega_2\omega_4\omega_9 \\ v_1v_6v_8 - \omega_1\omega_6\omega_8 & & v_1v_5v_9 - \omega_1\omega_5\omega_9 \end{aligned} \quad (4.13)$$

Expresando los correspondientes seis elementos de la base de Hilbert en forma matricial, vemos algo interesante. Las matrices obtenidas son precisamente las seis matrices  $3 \times 3$  de permutación (simétricas), que son las representaciones matriciales de las permutaciones de los componentes de vectores en  $\mathbf{R}^3$ . Por ejemplo, el elemento de la base de Hilbert  $(0, 0, 1, 0, 1, 0, 1, 0, 0)$  del primer polinomio en (4.13) corresponde a la matriz:

$$T_{13} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

que intercambia  $x_1, x_3$ , dejando fijo  $x_2$ . Similarmente, los otros elementos de la base de Gröbner dada (en el orden de (4.13)) son:

$$\begin{aligned} S &= \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, & S^2 &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \\ T_{12} &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, & T_{23} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, & I &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned}$$

Aquí  $S$  y  $S^2$  son las permutaciones cíclicas,  $T_{ij}$  intercambia  $x_i$  y  $x_j$ ,  $I$  es la identidad.

De hecho, es un bien conocido teorema combinatorio que las matrices  $n \times n$  de permutación forman la base de Hilbert para el monoide  $K_n$  para todo  $n \geq 2$ .

Esto nos da información muy valiosa para trabajar. Por la definición de una base de Hilbert tenemos, por ejemplo, que en el caso  $3 \times 3$  todo elemento  $M$  de  $K_3$  puede ser escrito como una combinación lineal:

$$M = aI + bS + cS^2 + dT_{12} + eT_{13} + fT_{23}$$

donde  $a, b, c, d, e, f$  son enteros no negativos. La suma de filas y columnas de  $M$  esta dada entonces por:

$$s = a + b + c + d + e + f$$

Podra parecer a primera vista que nuestro problema está resuelto para matrices  $3 \times 3$ . A saber, para un valor de la suma dado  $s$ , podra parecer que sólo necesitamos contar las maneras de escribir a  $s$  como una suma de a lo más 6 enteros no negativos  $a, b, c, d, e, f$ . Sin embargo, hay una dificultad adicional que hace el problema aun más interesante: Las 6 matrices de permutación no son linealmente independientes. De hecho hay una relación obvia:

$$I + S + S^2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = T_{12} + T_{13} + T_{23} \quad (4.14)$$

Esto significa que para toda  $s \geq 3$  hay combinaciones diferentes de coeficientes que producen la misma suma matricial. Cómo podemos considerar esto y eliminar conteos múltiples? Primero, esperamos que de hecho cada igualdad

$$aI + bS + cS^2 + dT_{12} + eT_{13} + fT_{23} = a'I + b'S + c'S^2 + d'T_{12} + e'T_{13} + f'T_{23} \quad (4.15)$$

donde  $a, \dots, f, a', \dots, f'$  son enteros no negativos, es una consecuencia de la relación en (4.14), en el sentido que si (4.15) es verdadero, entonces la diferencia de vectores

$$(a, b, c, d, e, f) - (a', b', c', d', e', f')$$

es un múltiplo entero del vector de coeficientes  $(1, 1, 1, -1, -1, -1)$  en la relación lineal dependiente:

$$I + S + S^2 - T_{12} - T_{13} - T_{23} = 0$$

que sigue de (4.14).

Dado esto, podemos resolver nuestro problema en el caso  $3 \times 3$  “retranslándolo” hacia el álgebra. A saber podemos identificar las 6-adas de coeficientes  $(a, b, c, d, e, f) \in \mathbf{Z}_{\geq 0}^6$  con monomios en 6 nuevas variables denotadas por  $x_1, \dots, x_6$ :

$$\alpha = (a, b, c, d, e, f) \leftrightarrow x_1^a x_2^b x_3^c x_4^d x_5^e x_6^f$$

Por (4.14), no obstante, vemos que queremos pensar en  $x_1x_2x_3$  y  $x_4x_5x_6$  como que son lo mismo. Esta observación indica que, en conteo, queremos considerar los elementos del anillo de cocientes:

$$R = k[x_1, \dots, x_6] / \langle x_1x_2x_3 - x_4x_5x_6 \rangle$$

representado por el monomio  $x^\alpha$ . Sea  $MS_3(s)$  el número de cuadros mágicos  $3 \times 3$  distintos con entradas no negativas, y la suma de filas y columnas igual a  $s$ . Nuestro siguiente objetivo es ver que  $MS_3(s)$  puede ser reinterpretado como la función de Hilbert del anterior anillo  $R$ .

Un ideal homogéneo  $I \subset k[x_1, \dots, x_n]$  da un anillo de cocientes  $R = k[x_1, \dots, x_n]/I$ , y la función de Hilbert  $H_R(s)$  es definida por:

$$H_R(s) = \dim_k k[x_1, \dots, x_n]_s / I_s = \dim_k k[x_1, \dots, x_n]_s - \dim_k I_s \quad (4.16)$$

donde  $k[x_1, \dots, x_n]_s$  es el vector espacio de polinomios homogéneos de grado total  $s$ , e  $I_s$  es el vector espacio de polinomios homogéneos de grado total  $s$  en  $I$ . La función de Hilbert de  $R = k[x_1, \dots, x_n]/I$  se denota por  $HF_I(s)$ . Ya que nuestro objeto es el ideal  $I$ , en lo que sigue, llamaremos a  $H_R(s)$  y  $HF_I(s)$  la función de Hilbert de  $I$ . Es un resultado básico que las funciones de Hilbert de  $I$  y  $\langle Tl(I) \rangle$  (para algún orden monomial) son iguales. Ahora podemos calcular la función de Hilbert contando el número de monomios estándar con respecto a  $I$  para cada grado total  $s$ , esto es, monomios de grado total  $s$  en el complemento de  $\langle Tl(I) \rangle$ .

**Proposición 4.2.5** *La función  $MS_3(s)$  iguala la función de Hilbert  $H_R(s) = HF_I(s)$  del ideal homogéneo  $I = \langle x_1x_2x_3 - x_4x_5x_6 \rangle$ .*

**Demostración.** El conjunto simple  $\{x_1x_2x_3 - x_4x_5x_6\}$  es una base de Gröbner para el ideal generado con respecto a algún orden monomial. Fijemos algún orden tal que el término líder del generador es  $x_1x_2x_3$ . Entonces los monomios estándar de grado total  $s$  en  $k[x_1, \dots, x_6]$  son los monomios de grado total  $s$  que no son divisibles por  $x_1x_2x_3$ .

Dado algún monomio  $x^\alpha = x_1^a x_2^b x_3^c x_4^d x_5^e x_6^f$ , sea  $A = \min(a, b, c)$ , y construyamos

$$\alpha' = (a - A, b - A, c - A, d + A, e + A, f + A)$$

Puesto que  $x^{\alpha'}$  no es divisible por  $x_1x_2x_3$ , es un monomio estándar, y es el residuo en la división de  $x^\alpha$  por  $x_1x_2x_3 - x_4x_5x_6$ .

Necesitamos ver que los cuadros mágicos  $3 \times 3$  con suma de filas y columnas  $s$  están en correspondencia uno a uno con los monomios estándar de grado  $s$ . Sea  $M$  un cuadro mágico, y consideremos la expresión:

$$M = aI + bS + cS^2 + dT_{12} + eT_{13} + fT_{23} \quad (4.17)$$

con  $\alpha = (a, \dots, f) \in \mathbf{Z}_{\geq 0}^6$ . Asociamos a  $M$  la forma estándar en  $R$  del monomio  $x^\alpha$ , es decir  $x^{\alpha'}$ . Esto da un mapeo bien definido del conjunto de cuadros mágicos a la colección de monomios estándar con respecto a  $I$ , ya que cualquiera de las dos expresiones (4.17) para  $M$  produce el mismo monomio estándar  $x^{\alpha'}$ . Más aun la suma de filas y columnas de  $M$  es la misma que el grado total de la imagen monomial.

Este mapeo es claramente sobre, ya que el vector de exponentes  $\alpha'$  de cualquier monomio estándar puede ser usado para dar los coeficientes en una expresión (4.17). Es también uno a uno, ya que si  $M$  en (4.17) y

$$M_1 = a_1I + b_1S + c_1S^2 + d_1T_{12} + e_1T_{13} + f_1T_{23}$$

mapea a el mismo monomio estándar  $\alpha'$ , entonces escribiendo  $A = \min(a, b, c)$ ,  $A_1 = \min(a_1, b_1, c_1)$ , tenemos:

$$\begin{aligned} & (a - A, b - A, c - A, d + A, e + Af + A) \\ & = (a_1 - A_1, b_1 - A_1, c_1 - A_1, d_1 + A_1, e_1 + A_1, f_1 + A_1). \end{aligned}$$

De lo que se tiene que  $(a, \dots, f)$  y  $(a_1, \dots, f_1)$  difieren por el vector

$$(A - A_1)(1, 1, 1, -1, -1, -1)$$

Así que los cuadros mágicos  $M$  y  $M_1$  son iguales. ■

Nos gustaría mencionar que también hay un camino muy conceptual para entender la relación entre el monoide  $K_3$  de nuestro problema original y el anillo  $R$  y la correspondiente variedad  $\mathbf{V}(x_1x_2x_3 - x_4x_5x_6)$ , usando la teoría de *variedades tóricas*. En particular, si  $A = \{\vec{m}_1, \dots, \vec{m}_6\} \subset \mathbf{Z}^9$  es el conjunto de vectores enteros correspondientes a las matrices de permutación  $3 \times 3$  anteriores (la base de Hilbert para  $K_3$ ), y definimos  $\phi_A : (\mathbf{C}^*)^9 \rightarrow \mathbf{P}^5$  por:

$$\phi_A(t) = (t^{\vec{m}_1}, \dots, t^{\vec{m}_6})$$

entonces se sigue que la variedad tórica  $X_A$  (la cerradura Zariski de la imagen de  $\phi_A$ ) es la variedad proyectiva  $\mathbf{V}(x_1x_2x_3 - x_4x_5x_6)$ . El ideal  $I_A =$

$\langle x_1x_2x_3 - x_4x_5x_6 \rangle$  es llamado el *ideal tórico* correspondiente a  $A$ . El ideal homogéneo definido de una variedad tórica es generado siempre por diferencias de monomios como en este ejemplo.

Para concluir, la proposición (4.2.5) resuelve el problema de conteo de cuadros mágicos  $3 \times 3$  como sigue. Por la proposición y por (4.16), para encontrar  $MS_3(s)$ , simplemente sustraemos el número de monomios no estándar del grado total  $s$  en 6 variables del número total de monomios de grado total  $s$  en 6 variables. Los monomios no estándar son aquellos divisibles por  $x_1x_2x_3$ ; removiendo tal factor, obtenemos un monomio arbitrario de grado total  $s - 3$ . Así que una expresión es la siguiente:

$$\begin{aligned} MS_3(s) &= \binom{s+5}{5} - \binom{(s-3)+5}{5} \\ &= \binom{s+5}{5} - \binom{s+2}{5} \end{aligned} \quad (4.18)$$

Por ejemplo,  $MS_3(1) = 6$  (los coeficientes binomiales  $\binom{m}{l}$  con  $m < l$  son cero),  $MS_3(2) = 21$  y  $MS_3(3) = 56 - 1 = 55$ .

Después de haber expuesto estas ideas y las aplicaciones de estos algoritmos, nos dirigiremos a mostrar un panorama muy elemental, pero con las características suficientes para poder hacer cálculos y desarrollos, del sistema computacional empleado para experimentar, plantear y resolver los problemas y ejercicios que hemos encontrado a lo largo de este trabajo. Este sistema se llama *Macaulay 2* y lo hemos empleado en una plataforma de Linux, aunque puede hacerse del mismo modo sobre Windows.

# Capítulo 5

## El entorno de *Macaulay 2*

Este capítulo nos permitirá tener una buena noción de los comandos de *Macaulay 2* para algunos cálculos elementales en geometría algebraica considerando que existe familiaridad con las bases de Gröbner que se desarrollaron en el capítulo 3.

### 5.1. Panorama histórico

Los sistemas de ecuaciones polinómicas aparecen en toda la matemática, ciencia e ingeniería. La geometría algebraica provee técnicas teóricas poderosas para estudiar los rasgos cualitativos y cuantitativos de sus conjuntos de soluciones. Los algoritmos recientemente desarrollados han hecho accesibles aspectos teóricos del tema a un amplio rango de matemáticos y científicos. El manejo algorítmico dentro del tema tiene dos objetivos principales: desarrollar las nuevas herramientas para la investigación dentro de la matemática y suministrar nuevas herramientas para modelar y resolver problemas que se presentan en la ciencia e ingeniería. Una buena sinergia surge, como nuevos teoremas, nuevos algoritmos y aplicaciones que aparecen, que llevan a nuevas preguntas teóricas.

Uno de los primeros paquetes de álgebra computacional apuntado a la geometría algebraica era *Macaulay*, el predecesor de *Macaulay 2*, escrito durante los años 1983-1993 por Dave Bayer y Mike Stillman. Ayudó a transformar la noción teórica de una resolución descriptiva en una estimulante nueva herramienta de investigación práctica, y se acostumbró ampliamente para la investigación y enseñanza en el álgebra conmutativa y la geometría



algebraica.

*Macaulay 2* es un sistema de álgebra computacional consagrado al apoyo de la investigación en geometría algebraica, álgebra conmutativa y sus aplicaciones. Está basado en la experiencia ganada de escribir y usar a su predecesor *Macaulay*, pero en este caso, fue escrito por Dan Grayson y Mike Stillman con el apoyo financiero de la Fundación Nacional de Ciencia de E.U.A. (N.S.F. por sus siglas en inglés) que comenzó a trabajar en 1993. También incorpora algunos códigos de otros autores, así como nuevas rutinas y bibliotecas.

*Macaulay 2* se dirige a apoyar el cómputo eficaz asociado con una amplia variedad de objetos matemáticos de alto nivel, incluyendo campos de Galois, campos de números, anillos de polinomios, álgebras exteriores, álgebra de Weyl, anillos de cocientes, ideales, módulos, homomorfismos de anillos y módulos, módulos graduados, mapeo entre módulos graduados, cadenas complejas, mapeo entre cadenas complejas, resoluciones libres, variedades algebraicas y haces coherentes.

Existe una diferencia importante entre los sistemas de álgebra computacional generales (*Maple*, *Mathematica*) y *Macaulay 2*. Antes de insertar una expresión en *Macaulay 2* que involucre variables (como  $x+y$ ), el usuario debe crear primero un anillo que contenga esas variables. Los anillos son objetos importantes de estudio en la geometría algebraica; los anillos de cocientes de anillos polinómicos, por ejemplo, encapsulan la información esencial sobre un sistema de ecuaciones polinómicas, incluyendo, por ejemplo, el campo al que pertenecen los coeficientes.

Daremos a continuación algunas explicaciones e ideas de los tipos de objetos matemáticos disponibles en *Macaulay 2* y presentaremos algunos ejemplos.

## 5.2. Estructuras, comandos y rutinas básicas

Nuestro primer ejemplo concierne a los objetos geométricos en el espacio afín complejo tridimensional. Empezamos dando el anillo de funciones polinomiales con coeficientes racionales:

```
i1 : R=QQ[x,y,z]
```

```
o1 = R
```

```
o1 : PolynomialRing
```

En *Macaulay 2* están disponibles varios órdenes nonomiales; ya que no especificamos uno explícitamente, los monomios en el anillo  $R$  serán tratados con el orden lexicográfico graduado inverso. Definimos un ideal generado por dos polinomios en este anillo y lo asignamos a la variable llamada  $c$ .

```
i2 : c=ideal(x^4-y^5, x^3-y^7)
```

```
o2 = ideal(-y^5+x^4, -y^7+x^3)
```

```
o2 : Ideal of R
```

Calculamos la base de Gröbner reducida de nuestro ideal:

```
i3 : gb c
```

```
o3 = | y5-x4 x4y2-x3 x8-x3y3 |
```

```
o3 : GroebnerBasis
```

La instrucción `gb` indica que queremos la base de Gröbner correspondiente a cierto ideal, en este ejemplo, nuestro ideal es  $c$ .

Presentaremos ahora un ejemplo elaborado para programación entera, el tema de este trabajo, y explicaremos en qué consiste cada comando. Nuestro ejemplo es el sistema lineal definido por la matriz:

```
i1 : A={{1, 1, 1, 1},
        {1, 5,10,25}}
```

```
o1 = {{1, 1, 1, 1},{1, 5, 10, 25}}
```

```
o1 : List
```

Para el estudio algebraico de problemas de programación entera, un buen punto de inicio es trabajar en un anillo de polinomios multigraduado, aquí en cuatro variables:

```
i2 : R=QQ[p,n,d,q, Degrees => transpose A]
```

```
o2 = R
```

```
o2 : PolynomialRing
```

El grado de cada variables es el vector columna correspondiente de la matriz.

Cada variable representa una de las cuatro monedas en E.U.A.. Podemos ver como esta el sistema:

i3 : degree d

o3 = {1, 10}

o3 : List

i4 : degree q

o4 = {1, 25}

o4 : List

Cada monomio representa una colección de monedas. Por ejemplo, suponga que tiene 4 *pennies*, 8 *nickels*, 10 *dimes* y 3 *quarters*:

i5 : degree( $p^4*n^8*d^{10}*q^3$ )

o5 = {25, 219}

o5 : List

Entonces tendría un total de 25 monedas que corresponderían a 2 dólares y 19 centavos. Hay otras nueve formas posibles de tener 25 monedas con la misma suma:

i6 : h=transpose basis({25,219},R)

o6 = {-25, -219} | p11n2d2q7 |  
 {-25, -219} | p9n8d2q6 |  
 {-25, -219} | p9n5d6q5 |  
 {-25, -219} | p9n2d10q4 |  
 {-25, -219} | p4n14d2q5 |  
 {-25, -219} | p4n11d6q4 |  
 {-25, -219} | p4n8d10q3 |  
 {-25, -219} | p4n5d14q2 |  
 {-25, -219} | p4n2d18q |

```

          9      1
o6 : Matrix R <--- R

```

El comando `basis` nos da la base del sistema multigrado respecto a nuestro anillo. El comando `transpose` calcula la transpuesta del objeto que se indique enseguida. Así, la base correspondiente al sistema anterior es un vector fila de nueve elementos. Para sólo contar el número de columnas de esta matriz usamos el comando:

```
i7 : rank source transpose h
```

```
o7 = 9
```

Podemos ver el número de formas de tener 10 dólares usando 100 monedas:

```
i8 : rank source basis({100,1000},R)
```

```
o8 = 182
```

Ahora usaremos el siguiente anillo con el orden lexicográfico y con las variables ordenadas:  $d > p > n > q$ .

```
i9 : s=QQ[x,y,d,p,n,q,
      MonomialOrder => Lex, MonomialSize => 16]
```

```
o9 = s
```

```
o9 : PolynomialRing
```

La opción `MonomialSize` obliga a *Macaulay 2* a usar más espacio para almacenar los exponentes de los monomios para evitar estar fuera del rango de almacenamiento.

Definimos un ideal con un generador para cada columna de la matriz A.

```
i10 : i=ideal(p-x*y, n-x*y^5, d-x*y^10, q-x*y^25)
```

```
o10 = ideal (- x*y + p, - x*y5 + n, - x*y10 + d, - x*y25 + q)
```

```
o10 : Ideal of s
```

El programa entero es resuelto por reducción en forma normal con respecto a la siguiente base de Gröbner conformada por binomios.

```
i11 : trasnpose gens gb i
```

```
o11 = {-6} | p5q-n6      |
      {-4} | d4-n3q     |
      {-3} | yn2-dp     |
      {-6} | yp4q-dn4   |
      {-4} | yd3-pnq    |
      {-6} | y2p3q-d2n2 |
      {-5} | y2d2n-p2q  |
      {-7} | y2d2p3-n5  |
      {-6} | y3p2q-d3   |
      {-6} | y3dp2-n3   |
      {-5} | y4p-n      |
      {-6} | y5n-d      |
      {-8} | y6d2-pq    |
      {-16}| y15d-q     |
      {-7} | xq-y5d2    |
      {-5} | xn-y3p2    |
      {-2} | xd-n2      |
      {-2} | xy-p       |
```

```

              18      1
o11 : Matrix s  <--- s
```

Fijamos el anillo de cocientes, de modo que la reducción a la forma normal será automática.

```
i12 : s'=s/i
```

```
o12 = s'
```

```
o12 : QuotientRing
```

Vemos que necesitamos al menos dos *dimes* para expresar un dólar con 10 monedas.

```
i13 : x^10*y^100
```

```

      2 6 2
o13 = d n q

```

```
o13 : s'
```

Como sabemos, el programa es infactible si y sólo si la forma normal contuviera la variable  $x$  o la variable  $y$ . Por ejemplo, no podemos expresar 10 dólares con menos de 40 monedas:

```
i14 : x^39*y^1000
```

```

      25 39
o14 = y q

```

```
o14 : s'
```

Ahora introducimos un nuevo término de orden en el anillo de polinomios, definido por asignación de un peso a cada variable. Específicamente, asignamos pesos para cada una de las monedas. Por ejemplo:

```
i15 : weight=(5,7,13,17)
```

```
o15 = (5, 7, 13, 17)
```

```
o15 : Sequence
```

Damos un nuevo anillo con el orden ponderado resultante y trabajaremos módulo el mismo ideal de antes.

```

i16 : t=QQ[x,y,p,n,d,q,
      Weights => {{1,1,0,0,0,0},{0,0,weight}},
      MonomialSize => 16]/
      (p-x*y, n-x*y^5, d-x*y^10, q-x*y^25)

```

Y ahora vemos como expresar por ejemplo un dólar con diez monedas.

```
i17 : x^10*y^100
```

```

      5 2 3
o17 = p d q

```

```
o17 : t
```

Tenemos una solución óptima que involucra los cuatro tipos de monedas:

```
i18 : x^234*y^5677
```

```
      2 4 3 225
o18 = p n d q
```

```
o18 : t
```

Finalmente, presentaremos la forma en que planteamos nuestro problema entero (4.3), así como los detalles encontrados en *Macaulay 2* para resolverlo.

```
i1 : S=QQ[x,y,a,b,c,d,MonomialOrder=>Lex]
```

```
o1 = S
```

```
o1 : PolynomialRing
```

```
i2 : I=ideal(a-x^5*y^3,b-x^4*y^2,c-x,d-y)
```

```
      5 3      4 2
o2 = ideal (- x y + a, - x y + b, - x + c, - y + d)
```

```
o2 : Ideal of S
```

```
i3 : transpose gens gb I
```

```
o3 = {-1} | y-d      |
      {-3} | bcd-a   |
      {-4} | a2c2-b3 |
      {-5} | ac3d-b2 |
      {-5} | b4d-a3c |
      {-6} | c4d2-b  |
      {-1} | x-c     |
```

```
      7      1
o3 : Matrix S <--- S
```

```
i4 : S'=S/I
```

```
o4 = S'
```

```
o4 : QuotientRing
```

```
i5 : x^37*y^20
```

```
      37 20
o5 = c d
```

```
o5 : S'
```

El hecho de que hayamos tenido este resultado es debido a que usamos un orden que no tenía la propiedad de Compatibilidad con la función objetivo  $l$  como un orden adaptado. Así, mostramos la resolución de este problema que, aunque sencilla, nos permitió ver el poder computacional de los algoritmos presentados en este trabajo. Hemos de hacer esta presentación, mostrando una sesión de *Macaulay 2* desde el inicio e indicando que se usó un orden ponderado que resultó del desarrollo del lema (4.1.7) el cual si tiene la propiedad de compatibilidad con  $l$ .

```
Macaulay 2, version 0.9.2
```

```
--Copyright 1993-2001, D. R. Grayson and M. E. Stillman
```

```
--Singular-Factory 1.3b, copyright 1993-2001, G.-M. Greuel,  
et al.
```

```
--Singular-Libfac 0.3.2, copyright 1996-2001, M. Messollen
```

```
i1 : S=QQ[x,y,a,b,c,d, MonomialOrder => Lex, Weights =>  
      {{1,1,0,0,0,0},{0,0,1,1,2,2}}, MonomialSize => 16]
```

```
o1 = S
```

```
o1 : PolynomialRing
```

```
i2 : I=ideal(a-x^5*y^3,b-x^4*y^2,c-x,d-y)
```

```
      5 3      4 2
o2 = ideal (- x y + a, - x y + b, - x + c, - y + d)
```



o2 : Ideal of S

i3 : transpose gens gb I

```
o3 = {-1} | y-d      |
      {-3} | bcd-a   |
      {-4} | a2c2-b3 |
      {-5} | ac3d-b2 |
      {-5} | b4d-a3c |
      {-6} | c4d2-b  |
      {-1} | x-c     |
```

```
o3 : Matrix S 7 <--- S 1
```

i4 : S'=S/I

o4 = S'

o4 : QuotientRing

i5 : x<sup>37</sup>\*y<sup>20</sup>

```
o5 = a4 b4 c
```

o5 : S'

# Conclusiones

- El método de las bases de Gröbner siempre da un óptimo (si existe) de un problema de programación entera, si éste tiene una estructura bien definida de forma estándar de programación lineal.
- El uso de órdenes ponderados para obtener las bases de Gröbner y realizar las reducciones de los monomios generados por el proceso de exponenciación, permite que la solución de un problema de programación entera sea compatible con la función objetivo.
- Es posible, con algunos conocimientos más profundos del sistema, crear rutinas más sencillas para el programa *Macaulay 2* a fin de agilizar los cálculos.
- En ninguno de los casos que se resolvieron se presentó alguna dificultad en el proceso de cálculo, no se encontró un límite de datos de entrada para el programa *Macaulay 2*, debido a que los problemas planteados en los ejemplos son relativamente pequeños. Sin embargo, es necesario profundizar esta información a fin de hacer un aporte más significativo a la ciencia.

# Bibliografía

- [1] P. Conti & C. Traverso: *Buchberger algorithm and integer programming in Applied Algebra*, Algorithms and Error-Correcting Codes AAECC'9 (H.F. Mattson, T. Mora, T.R.N. Rao editions). Lecture Notes in Computational Science, vol. 539, Springer Verlag, Berlin & New York, 1991.
- [2] David Cox, John Little and Donal O'Shea: *Ideals, varieties and algorithms*. Springer-Verlag, New York, second edition. 1997. An introduction to computational algebraic geometry and commutative algebra.
- [3] David Cox, John Little and Donal O'Shea: *Using algebraic geometry*. Springer-Verlag, New York, 1998.
- [4] David Eisenbud: *Commutative algebra with a view toward algebraic geometry*. Springer-Verlag, New York, 1995.
- [5] Bernd Sturmfels: *Gröbner bases and convex polytopes*, volume 8 of *University Lecture Series*. American Mathematical Society, Providence, RI, 1996.
- [6] W.J.Cook, W.H.Cunningham, W.R.Pulleyblank and A.Schrijver: *Combinatorial optimization*. Wiley, New York, 1998.
- [7] G.L. Nemhauser and L.A. Wolsey: *Integer and combinatorial optimization*. Wiley, New York, 1998.
- [8] A. Schrijver: *Theory of linear and integer programming*. Wiley, Chichester, 1986.
- [9] L.A. Wolsey: *Integer programming*. Wiley, New York, 1998.
- [10] W. Cook, A. Gerards, A. Schrijver and É. Tardos: *Sensistivity theorems in integer linear programming*, Mathematical Programming 34, 1986.

- [11] G. Dantzig, R. Fulkerson and S. Johnson: *Operations research 2*. 1954.
- [12] A.M. Geoffrion: *Lagrangean relaxation for integer programming*, Mathematical Programming Study 2. 1974.
- [13] R.E. Gomory: *An algorithm for integer solutions of linear programs*, Recent Advances in Mathematical Programming (R.L.Graves, P. Wolfe, eds.), McGraw-Hill, New York, 1963.
- [14] Hamdy A. Taha: *Investigación de operaciones: una introducción*. Prentice Hall, México, 1998.
- [15] Wayne L. Winston: *Investigación de operaciones*. Prentice Hall, México, 1994.
- [16] Gary D. Eppen y F.G. Gould: *Investigación de operaciones en la ciencia administrativa*. Prentice Hall, México, 1987.
- [17] Thomas L. Saaty: *Optimization in integer and related extremal problems*. McGraw-Hill, New York, 1970.
- [18] Mokhtar S. Bazaraa, et. al.: *Programación lineal y flujo en redes*. Limusa-Noriega, México, 1998.
- [19] Frederick S. Hillier, et. al.: *Introducción a la investigación de operaciones*. McGraw-Hill, México, 1997.
- [20] David G. Luenberger: *Programación lineal y no lineal*. Addison-Wiley, México, 1989.
- [21] Juan Prawda Witemberg: *Métodos y modelos de investigación de operaciones*. Limusa, México, 1976.
- [22] Leandro Pardo Llorente, et. al.: *Programación lineal entera: aplicaciones prácticas en la empresa*. Madrid, 1990.
- [23] Hamdy A. Taha: *Integer programming: theory, applications and computations*. Academic Press, New York, 1975.