

---

---

**LA COLUMNA DE MATEMÁTICA COMPUTACIONAL**

Sección a cargo de

**Tomás Recio**

---

---

**Álgebra computacional y programación entera no lineal**

por

**Jesús Gago Vargas y María Isabel Hartillo Hermoso****INTRODUCCIÓN**

Ante todo, querido lector, paciencia. Vamos a realizar un viaje que nos llevará desde el Álgebra hasta la Investigación Operativa, desde los anillos de polinomios hasta el cálculo de la mejor solución de un problema cuyas variables son enteras. Comencemos por diferenciar un problema continuo de un problema discreto. Supongamos que queremos calcular el valor mínimo de la función lineal  $f(x_1, x_2) = x_1 + x_2$  cuando  $0 \leq x_1 \leq 4$ ,  $0 \leq x_2 \leq 5$  y  $10x_1 + 6x_2 \geq 15$ .

La función  $f$  se denomina función objetivo o coste. Si las variables  $x_1$  y  $x_2$  pueden ser reales, la solución óptima es  $z_c = (3/2, 0)$ , como aparece en la figura 1, con coste  $f(z_c) = 3/2$ . Este es un problema continuo. Si las variables solamente pueden ser enteras, es un problema discreto, y una solución óptima es  $z_d = (2, 0)$ , con coste  $f(z_d) = 2$ . Hay otra solución discreta, que es el punto  $(1, 1)$ . En el caso continuo, la *región factible* está formada por el cierre convexo de los puntos de  $\mathbb{R}^2$  que cumplen las restricciones. En la figura 1 aparece como sombreada. En el caso discreto, por los puntos con coordenadas enteras que se encuentran en la zona anterior. Un problema continuo puede tener una región factible no vacía, y el mismo problema discreto no. Los problemas que vamos a tratar son discretos, donde la función de coste es lineal y el conjunto de las restricciones contendrá algunas lineales y otras más complicadas.

La primera idea a explicar es la relación entre los puntos discretos de la región factible y los exponentes de ciertos polinomios. Por ello, comenzamos con lo básico.

**1. ANILLOS DE POLINOMIOS**

Una buena introducción a lo que sigue en la sección se puede encontrar, por ejemplo, en [5, cap. 1 y 2], con abundantes ejemplos. En lo que sigue,  $\mathbb{K}$  denota un

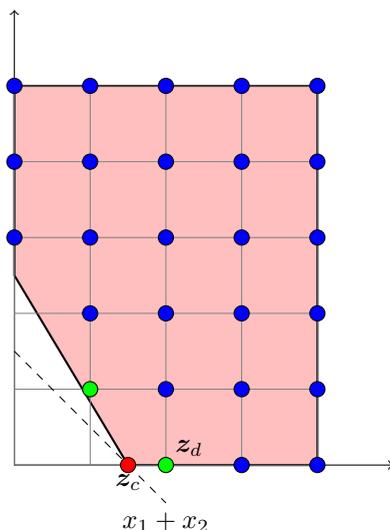


Figura 1: Región factible del problema continuo y discreto.

cuerpo y  $\mathbb{K}[x_1, \dots, x_n]$  el anillo de polinomios en  $n$  indeterminadas con coeficientes en  $\mathbb{K}$ . Los monomios del anillo se notarán como  $\mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$  para  $\alpha \in \mathbb{Z}_+^n$ . El vector  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \in \mathbb{Z}_+^n$  se denomina *vector de exponentes*. El grado de  $\mathbf{x}^\alpha$  es  $\deg(\mathbf{x}^\alpha) = \sum_{i=1}^n \alpha_i$ . El grado de un polinomio  $f = \sum_{\alpha \in \mathbb{Z}_+^n} f_\alpha \mathbf{x}^\alpha$ , que escribiremos como  $\deg(f)$ , es el máximo grado de  $\mathbf{x}^\alpha$  para  $f_\alpha \neq 0$ .

DEFINICIÓN 1.1. Decimos que un conjunto de polinomios  $I \subset \mathbb{K}[x_1, \dots, x_n]$  forma un *ideal* si satisface las siguientes propiedades:

1.  $0 \in I$ .
2. Si  $f, g \in I$ , entonces  $f + g \in I$ .
3. Si  $f \in I$  y  $g \in \mathbb{K}[x_1, \dots, x_n]$ , entonces  $gf \in I$ .

Si  $F \subset \mathbb{K}[x_1, \dots, x_n]$ , entonces el conjunto

$$\langle F \rangle = \left\{ f = \sum_{i=1}^k h_i f_i \mid f_i \in F, h_i \in \mathbb{K}[x_1, \dots, x_n], i = 1, \dots, k \right\}$$

es el ideal generado por  $F$  en  $\mathbb{K}[x_1, \dots, x_n]$ .

EJEMPLO 1.2. Consideremos en  $\mathbb{R}[x_1, x_2]$  los polinomios  $f_1 = x_1^2 + x_2^2 - 1$ ,  $f_2 = x_1 - x_2$  y  $f_3 = x_1^2$ . Entonces

$$f_1 + (x_1 + x_2)f_2 - 2f_3 = (x_1^2 + x_2^2 - 1) + x_1^2 - x_2^2 - 2x_1^2 = -1.$$

Esto significa que  $-1 \in \langle f_1, f_2, f_3 \rangle$ , por lo que este ideal es todo el anillo de polinomios.

1.1. ORDEN MONOMIAL Y DIVISIÓN

En primer lugar establecemos una ordenación de los términos de la forma  $\mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$  en  $n$  indeterminadas.

DEFINICIÓN 1.3. Un *orden monomial*, o de términos,  $\succ$  en  $\mathbb{K}[x_1, \dots, x_n]$  es una relación de orden entre monomios que verifica las siguientes condiciones:

1.  $\succ$  es un orden total: para cualquier par de monomios  $\mathbf{x}^\alpha$  y  $\mathbf{x}^\beta$  se tiene que  $\mathbf{x}^\alpha \succ \mathbf{x}^\beta$  o bien  $\mathbf{x}^\beta \succ \mathbf{x}^\alpha$ . Si ambas se dan, entonces  $\alpha = \beta$ .
2.  $\succ$  es compatible con el producto: si  $\mathbf{x}^\alpha \succ \mathbf{x}^\beta$ , entonces  $\mathbf{x}^\gamma \mathbf{x}^\alpha \succ \mathbf{x}^\gamma \mathbf{x}^\beta$ .
3.  $\succ$  es un buen orden: toda familia de monomios en  $\mathbb{K}[x_1, \dots, x_n]$  tiene un elemento mínimo con respecto a  $\succ$ .

Lo anterior es equivalente a decir que toda sucesión estrictamente decreciente de monomios termina:

$$\mathbf{x}^{\alpha_1} \succ \mathbf{x}^{\alpha_2} \succ \cdots \succ \mathbf{x}^{\alpha_k} = \mathbf{x}^{\alpha_{k+1}}.$$

EJEMPLO 1.4. 1. Orden lexicográfico. Es la ordenación de un diccionario. Decimos que  $\mathbf{x}^\alpha \succ_{\text{lex}} \mathbf{x}^\beta$  si la primera entrada no nula de  $\alpha - \beta$  es positiva. Por ejemplo,  $x_1^2 \succ_{\text{lex}} x_2^{100}$ , porque  $(2, 0) - (0, 100) = (2, -100)$  y la primera entrada no nula es positiva.

2. Orden graduado lexicográfico. Decimos que  $\mathbf{x}^\alpha \succ_{\text{deglex}} \mathbf{x}^\beta$  si  $\sum_{i=1}^n \alpha_i > \sum_{i=1}^n \beta_i$ , o, en caso de igualdad  $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i$ , el empate se decide por el orden lexicográfico  $\mathbf{x}^\alpha \succ_{\text{lex}} \mathbf{x}^\beta$ . Así,  $x_2^{100} \succ_{\text{deglex}} x_1^2$ , porque el grado del primer monomio es mayor.

3. Orden graduado inverso lexicográfico. Decimos que  $\mathbf{x}^\alpha \succ_{\text{degrevlex}} \mathbf{x}^\beta$  si  $\sum_{i=1}^n \alpha_i > \sum_{i=1}^n \beta_i$ , o, en caso de igualdad  $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i$ , la última entrada no nula de  $\alpha - \beta$  es negativa. Con este orden,  $x_1^4 x_2^3 x_3^2 \succ_{\text{degrevlex}} x_1^2 x_2^3 x_3^4$ .

En lugar de  $\mathbf{x}^\alpha \succ \mathbf{x}^\beta$  escribiremos  $\alpha \succ \beta$ , ya que podemos ver  $\succ$  como un orden en  $\mathbb{Z}_+^n$ , el espacio de los vectores exponente. Más adelante definiremos otro orden monomial que nos será de gran utilidad en optimización.

Usaremos la notación  $\text{LT}_\succ(g)$  para referirnos al término líder de  $g$  respecto al orden monomial  $\succ$ . Cuando no haya lugar a confusión, quitaremos la referencia al orden y escribiremos simplemente  $\text{LT}(g)$ .

DEFINICIÓN 1.5. Un ideal  $I \subset \mathbb{K}[x_1, \dots, x_n]$  es un ideal *monomial* si existe un subconjunto  $A$  (no necesariamente finito) de monomios tales que  $I = \langle \mathbf{x}^\alpha \mid \alpha \in A \rangle$ .

Una característica interesante de un ideal monomial es que es finitamente generado, es decir, existe un subconjunto finito  $B \subset A$  tal que  $I = \langle \mathbf{x}^\alpha \mid \alpha \in A \rangle = \langle \mathbf{x}^\alpha \mid \alpha \in B \rangle$ . No es más que una versión algebraica del lema de Gordan-Dickson, que dice que un conjunto infinito  $S \subset \mathbb{Z}_+^n$  tiene un número finito de puntos minimales con respecto a la relación  $\leq$  componente a componente (véase [5, cap. 2, secc. 4]).

DEFINICIÓN 1.6. Dado un ideal  $I \subset \mathbb{K}[x_1, \dots, x_n]$ , notaremos por  $\text{LT}_\succ(I)$  al conjunto de los monomios líder de los polinomios en el ideal  $I$ , esto es,

$$\text{LT}_\succ(I) = \{\text{LT}_\succ(f) \mid f \in I\}.$$

## 1.2. BASES DE GRÖBNER

Lo que sigue se puede encontrar con más detalle en [5, cap. 2]. Fijamos un orden monomial  $\succ$  y buscamos un conjunto especial asociado al ideal  $I$ . En primer lugar, observemos que  $\langle LT(I) \rangle$  es un ideal monomial. Por el lema de Gordan-Dickson, existe un conjunto finito de polinomios  $g_1, g_2, \dots, g_s \in I$  tales que  $\langle LT(I) \rangle = \langle LT(g_1), LT(g_2), \dots, LT(g_s) \rangle$ . A un conjunto finito de polinomios  $G = \{g_1, g_2, \dots, g_s\}$  que verifique esta condición lo llamamos *base de Gröbner* de  $I$  respecto al orden  $\succ$ .

Se tiene que una base de Gröbner genera el ideal de partida. Además, existe un algoritmo (Buchberger) que permite calcular una base de Gröbner de un ideal  $I$  con respecto a un orden monomial  $\succ$ , si partimos de un conjunto finito de generadores de  $I$ , tal como se prueba en [5, cap. 2, secc. 7]. Una base de Gröbner no es única. Para fijar una especial, se puede construir una base de Gröbner *reducida*  $G_0$ , que es una base de Gröbner del ideal  $I$  tal que cada polinomio  $p \in G_0$  verifica que el coeficiente del término líder es igual a 1 y ningún monomio de  $p$  está en  $LT(G_0 - \{p\})$ .

## 2. PROGRAMACIÓN ENTERA LINEAL

La aplicación de las bases de Gröbner a la programación entera lineal se basa en el comportamiento de los exponentes de un cierto ideal asociado al problema. Una aproximación a este problema con ejemplos se puede ver en [4, cap. 8], y otra un poco más avanzada en [14].

Consideremos la familia de problemas

$$(IP)(\mathbf{b}) : \min\{\mathbf{c}^t \mathbf{z} \mid A\mathbf{z} = \mathbf{b}, \mathbf{z} \in \mathbb{Z}_+^n\},$$

donde  $A \in \mathbb{Z}^{m \times n}$  es una matriz fija de enteros,  $\mathbf{c} \neq \mathbf{0}$  es un vector de coste fijo y el término derecho  $\mathbf{b} \in \mathbb{Z}^m$  puede variar. El conjunto de puntos factibles está formado por los correspondientes al retículo del poliedro

$$P(\mathbf{b}) = \text{conv}\{\mathbf{z} \mid A\mathbf{z} = \mathbf{b}, \mathbf{z} \in \mathbb{Z}_+^n\}.$$

Vamos a suponer que el problema lineal (continuo)

$$(LP)(\mathbf{b}) : \min\{\mathbf{c}^t \mathbf{z} \mid A\mathbf{z} = \mathbf{b}, \mathbf{z} \geq \mathbf{0}\}$$

está acotado para todo  $\mathbf{b}$ . Es una hipótesis natural para garantizar la existencia de solución. Esto implica que el problema lineal entero  $(IP)(\mathbf{b})$  está también acotado, y es posible transformar el problema original a uno donde la función de coste y la matriz de restricciones  $A$  tienen coeficientes no negativos, como se explica en [7].

La idea fundamental que conecta los polinomios con la programación entera lineal es la definición de un orden adecuado. Sea  $\mathbf{c}$  un vector de coste con componentes no negativas y elegimos un orden monomial  $\succ$  cualquiera. Definimos un orden monomial compatible con  $\mathbf{c}$  como

$$\mathbf{y} \succ_{\mathbf{c}} \mathbf{z} \Leftrightarrow \begin{cases} \mathbf{c}^t \mathbf{y} > \mathbf{c}^t \mathbf{z}, & \text{o bien} \\ \mathbf{c}^t \mathbf{y} = \mathbf{c}^t \mathbf{z} & \text{e } \mathbf{y} \succ \mathbf{z}, \end{cases} \quad \text{para } \mathbf{y}, \mathbf{z} \in \mathbb{Z}_+^n.$$

En otras palabras, comparamos los exponentes según el coste  $\mathbf{c}$  y, en caso de empate, usamos un orden fijado de antemano. Entonces los problemas

$$(\text{IP})_{\succ_{\mathbf{c}}}(\mathbf{b}) : \min_{\succ_{\mathbf{c}}} \{z \mid Az = \mathbf{b}, z \in \mathbb{Z}_+^n\}$$

tienen una única solución óptima o no hay puntos factibles. Recordemos que el problema  $(\text{LP})(\mathbf{b})$  puede tener varios puntos factibles de coste mínimo. El orden  $\succ_{\mathbf{c}}$  determina cuál es el menor mediante el desempate.

Los polinomios se obtienen a partir de la matriz  $A$ . Notaremos por  $I_A$  al ideal de  $\mathbb{K}[x_1, \dots, x_n]$  generado por

$$I_A := \langle \mathbf{x}^u - \mathbf{x}^v \mid Au = Av; \mathbf{u}, \mathbf{v} \in \mathbb{Z}_+^n \rangle.$$

No es inmediato calcular un conjunto generador finito de  $I_A$ . Este ideal está muy relacionado con el retículo  $\ker(A) \cap \mathbb{Z}^n$ , pues  $\mathbf{u} - \mathbf{v}$  es un vector de componentes enteras que pertenece a  $\ker(A)$ . Podemos restringir nuestra atención a binomios  $\mathbf{x}^u - \mathbf{x}^v$  con soporte disjunto. Si  $\text{mcd}(\mathbf{x}^u, \mathbf{x}^v) = \mathbf{x}^\gamma$ , entonces el generador  $\mathbf{x}^u - \mathbf{x}^v \in I_A$  se puede cambiar por  $\mathbf{x}^{u-\gamma} - \mathbf{x}^{v-\gamma}$ , que también está en  $I_A$ . Esta observación nos permite codificar los binomios  $\mathbf{x}^u - \mathbf{x}^v$  con soporte disjunto como el vector  $\mathbf{u} - \mathbf{v}$ . Si en un vector  $\mathbf{w} \in \mathbb{Z}^n$  separamos las componentes positivas y negativas, podemos escribir  $\mathbf{w} = \mathbf{w}_+ - \mathbf{w}_-$ , donde  $\mathbf{w}_+, \mathbf{w}_- \in \mathbb{Z}_+^n$ . Por ejemplo,

$$\mathbf{w} = (2, 0, -1, -3) = (2, 0, 0, 0) - (0, 0, 1, 3) = \mathbf{w}_+ - \mathbf{w}_-.$$

Entonces cada vector  $\mathbf{w} \in \ker(A) \cap \mathbb{Z}^n$  se aplica de forma unívoca en el binomio  $\mathbf{x}^{\mathbf{w}_+} - \mathbf{x}^{\mathbf{w}_-}$ , que es de soporte disjunto.

Sabemos que el ideal  $I_A$  tiene un conjunto finito de generadores y el algoritmo de cálculo de una base de Gröbner de  $I_A$  tiene la buena propiedad de producir únicamente binomios de la forma  $\mathbf{x}^{\mathbf{u}_+} - \mathbf{x}^{\mathbf{u}_-}$ .

**TEOREMA 2.1** ([3]). *Sea  $\succ$  un orden monomial fijado,  $A_{m \times n}$  una matriz entera dada y  $\mathbf{c} \in \mathbb{R}_+^n$  un vector de coste. Sea  $G_{\succ_{\mathbf{c}}}$  la base de Gröbner reducida minimal de  $I_A$  con respecto a  $\succ_{\mathbf{c}}$ . Entonces, para cualquier vector  $\mathbf{b}$  y cualquier solución factible no óptima  $\mathbf{z}_0$ , existe un binomio  $\mathbf{x}^u - \mathbf{x}^v$  tal que  $\mathbf{z}_0 - \mathbf{u} + \mathbf{v}$  es una solución factible mejor, respecto a  $\succ_{\mathbf{c}}$ , que  $\mathbf{z}_0$ .*

**DEMOSTRACIÓN.** Sea  $\mathbf{z}^*$  la única solución óptima del problema  $(\text{IP})_{\succ}(\mathbf{b})$ . Entonces  $A\mathbf{z}_0 = A\mathbf{z}^*$  y  $\text{LT}(\mathbf{x}^{\mathbf{z}_0} - \mathbf{x}^{\mathbf{z}^*}) = \mathbf{x}^{\mathbf{z}_0}$ . Por la definición de base de Gröbner, existe algún elemento  $\mathbf{x}^u - \mathbf{x}^v \in G_{\succ_{\mathbf{c}}}$  tal que  $\mathbf{x}^u$  divide a  $\mathbf{x}^{\mathbf{z}_0}$ . Sea

$$\mathbf{x}^{\mathbf{z}_1} = \mathbf{x}^{\mathbf{z}_0} - \mathbf{x}^{\mathbf{z}_0 - \mathbf{u}}(\mathbf{x}^u - \mathbf{x}^v) = \mathbf{x}^{\mathbf{z}_0 - \mathbf{u} + \mathbf{v}}.$$

Por construcción,  $\mathbf{z}_0 \succ_{\mathbf{c}} \mathbf{z}_1$  y  $\mathbf{z}_1 \geq \mathbf{0}$ . Además,

$$A\mathbf{z}_1 = A(\mathbf{z}_0 - \mathbf{u} + \mathbf{v}) = A\mathbf{z}_0 + A(-\mathbf{u} + \mathbf{v}) = \mathbf{b} + \mathbf{0} = \mathbf{b}.$$

Tenemos así el resultado. □

El resultado anterior proporciona un método para calcular el óptimo a partir de un punto factible  $z_0$ . Si  $z_0 = z^*$  es el óptimo, no podremos reducirlo con ningún vector que proceda de los exponentes de  $G_{>c}$ . Si no es el óptimo, podremos encontrar un punto  $z_1$  mejor. Este proceso es finito, con lo que tendremos el óptimo.

Pero tenemos dos grandes problemas: no conocemos un conjunto generador finito de  $I_A$  y necesitamos un punto factible desde el que aplicar el procedimiento anterior. Sin entrar en detalles, es posible crear un problema entero ampliado, de forma similar a como se hace en la fase I del método del simplex, que nos permite determinar si el problema es factible y, en tal caso, calcular un punto de la región. Además, se puede obtener fácilmente un conjunto finito de generadores del problema ampliado y deducir, mediante una técnica de eliminación, un conjunto finito de generadores del ideal  $I_A$ .

Esto cierra el círculo: todo problema de programación lineal entera puede ser resuelto mediante el cálculo de la base de Gröbner de un cierto ideal. La siguiente pregunta es qué métodos tenemos para ese cálculo.

### 3. CONJUNTO TEST

El proceso que hemos visto antes obtiene mejores puntos desde un punto factible mediante un conjunto de vectores, representados por los exponentes de la base de Gröbner asociada. Este conjunto recibe el nombre de *conjunto test* ([14, p. 171]). En este caso, es finito, no depende del punto de partida y es independiente del vector  $b$ . Existen otros problemas de optimización donde es posible calcular un conjunto con unas propiedades similares a las del conjunto test. Por ejemplo, consideremos

$$\min\{f(z) \mid Az = b, l \leq z \leq u, z \in \mathbb{Z}^n\},$$

donde  $A$  es una matriz entera de orden  $m \times n$ ,  $b \in \mathbb{Z}^m$ ,  $l, u \in \mathbb{Z}^n$  y  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  es una función convexa de la forma  $f(z) = \sum_{j=1}^n f_j(z_j)$ , con  $f_j : \mathbb{R} \rightarrow \mathbb{R}$  funciones convexas,  $j = 1, \dots, n$ . Entonces existe, y es posible calcular, un conjunto finito  $\mathcal{G}$ , que se denomina base de Graver, tal que para un punto factible  $z_0$  no óptimo, existe  $g \in \mathcal{G}$  y un salto  $\alpha \in \mathbb{Z}_+$  tal que  $f(z_0 - \alpha g) < f(z_0)$  (véase [11]).

El resultado del teorema 2.1 nos dice que el conjunto de exponentes de la base de Gröbner reducida del ideal  $I_A$  es un conjunto test de los problemas (IP)( $b$ ).

El cálculo de dicho conjunto test no es sencillo. Un programa muy eficiente para ello es `4ti2` ([1]), y se basa en lo que se conoce como algoritmo geométrico de Buchberger ([14, secc. 4]), que trabaja con vectores en lugar de polinomios, más ciertos criterios que eliminan cálculos en la construcción de la base. Tenemos así una conexión entre un desarrollo teórico y su aplicación: una mejora algebraica permite una mejora en el cálculo del óptimo de un problema.

### 4. MEJORES MÉTODOS

No obstante, tras esta descripción algebraica de la resolución de un problema de optimización, es natural que surja la siguiente pregunta: ¿es competitivo este método? La respuesta es no. Basta leer, por ejemplo, en [10], el extraordinario desarrollo

que ha tenido la programación entera en los últimos años. Y no solamente en teoría, sino que herramientas como CPLEX ([6]), BONMIN ([2]) o BARON ([12]), entre otros, incorporan muy buenos algoritmos para resolver problemas del tipo (IP)( $\mathbf{b}$ ). Dicho de otra forma: desde el punto de vista práctico, el cálculo del conjunto test para un problema lineal entero no es el método más eficiente para resolverlo. Las técnicas de *branch and bound* han demostrado ser muy efectivas para tratar problemas con un buen número de variables y ecuaciones (vea, por ejemplo, [10, p. 625]).

Entonces, ¿para qué sirve todo lo anterior? En primer lugar, para dar cotas de complejidad de algunos problemas, como en [11]. En segundo lugar, una forma diferente de considerar un conjunto test nos permite abordar ciertos problemas no lineales.

## 5. PROGRAMACIÓN ENTERA NO LINEAL

El conjunto test  $\mathcal{G}$  del problema (IP)( $\mathbf{b}$ ) nos lleva desde un punto factible  $\mathbf{z}_0$  hasta el óptimo  $\mathbf{z}^*$  mediante movimientos de la forma  $\mathbf{z}_0 - \mathbf{g}$ , para  $\mathbf{g} \in \mathcal{G}$ . Esto significa que desde el óptimo  $\mathbf{z}^*$  se puede alcanzar *cualquier* punto factible de la región  $A\mathbf{x} = \mathbf{b}$  con los vectores del conjunto  $\mathcal{G}$ , pero con movimientos de la forma  $\mathbf{z}_0 + \mathbf{g}$ , que proporciona un punto  $\mathbf{z}_1$  mayor con respecto a  $\succ_{\mathbf{c}}$ .

Consideremos entonces un problema de la forma

$$(\text{NLIP}) : \min\{\mathbf{c}^t \mathbf{z} \mid A\mathbf{z} = \mathbf{b}, Q_1(\mathbf{z}) \leq 0, \dots, Q_r(\mathbf{z}) \leq 0, \mathbf{z} \in \mathbb{Z}_+^n\},$$

donde  $\mathbf{c} \in \mathbb{Z}_+^n$  es un vector de coste,  $A_{m \times n}$  es una matriz entera, y  $Q_1, \dots, Q_r : \mathbb{R}^n \rightarrow \mathbb{R}$  son funciones *arbitrarias*. Debemos insistir en el carácter general de las funciones  $Q_i$ . Pueden ser lineales que no queramos incorporar en la matriz  $A$ , definidas a trozos, asociadas a una distribución de probabilidad o lo que se nos ocurra. La única condición es que se puedan calcular para cada  $\mathbf{z} \in \mathbb{Z}_+^n$ .

La idea es usar el problema lineal relajado

$$(\text{LIP}) : \min\{\mathbf{c}^t \mathbf{z} \mid A\mathbf{z} = \mathbf{b}, \mathbf{z} \in \mathbb{Z}_+^n\},$$

bajo la hipótesis de que sea acotado. Supongamos que tenemos disponible el conjunto test  $\mathcal{G}$  de este problema y un punto factible. Podemos entonces obtener el óptimo  $\mathbf{z}^*$  mediante el proceso de reducción. Si  $\mathbf{z}^*$  verifica que  $Q_i(\mathbf{z}^*) \leq 0$ ,  $i = 1, \dots, r$ , entonces es el óptimo buscado. En otro caso, todo punto factible del problema lineal relajado (LIP) puede ser alcanzado. En particular, podemos llegar a cualquier punto factible del problema (NLIP), que es un subconjunto de los puntos factibles de (LIP).

La idea central es que no es necesaria una enumeración exhaustiva de los puntos factibles de (NLIP). Si  $\mathbf{p}'$  es un punto factible de (NLIP), con coste  $\mathbf{c}^t \mathbf{p}'$ , todos los caminos que partan desde puntos cuyo coste es mayor que  $\mathbf{c}^t \mathbf{p}'$  se pueden podar. En la figura 2 se corresponden con los puntos marcados como «Poda». Recordemos que un movimiento de la forma  $\mathbf{p}' + \mathbf{g}$ , con  $\mathbf{g} \in \mathcal{G}$ , proporciona un punto que no mejora el valor de  $\mathbf{c}^t \mathbf{p}'$ . Cuando un camino se poda, todos los puntos que están en él se descartan. Observemos que si  $\mathbf{p}' + \mathbf{g}$  tiene alguna componente negativa, no es

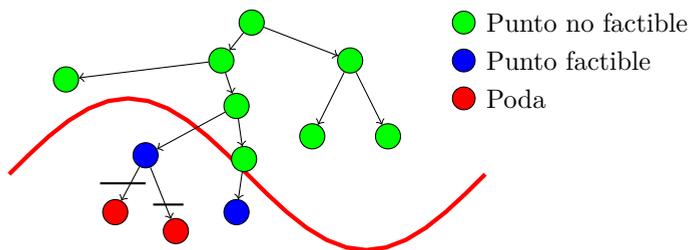


Figura 2: Proceso de poda.

factible y se descarta. Si el punto  $\mathbf{p}'$  no es factible para (NLIP), porque para alguna restricción se tiene que  $Q_i(\mathbf{p}') > 0$ , lo almacenamos y debemos estudiar lo que ocurre con sus descendientes, esto es, los puntos de la forma  $\mathbf{p}' + \mathbf{g}$ ,  $\mathbf{g} \in \mathcal{G}$ . En la figura 2 son los puntos marcados como «Punto no factible».

Al final obtendremos un óptimo del problema (NLIP) o un certificado de que el problema tiene una región factible vacía. Lo anterior es la idea desarrollada en [13] y que vamos a detallar con un ejemplo.

## 6. UN EJEMPLO

Vamos a tratar de minimizar el coste en un sistema serie-paralelo con elección múltiple (véase la figura 3), con una restricción de fiabilidad mínima. El problema se puede modelar de la siguiente forma:

$$\begin{aligned}
 (\text{NLIP}) : c(\mathbf{x}) &= \sum_{i=1}^n \sum_{j=1}^{k_i} c_{ij} x_{ij}, \\
 \sum_{j=1}^{k_i} x_{ij} &\geq 1, \quad i = 1, \dots, n, \\
 R(\mathbf{x}) &= \prod_{i=1}^n \left[ 1 - \prod_{j=1}^{k_i} (1 - r_{ij})^{x_{ij}} \right] \geq R_0, \\
 0 \leq x_{ij} &\leq u_{ij}, \quad x_{ij} \in \mathbb{Z}_+; \quad i = 1, \dots, n; \quad j = 1, \dots, k_i,
 \end{aligned}$$

donde

- $r_{ij} \in (0, 1)$  es la fiabilidad del dispositivo a colocar en la componente  $j$  en el subsistema  $i$ .
- $x_{ij}$  es el número de dispositivos que se colocan en la componente  $j$  en el subsistema  $i$ ; notaremos

$$\mathbf{x} = (x_{11}, \dots, x_{1k_1}, \dots, x_{n1}, \dots, x_{nk_n}).$$

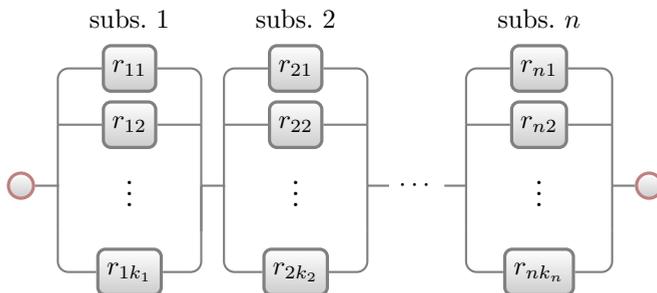


Figura 3: Sistema serie-paralelo con elección múltiple.

- $u_{ij}$  es la cota superior de dispositivos que se pueden colocar en la componente  $j$  en el subsistema  $i$ .
- $c_{ij}$  es el coste del dispositivo a colocar en la componente  $j$  en el subsistema  $i$ .
- $1 - \prod_{j=1}^{k_i} (1 - r_{ij})^{x_{ij}}$  es la fiabilidad del subsistema  $i$ .
- $R(\mathbf{x})$  es la fiabilidad del sistema completo.
- $R_0$  es el nivel mínimo de fiabilidad.

El problema es discreto, pues se trata de determinar cuántos dispositivos tenemos que colocar en cada componente, al menor coste y con una fiabilidad mínima dada por  $R_0$ . Es no lineal, por la restricción definida por la función  $R(\mathbf{x})$ . El problema lineal relajado, que resulta tras eliminar la restricción no lineal, se traduce en

$$\begin{aligned}
 \text{(LIP)} : \quad & \sum_{i=1}^n \sum_{j=1}^{k_i} c_{ij} x_{ij}, \\
 & \sum_{j=1}^{k_i} x_{ij} - d_i = 1, \quad i = 1, \dots, n, \\
 & x_{ij} + t_{ij} = u_{ij}; \quad i = 1, \dots, n; \quad j = 1, \dots, k_i, \\
 & x_{ij}, d_i, t_{ij} \in \mathbb{Z}_+ \text{ para todo } i, j.
 \end{aligned}$$

Las desigualdades hay que convertirlas en igualdades, para lo que usamos variables de holgura  $d_i, t_{ij}$ . Es posible obtener una fórmula del conjunto test asociado a (LIP) sin necesidad de aplicar el algoritmo de Buchberger. En concreto, el conjunto

$$G = \{x_{ij}d_i - t_{ij}, x_{iq}t_{ip} - x_{ip}t_{iq}\}$$

para  $i = 1, \dots, n, j = 1, \dots, k_i, 1 \leq q < p \leq k_i$  es una base de Gröbner reducida asociada al ideal  $I_A$  de (LIP) (véase [8]), con respecto al orden  $\succ_c$  dado por los costes, ordenados como  $c_{iq} \geq c_{ip}$  para  $q < p$ , y tomando  $\mathbf{x} > \mathbf{d} > \mathbf{t}$  y cada bloque

con orden lexicográfico

$$\begin{aligned}x_{11} &> \cdots > x_{1k_1} > x_{21} > \cdots > x_{nk_n}, \\d_1 &> \cdots > d_n, \\t_{11} &> \cdots > t_{1k_1} > t_{21} > \cdots > t_{nk_n}.\end{aligned}$$

Los términos líder aparecen subrayados.

Consideramos un caso pequeño con  $n = 2$  subsistemas, cada uno de ellos con  $k_1 = k_2 = 3$  componentes, y dispositivos con los siguientes valores:

$$\begin{aligned}r_{11} = 0.95, \quad r_{12} = 0.92, \quad r_{13} = 0.90, \quad r_{21} = 0.94, \quad r_{22} = 0.90, \quad r_{23} = 0.85, \\c_{11} = 10, \quad c_{12} = 8, \quad c_{13} = 6, \quad c_{21} = 9, \quad c_{22} = 7, \quad c_{23} = 5.\end{aligned}$$

Exigimos una fiabilidad mínima  $R_0 = 0.98$  y cotas superiores iguales a 3. El problema (NLIP) tiene solución, pues para  $x_{ij} = 3$  la fiabilidad del sistema es 0.9999. Consideramos el problema lineal relajado

$$\text{mín} \sum_{i,j} c_{ij}x_{ij}, \quad \sum_j x_{ij} \geq 1; \quad i = 1, 2; \quad 0 \leq x_{ij} \leq 3.$$

Para el cálculo de este óptimo no hace falta ningún algoritmo: asigna las componentes más baratas en cada subsistema, es decir,

$$x_{11} = 0, \quad x_{12} = 0, \quad x_{13} = 1, \quad x_{21} = 0, \quad x_{22} = 0, \quad x_{23} = 1.$$

Este punto no es factible para el problema (NLIP), pues  $R(\mathbf{x}) = 0.765$ . El óptimo lineal tiene la forma

$$\mathbf{z}_0 = (\underbrace{0, 0, 1, 0, 0, 1}_{x_{ij}}, \underbrace{0, 0}_{d_i}, \underbrace{3, 3, 2, 3, 3, 2}_{t_{ij}}),$$

aunque escribiremos, para abreviar, las seis primeras componentes en lo que sigue. El conjunto test se compone de doce vectores, que notaremos por  $\mathbf{g}_k$ . En este caso, tenemos

$$\begin{aligned}\mathbf{g}_1 &= (1, 0, 0, 0, 0, 0), \quad \mathbf{g}_2 = (0, 1, 0, 0, 0, 0), \quad \mathbf{g}_3 = (0, 0, 1, 0, 0, 0), \\ \mathbf{g}_4 &= (0, 0, 0, 1, 0, 0), \quad \mathbf{g}_5 = (0, 0, 0, 0, 1, 0), \quad \mathbf{g}_6 = (0, 0, 0, 0, 0, 1), \\ \mathbf{g}_7 &= (1, -1, 0, 0, 0, 0), \quad \mathbf{g}_8 = (1, 0, -1, 0, 0, 0), \quad \mathbf{g}_9 = (0, 1, -1, 0, 0, 0), \\ \mathbf{g}_{10} &= (0, 0, 0, 1, -1, 0), \quad \mathbf{g}_{11} = (0, 0, 0, 1, 0, -1), \quad \mathbf{g}_{12} = (0, 0, 0, 0, 1, -1).\end{aligned}$$

Calculamos los puntos  $\mathbf{z}_0 + \mathbf{g}_k$ ,  $k = 1, \dots, 12$ , y la fiabilidad de cada uno de ellos. Así, por ejemplo,

$$\begin{aligned}\mathbf{p}_1 = \mathbf{z}_0 + \mathbf{g}_1 &= (1, 0, 1, 0, 0, 1), \quad R(\mathbf{p}_1) = 0.8457, \quad \mathbf{c}(\mathbf{p}_1) = 21, \\ \mathbf{p}_2 = \mathbf{z}_0 + \mathbf{g}_2 &= (0, 1, 1, 0, 0, 1), \quad R(\mathbf{p}_2) = 0.8432, \quad \mathbf{c}(\mathbf{p}_2) = 19.\end{aligned}$$

Todos estos puntos, al no ser factibles para el problema (NLIP), deben ser procesados de nuevo. Si los guardamos ordenados por su coste, necesitamos 86 iteraciones para llegar a un punto factible, que es el óptimo  $\mathbf{z}^* = (1, 0, 1, 0, 0, 3)$ , con coste igual a 31.

Este método tiene dos problemas:

1. El cálculo del conjunto test, asociado a una base de Gröbner, del problema (LIP) puede ser costoso.
2. En el procedimiento se crea una lista de puntos, y para cada uno de ellos hay que analizar sus descendientes. Una idea inicial es ordenar esta lista según el coste de la función objetivo. En el momento que aparezca un punto factible de (NLIP), todos los puntos de la lista de coste mayor o igual se pueden borrar. No obstante, el número de registros a procesar hasta alcanzar un punto factible suele ser grande.

Con respecto al primero, hay problemas en los que es posible dar una expresión del conjunto test, como en [8, 9]. En otros, cuando no es posible, las mejoras que incorpora `4ti2` han permitido tratar problemas con un número medio de variables y restricciones.

En cuanto al segundo, es interesante considerar una expresión que combine el coste de un punto y su distancia a la región factible del problema (NLIP). En general, un punto  $\mathbf{p}'$  factible para (LIP) pero no para (NLIP) tendrá un coste bajo y estará lejos de la región factible del problema a tratar. En optimización es habitual usar lo que se conoce como una función de penalización, que establece un equilibrio entre ambos valores. Por ejemplo, si fijamos un valor constante  $\mu$ , definimos  $T(\mathbf{x}) = \mathbf{c}^t \mathbf{x} + \mu p(\mathbf{x})$ , donde

$$p(\mathbf{x}) = \sum_{j=1}^r \max(Q_j(\mathbf{x}), 0).$$

El valor de  $\mu$  admite diversas estimaciones o se afina mediante experimentos. La idea es mantener ordenada la lista de puntos pendientes de tratar en forma ascendente según el valor de  $T$ . El resultado de los problemas tratados en [9] es un cálculo muy rápido de un óptimo.

En el ejemplo anterior, consideramos la función de penalización  $T(\mathbf{x}) = \mathbf{c}^t \mathbf{x} + \mu p(\mathbf{x})$ , donde

$$p(\mathbf{x}) = \max(R_0 - R(\mathbf{x}), 0), \quad \mu = 100.$$

En 51 iteraciones se alcanza un punto factible y en 64 el óptimo  $\mathbf{z}^*$ . El proceso de certificación de que, efectivamente, es un óptimo del problema, revisa unos 120 registros. Este comportamiento se repite cuando incrementamos los valores de  $n$ ,  $k_i$  y  $u_i$ , tal como se muestra en [9] para valores mayores. El uso de una función de penalización junto al conjunto test permite calcular muy rápidamente un punto factible del problema (NLIP), incluso el óptimo, de forma competitiva con respecto a otros resolutores, tal como vemos en la comparativa de la figura 4.

Recordemos que este método también certifica que un punto es el óptimo. Los casos estudiados en [8] y [9] generan un proceso de certificación largo, pues el número de puntos a tratar es bastante elevado.

## 7. CONCLUSIONES

Como resumen de todo lo anterior, el método propuesto resuelve problemas discretos en donde pueden aparecer restricciones no lineales dadas por funciones *arbi-*

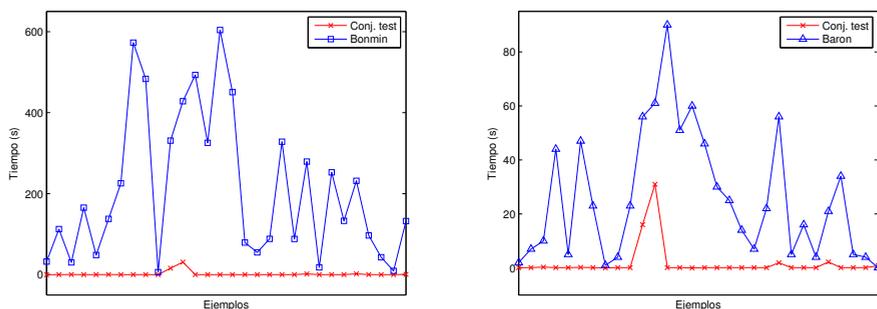


Figura 4: Tiempo para encontrar un óptimo: conjunto test, Bonmin y Baron.

*trarias*, y proporciona en múltiples ocasiones mejores resultados que los resolutores más competitivos. La única condición sobre dichas funciones es que puedan ser evaluadas en cada punto. El fundamento es la construcción de un conjunto test para un problema lineal relajado, que se basa en la aplicación de un algoritmo cuyos orígenes encontramos en el Álgebra Computacional. En algunos casos, los vectores del conjunto test se pueden calcular previamente, mediante la teoría de bases de Gröbner.

## REFERENCIAS

- [1] 4TI2 TEAM, 4ti2-a software package for algebraic, geometric and combinatorial problems on linear spaces, <http://www.4ti2.de>
- [2] P. BONAMI AND CONTRIBUTORS, Bonmin: basic open-source nonlinear mixed integer programming, <https://projects.coin-or.org/Bonmin>
- [3] P. CONTI Y C. TRAVERSO, Buchberger algorithm and integer programming, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, Lecture Notes in Comput. Sci., vol. 539, 130–139, Springer, Berlin, 1991.
- [4] D. A. COX, J. LITTLE Y D. O'SHEA, *Using Algebraic Geometry*, 2nd. ed., Graduate Texts in Mathematics, vol. 185, Springer, New York, 2005.
- [5] D. A. COX, J. LITTLE Y D. O'SHEA, *Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 4th. ed., Undergraduate Texts in Mathematics, Springer, Cham, 2015.
- [6] CPLEX OPTIMIZER, High-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming, <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html>
- [7] J. A. DE LOERA, R. HEMMECKE Y M. KÖPPE, *Algebraic and Geometric Ideas in the Theory of Discrete Optimization*, MOS-SIAM Series on Optimization, vol. 14, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2013.

- [8] J. GAGO-VARGAS, I. HARTILLO, J. PUERTO Y J. M. UCHA, Exact cost minimization of a series-parallel reliable system with multiple component choices using an algebraic method, *Comput. Oper. Res.* **40** (2013), no. 11, 2752–2759.
- [9] J. GAGO-VARGAS, I. HARTILLO, J. PUERTO Y J. M. UCHA, An improved test set approach to nonlinear integer problems with applications to engineering design, *Comput. Optim. Appl.* **62** (2015), no. 2, 565–588.
- [10] M. JÜNGER, T. LIEBLING, D. NADDEF, G. NEMHAUSER, W. PULLEYBLANK, G. REINELT, G. RINALDI Y L. WOLSEY (EDS.), *50 years of integer programming 1958–2008*, Springer, Berlin, 2010.
- [11] S. ONN, *Nonlinear discrete optimization: an algorithmic theory*, Zurich Lectures in Advanced Mathematics, European Mathematical Society (EMS), Zürich, 2010.
- [12] N. V. SAHINIDIS, BARON 14.3.1: Global Optimization of Mixed-Integer Nonlinear Programs, *User's Manual*, <http://www.minlp.com/downloads/docs/baronmanual.pdf>
- [13] S. R. TAYUR, R. R. THOMAS Y N. R. NATRAJ, An algebraic-geometry algorithm for scheduling in presence of setups and correlated demands, *Math. Programming* **69** (1995), no. 3, 369–401.
- [14] G. M. ZIEGLER, Gröbner bases and integer programming, *Some Tapas of Computer Algebra*, Algorithms Comput. Math., vol. 4, 168–183, Springer, Berlin, 1999.

JESÚS GAGO VARGAS, DPTO. DE ÁLGEBRA, UNIVERSIDAD DE SEVILLA  
Correo electrónico: [gago@us.es](mailto:gago@us.es)

MARÍA ISABEL HARTILLO HERMOSO, DPTO. DE MATEMÁTICA APLICADA I, UNIVERSIDAD DE SEVILLA  
Correo electrónico: [hartillo@us.es](mailto:hartillo@us.es)