

## ALGORITMO DEL ELIPSOIDE INTERIOR PARA PROGRAMACIÓN LINEAL

ÁNGEL SALAMANCA FERNÁNDEZ  
JESÚS JUAN RUIZ

Universidad Politécnica de Madrid

*En este artículo se desarrolla un algoritmo de puntos interiores para programación lineal a partir de consideraciones geométricas. En cada iteración del método se dispone de un punto interior al politopo. Con centro en dicho punto se obtiene un elipsoide interior a dicho politopo. La optimización de la función objetivo lineal sobre el elipsoide se obtiene mediante la solución de un problema de mínimos cuadrados. El punto resultante se adopta para la siguiente iteración. Se proponen dos métodos diferentes para resolver el problema de mínimos cuadrados con el fin de reducir el número de operaciones requeridas en cada iteración. Finalmente, se ha desarrollado una aplicación informática para los métodos anteriores en FORTRAN 77 para un IBM PC AT, con el fin de comprobar empíricamente el comportamiento del algoritmo en comparación con el método Simplex.*

**An inner ellipsoid algorithm for linear programming.**

**Keywords:** Programación lineal, mínimos cuadrados, algoritmo de Karmarkar, método del Simplex.

Clasificación AMS: 90C06, 68C05.

---

—Ángel Salamanca Fernández, Jesús Juan Ruiz. Laboratorio de Estadística. E.T.S. de Ingenieros Industriales. Universidad Politécnica de Madrid. Subvencionado en parte por la Dirección General de Política Científica y Técnica, MEC, Proyecto PB87-0808.

—Article rebut el febrer de 1989.

## 1. INTRODUCCIÓN

La programación lineal aparece con frecuencia en distintos e importantes problemas de decisión como, por ejemplo, la composición óptima de mezclas o la programación de actividades en el tiempo. El desarrollo de método del Simplex por Dantzig (1951) permitió obtener la solución algorítmica de estos problemas y ha sido el único procedimiento empleado desde entonces, sin que ninguno de los métodos sustitutos propuestos en las últimas décadas haya sido capaz de igualarlo. Con la publicación del método proyectivo de Karmarkar (1984) el interés en la programación lineal ha renacido y es posible que entre los nuevos algoritmos de puntos interiores propuestos surja una buena alternativa al método clásico (ver Gill *et al.*, 1986).

En este artículo se construye y analiza un algoritmo de puntos interiores equivalente al de Vanderbei *et al.*, (1986) a partir de consideraciones puramente geométricas. La deducción del algoritmo es original y uno de sus atractivos radica fundamentalmente en su contenido geométrico. El algoritmo, a diferencia del método de Karmarkar, trabaja con el problema en su formulación estándar —como el Simplex— y no necesita ningún conocimiento a priori del óptimo de la función objetivo.

Dado un punto interior al polítopo definido por las restricciones del problema lineal, se define un elipsoide centrado en él y estrictamente incluido en la región admisible. La optimización sobre el elipsoide de la función objetivo proporciona un nuevo punto interior más próximo a la solución óptima. De esta manera, el método genera una sucesión infinita de puntos interiores que converge a la solución del problema.

El proceso requiere resolver un problema de mínimos cuadrados en cada iteración. La resolución directa mediante la descomposición triangular de Cholesky o métodos estándares similares convierten el algoritmo de optimización en lento y sumamente ineficiente. En este trabajo se indican dos procedimientos especiales que resuelven en parte este inconveniente y agilizan la convergencia del método.

## 2. FÓRMULA DEL ELIPSOIDE INTERIOR

Los métodos de puntos interiores son bien conocidos en programación no lineal con restricciones, especialmente los métodos de penalización y los métodos de barrera (Fiacco y McCormick, 1968). Hasta muy recientemente, la aplicación de tales algoritmos se limitó a la programación no lineal.

La diferencia esencial entre las condiciones de optimalidad derivadas del problema general de programación no lineal sin restricciones y las que corresponden al caso no lineal restringido (Kuhn-Tucker) proporcionan una generalización directa de los procedimientos para problemas sin restricciones a aquellos que sí las tienen. Por ejemplo, avanzar en la dirección del gradiente en la maximización de una función, no es un método razonable cuando se está en presencia de restricciones de desigualdad, debido a que el gradiente no tiene porqué anularse en el óptimo. Este hecho es especialmente obvio en el caso de programación lineal, cuando el gradiente es constante. Las adaptaciones de los métodos sin restricciones a la programación lineal provocan la pérdida de la interioridad de los puntos. Éste es el caso del método del gradiente reducido, analizado por Kallio y Orchard-Hays (1981). Sin embargo, el avance en la dirección de la máxima pendiente se puede realizar si se tiene cuidado en adaptar en cada iteración la métrica a los límites de la región admisible. La naturaleza de las restricciones en programación lineal hace que esto sea posible.

## 2.1 Elipsoide interior para la formulación canónica

Se considera el caso sencillo de  $n$  restricciones de desigualdad en  $\mathbb{R}^n$ , junto con una forma lineal cuyo máximo condicionado es el único vértice del politopo, esto es

$$(1) \quad \begin{array}{l} \text{Maximizar } c^T x \\ \text{restringido a } A^T x \leq b \end{array}$$

donde  $A$  es una matriz cuadrada de rango máximo y  $b, c$  son vectores constantes de  $\mathbb{R}^n$ . Sea  $P$  el politopo  $n$  dimensional de  $\mathbb{R}^n$  definido por las desigualdades anteriores. Si se dispone de un punto interior  $a \in \mathbb{R}^n$ , es posible inscribir un elipsoide que esté centrado en  $a$  y sea tangente a cualquier hiperplano que delimita  $P$ .

Dado que las siguientes matrices son definidas positivas:

$$(2) \quad \begin{array}{l} D = \text{diag}(d_1, d_2, \dots, d_n) \\ Q = AD^{-2}A^T, \end{array}$$

donde  $d_i = b_i - A_i^T a > 0$ , se puede definir el elipsoide  $E$  mediante la siguiente ecuación,

$$(3) \quad (x - a)^T Q (x - a) = 1.$$

Evidentemente, el centro de  $E$  es  $a$ . El máximo de la función objetivo  $c^T x$  sobre  $E$  se alcanza en la dirección  $d$ , que se obtiene resolviendo el siguiente sistema de ecuaciones lineales,

$$(4) \quad Qd = c,$$

la longitud de avance en esa dirección está determinada por el escalar

$$\alpha = (c^T d)^{-1/2}$$

y el nuevo punto es

$$(5) \quad \bar{a} = a + \alpha d$$

Es interesante comprobar que el incremento en la función objetivo dado por el proceso de optimización descrito es inverso al escalar  $\alpha$ .

$$(6) \quad \bar{z} - z = c^T (\bar{a} - a) = \alpha c^T d = \frac{1}{\alpha}$$

De esta manera se puede construir una sucesión infinita de puntos interiores que convergen al óptimo  $x^*$ . El método es fácil de generalizar para aquellos casos en los que intervengan más hiperplanos. Dado el programa lineal siguiente:

$$(7) \quad \begin{aligned} &\text{maximizar } c^T x \\ &\text{restringido a } A^T x \leq b, \\ &A \in \mathbb{R}^{(n,m)} \text{ con } n \leq m \\ &b \in \mathbb{R}^m \text{ y } x, c \in \mathbb{R}^n \end{aligned}$$

Siendo  $A$  de rango máximo, sea  $P$  el politopo definido por  $A^T x \leq b$  y  $a$  un punto interior a él. Es posible generalizar la ecuación (3) para obtener un elipsoide centrado en  $a$ . El elipsoide así calculado no es, en general, tangente a las diferentes caras de  $P$ , pero su interioridad está garantizada.

**Lema 1.**

Si  $a$  es punto interior a  $P(A^T x \leq b)$  y  $Q$  la matriz definida en (2), entonces el elipsoide definido por la ecuación

$$(8) \quad (x - a)^T Q(x - a) = 1$$

está contenido en  $P$ .

### **Demostración:**

Es equivalente a decir que la distancia desde cualquier hiperplano  $P_j$ ,  $A_j^T x = b_j$ , donde  $A_j$  es una columna de  $A$ , en la métrica definida por  $Q$  es mayor o igual a uno. Teniendo en cuenta que la distancia al cuadrado entre dos puntos  $x$ ,  $y$  según la métrica definida por  $Q$  viene dada por

$$d_Q(x, y)^2 = (x - y)^T Q(x - y),$$

y llamando  $d_Q(a, P_j)$  a la distancia entre el punto  $a$  y su proyección ortogonal según  $Q$  sobre  $P_j$ , se puede comprobar que

$$d_Q(a, P_j)^2 = \frac{(b_j - A_j^T a)^2}{A_j^T Q^{-1} A_j}.$$

Si consideramos la matriz  $B = AD^{-1}$  y el siguiente problema de mínimos cuadrados  $B^T u = e_j$ , los residuos asociados a la solución del problema son:

$$r^2 = \|e_j - B^T (BB^T)^{-1} B e_j\|^2 = 1 - B_j^T Q^{-1} B_j \geq 0$$

$$d_Q(a, P_j)^2 = \frac{1}{B_j^T Q^{-1} B_j} \geq 1.$$

■

Para cualquier punto interior a  $P$ , el lema 1 proporciona un elipsoide interior. La maximización de la forma lineal sobre dicho elipsoide es exactamente igual que en el caso de  $n = m$ . Es elemental comprobar que la relación (6) también se cumple en el caso general.

## **2.2 Elipsoide interior en la formulación estándar**

La fórmula del elipsoide interior es la base que va a servir para definir un nuevo algoritmo de programación lineal. Antes de analizar éste, se va a deducir la ecuación del elipsoide para los programas lineales definidos en forma estándar:

$$(9) \quad \begin{array}{l} \text{minimizar } c^T x \\ \text{restringido a } Ax = b, x \geq 0 \end{array}$$

Para cualquier punto  $a$  interior al politopo  $P$ , se considera el siguiente elipsoide:

$$\begin{aligned} a &= (a_1, a_2, \dots, a_n) \\ D &= \text{diag}(a_1, a_2, \dots, a_n) \\ E' &= \{x \in \mathbb{R}^n : (x - a)^T D^{-2} (x - a) = 1\}. \end{aligned}$$

Los puntos interiores a  $E'$  cumplen

$$\sum_{j=1}^n \frac{(x_j - a_j)^2}{a_j^2} < 1,$$

como todos los términos del sumatorio son positivos, cada uno de ellos debe verificar que  $(x_j - a_j)^2 < a_j^2$  y por consiguiente  $x_j > 0$ , por lo que cumplen las restricciones de no negatividad. Veamos que su intersección con el nuevo politopo  $P = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$  es un elipsoide  $E$ , equivalente al dado por la ecuación cuya demostración se ha realizado. Considerando la partición habitual de  $A$  en programación lineal:

$$A = [B, N]$$

donde  $B$  es una matriz cuadrada de rango máximo y  $N$  es rectangular y la misma partición en el vector  $x$ .

$$\begin{aligned} X &= [x_B, x_N] \\ Bx_B + Nx_N &= b \end{aligned}$$

En el espacio de las variables no básicas  $x_N$ , la ecuación de  $E$  es

$$\begin{aligned} (x_N - a_N)^T Q (x_N - a_N) &= 1 \\ Q &= D_N^{-2} + N^T B^{-T} D_B^{-2} B^{-1} N \end{aligned}$$

que no es otra cosa que la fórmula (8) del elipsoide aplicada al politopo equivalente dado por las siguientes desigualdades:

$$\begin{aligned} B^{-1}Nx_N &\leq B^{-1}b \\ -x_N &\leq 0. \end{aligned}$$

### 3. EL ALGORITMO DEL ELIPSOIDE INTERIOR

El método genera una sucesión infinita de puntos interiores  $\{a_k\}$ ,  $a_{k+1} = f(a_k)$ . En la figura 1 se observa una representación gráfica del método en dos dimensiones. La función  $f$  es calculada para el problema (7) como sigue:

I. Se definen las matrices definidas positivas

$$\begin{aligned} D &= \text{diag}(b_1 - A_1^T a, \dots, b_m - A_m^T a) \\ Q &= AD^{-2}A^T \end{aligned}$$

y se resuelve el sistema

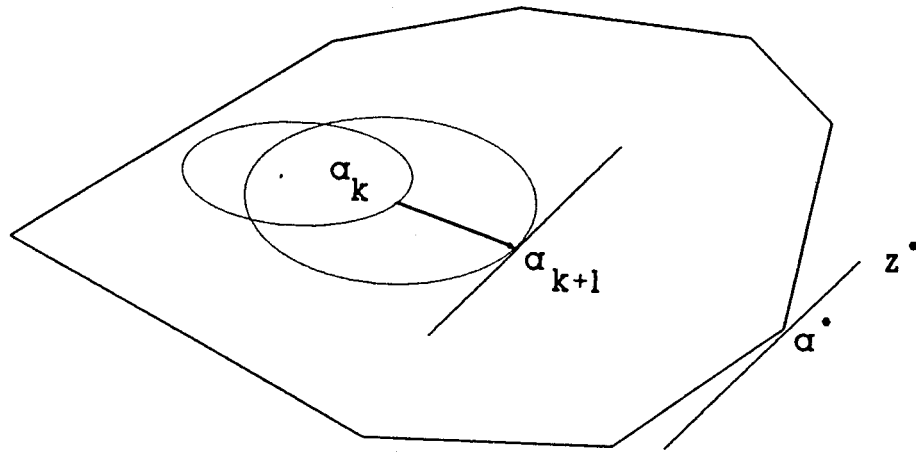
$$Qd = c$$

II. Se calcula la longitud de paso

$$\alpha = (c^T d)^{-1/2}$$

III. Se obtiene el nuevo punto  $f(a)$

$$f(a) = a + \alpha d$$



**Figura 1.** Representación gráfica del Método del Elipsoide Interior.

Para el problema en su formulación estándar (9) el algoritmo es el siguiente:

**I.** Se definen las matrices

$$(10) \quad D = \text{diag} (a_1, a_2, \dots, a_n)$$

$$B = AD$$

y se calcula la dirección de avance

$$(11) \quad d = D(I - B^T (BB^T)^{-1} B)Dc$$

**II.** Cálculo de la longitud de avance

$$\alpha = (c^T d)^{-1/2}$$

**III.** Nuevo punto  $f(a)$

$$f(a) = a - \alpha d$$



Para completar el método es necesario definir el procedimiento que permita obtener el punto de partida y fijar el criterio de finalización. El procedimiento de inicialización es similar a la fase I del método del simplex. Consiste en definir un programa lineal artificial equivalente asociado al problema de admisibilidad  $Ax = b, x \geq 0$  (Karmarkar, 1984). Considerando cualquier punto  $a_0$  de  $\mathbb{R}^n$  con todas sus componentes positivas y el problema

$$(12) \quad \begin{array}{l} \text{Minimizar } \sigma \\ \text{restringido a } Ax + (b - Aa_0)\sigma = b \\ x, \sigma \geq 0 \end{array}$$

donde  $\sigma \in \mathbb{R}$ . Un punto inicial para (12) es  $x = a_0$  y  $\sigma = 1$ . Comenzando con él y utilizando el algoritmo del elipsoide interior para resolver el problema se llega a la solución de la primera fase. Si en ésta,  $\sigma > 0$ , el problema original no tiene solución admisible. En otro caso,  $\sigma = 0$  y el punto solución es un punto inicial para el problema original (9). El algoritmo global es convergente (Barnes, 1986, Vanderbei *et al*, 1985).

#### 4. CÁLCULO DE LAS VARIABLES DUALES POR MÍNIMOS CUADRADOS

Es conocido que el método de simplex proporciona en cada iteración una solución básica no admisible del problema dual junto con la solución primal. Al final del algoritmo se consigue la factibilidad de la solución dual. En el método del elipsoide interior es fácil obtener una solución inadmisibile del problema dual en cada iteración. La sucesión de soluciones duales se utilizará para definir el criterio de finalización. Para encontrar la solución del problema dual, es necesario resolver un problema de mínimos cuadrados derivado de la norma que define la matriz  $D$  en (10).

Dado un punto interior  $a$ , se considera el siguiente sistema rectangular obtenido a partir de (9),  $DA^T\pi = Dc$ , en términos de mínimos cuadrados

$$(13) \quad \min_{\pi} \|Dc - DA^T\pi\|$$

cuya solución es,

$$(14) \quad \pi = (AD^2A^T)^{-1} AD^2c.$$

Este vector se obtiene en el proceso descrito en (11). En cada iteración es posible definir el vector de costes reducidos asociados al vector  $\pi$ , o sea el vector de residuos del problema (13).

$$(15) \quad r = Dc - DA^T\pi$$

y  $s = D^{-1}r = c - A^T\pi$ , son los costes reducidos del problema lineal (9). De hecho, la dirección de avance determinada en (11) puede escribirse en función de este vector,  $d = -Dr = -D^2s$ . Los costes reducidos se calculan en cada iteración sin ninguna operación extra.

## 5. CRITERIO DE FINALIZACIÓN

El algoritmo del elipsoide interior genera una sucesión infinita de puntos interiores (soluciones primales) y soluciones duales que convergen al óptimo del problema lineal. Junto con estas últimas, proporciona otra sucesión de costes reducidos  $s_k$  que tienden a cumplir las condiciones de optimalidad,  $\{s_k\} \rightarrow s^* \geq 0$ . Esta propiedad es la que se emplea para fijar el final del proceso.

Una forma sencilla de implementarlo, consiste en definir una constante positiva  $\epsilon$  que se escogerá en función de la precisión del medio de cálculo del que se dispone y de la precisión que se quiera obtener en la solución final. El proceso finaliza cuando se cumple la siguiente condición,  $s_k \geq -\epsilon$ . Se puede observar que no existe diferencia entre este criterio y el utilizado en el algoritmo del simplex. Es justamente la aplicación de las condiciones de Kuhn-Tucker. Una ventaja de este procedimiento es que no necesita cálculos adicionales, debido a que en cada iteración se obtienen los costes reducidos. El siguiente teorema proporciona la base teórica necesaria que justifica el criterio de finalización elegido.

### **Teorema.**

Sea  $a \in P$  un punto estrictamente interior y por tanto no óptimo para (9). Entonces el vector de costes reducido asociado  $s$  tiene al menos un elemento negativo.

### **Demostración:**

Si  $P$  está acotado, existe un vector positivo  $v \in \mathbb{R}^n$  que cumple  $v^T x = 1$ , y como la dirección  $-D^2 s$  es admisible para  $P$ , o sea pertenece al núcleo de  $A$ , se verifica que

$$\sum v_j a_j^2 s_j = 0$$

y como tanto  $v_j$  como  $a_j$  son estrictamente positivos, entonces o bien existe algún  $s_j$  negativo o todos son cero, en este caso  $c$  es una combinación lineal de las filas de la matriz  $A$  y el objetivo es constante en  $P$  y por tanto cualquier punto es óptimo en contradicción con el enunciado del teorema. ■

## **6. RESOLUCIÓN DEL PROBLEMA DE MÍNIMOS CUADRADOS**

Para problemas de dimensión media y superior, el tiempo de ejecución de una iteración del algoritmo está dominado por la solución del problema de mínimos cuadrados (13) o lo que es lo mismo, el cálculo de la dirección de avance (11).

Es imprescindible, si se desea que el algoritmo sea eficiente, diseñar métodos que adaptándose a las características peculiares de los problemas que la programación lineal genera, los resuelva de manera eficiente. La rapidez de éstos condiciona la del algoritmo.

### **6.1. Primer Método: Actualización Incompleta de la Descomposición Triangular de $AD^2A^T$**

La solución del problema de mínimos cuadrados (13) se obtiene resolviendo el sistema de ecuaciones

$$(16) \quad AD^2A^T \pi = D^2c.$$

Si  $A$  es de rango máximo y se tiene en cuenta que la matriz  $D$ , por construcción, tiene todos sus elementos de la diagonal principal estrictamente positivos, la matriz resultante  $Q = AD^2A^T$  es una matriz simétrica de rango máximo y definida positiva. Un método para obtener la solución de este sistema es la factorización de la matriz  $Q = LL^T$ , donde  $L$  es una matriz triangular inferior. El vector  $\pi$  se obtiene resolviendo los dos sistemas triangulares,  $Lw = D^2c$  y  $L^T \pi = w$ .

De una iteración a otra, la única variación del sistema (16) surge en la matriz diagonal  $D$ , cuyas componentes son las coordenadas del punto de partida de cada iteración. Esto aconseja, como sugirió Karmarkar (1984), aprovechar la solución y descomposición de la iteración anterior para acelerar los cálculos de la siguiente. Es interesante realizar alguna observación sobre el comportamiento geométrico del algoritmo. En primer lugar, la mayor parte de las iteraciones transcurren en las proximidades del óptimo y los puntos consecutivos se encuentran suficientemente cercanos como para que en algún apartado de los cálculos se pueda tomar uno de ellos como aproximación del otro. En segundo lugar, y de acuerdo con la teoría de Meggido y Shub (Gay, 1987), el camino que producen los distintos puntos soluciones es casi paralelo a las caras (aristas) del politopo. Esto significa que los cambios entre dos puntos consecutivos, como se comprueba en los resultados numéricos, sólo afectan a unas pocas componentes del vector. Aprovechando este comportamiento, se ha diseñado un método que partiendo de la descomposición triangular de Cholesky de la matriz  $Q$  en una iteración, conduce a la solución de los siguientes sistemas de forma aproximada mediante actualizaciones de rango uno de los factores.

#### **Esquema de Actualización.**

Se supone que se tiene la descomposición de Cholesky de la matriz  $Q = LL^T$  para una matriz diagonal dada  $D$ , definida en (10) y después de una iteración, el nuevo punto  $\bar{a}$ , da lugar a la nueva matriz diagonal

$$\bar{D} = \text{diag}(\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n),$$

y a partir de ella construye la siguiente matriz,

$$D = \text{diag}(\hat{a}_1, \hat{a}_2, \dots, \hat{a}_n)$$

donde

$$\hat{a}_j = a_j \text{ si } |\log(\bar{a}_j/a_j)| \leq \log(\sigma)$$

en caso contrario

$$\hat{a}_j = \bar{a}_j$$

siendo  $\sigma$  un número real positivo constante mayor que 1 que se determina empíricamente. Cuanto mayor sea el valor  $\sigma$  peor es la aproximación y viceversa.

El significado de la aproximación es el siguiente, si por ejemplo tomamos  $\sigma = 2$ , se mantendrá el valor de la antigua coordenada del punto siempre que ésta no se modifique en más del doble o en menos de la mitad y en el caso de que el cambio producido sea mayor se tomará el nuevo valor. No es necesario que  $\sigma$  sea constante en todas las iteraciones y, en muchos casos, es más adecuado fijar un valor para las primeras iteraciones, en las cuales la longitud de paso es mayor y por lo tanto se producen un mayor número de cambios, y otro valor menor para las iteraciones que se producen en las proximidades de la frontera donde los cambios entre dos puntos consecutivos son más sutiles.

Sea  $J$  es el conjunto de índices de las variables que se han modificado según el criterio indicado. Se tiene que

$$(17) \quad \begin{aligned} \hat{Q} &= A\hat{D}^2 A^T = AD^2 A^T + A(\hat{D}^2 - D^2) A^T \\ \hat{Q} &= Q + \sum_{j \in J} (\hat{a}_j^2 - a_j^2) A_j A_j^T \end{aligned}$$

donde  $A_j$  es la columna  $j$  de la matriz  $A$ . Si el número de índices que han variado es  $p$ , la nueva descomposición se consigue con  $p$  actualizaciones de rango uno.

Si se supone que  $p = 1$  y se llama  $\beta$  al valor del paréntesis en (17) se tiene

$$\hat{Q} = Q + \beta A_j A_j^T = LL^T + \beta A_j A_j^T$$

y si  $v$  es el vector solución de  $Lv = A_j$ , entonces

$$(18) \quad \hat{Q} = L(I + \beta vv^T) L^T$$

pudiéndose descomponer

$$(19) \quad I + \beta vv^T = (I + \mu vv^T)(I + \mu vv^T)$$

tomando  $\mu$  el valor

$$\mu = \frac{\beta}{1 + (1 + \beta v^T v)^{1/2}}.$$

La expresión bajo la raíz cuadrada es un múltiplo positivo del determinante de  $Q$ . Si  $Q$  es definida positiva  $\mu$  es real. Sustituyendo (19) en (18),

$$\hat{Q} = L(I + \mu vv^T)(I + \mu vv^T) L^T$$

y si  $E = I + \mu vv^T$

$$\begin{aligned} P &= LE, \\ \hat{Q} &= PP^T \end{aligned}$$

Si  $p$  es mayor que uno, entonces

$$P = LE_1 E_2 \cdots E_p$$

donde  $E_k$  se obtiene de la siguiente manera, (se ha supuesto para simplificar que los índices  $j$  que han variado son los  $p$  primeros),

$$\hat{Q} = LE_1 \cdots E_{k-1} E_{k-1} \cdots E_1 L^T + \sum_{j=k}^p \beta_j A_j A_j^T$$

se calcula  $v_k$ , de forma que

$$LE_1 \cdots E_{k-1} v_k = A_k$$

sabiendo que la inversa de  $(I + \mu vv^T)$  por la fórmula de Sherman-Morrison es

$$(I + \mu vv^T)^{-1} = I - \frac{\mu}{1 + \mu v^T v} vv^T,$$

el resto del proceso es similar al caso  $p = 1$ . En lugar de obtener la inversa se puede triangular cada  $E_k$  mediante una secuencia de matrices de Givens, la descomposición de Cholesky o matrices de Householder. Cualquiera de estos procedimientos necesita realizar un número mayor de operaciones siendo preferible el propuesto.

### Algoritmo.

Se tiene la factorización de la matriz  $Q = LL^T$  o bien una actualización de ella

$$Q = P_0 P_0^T = L E_1 \cdots E_n E_n \cdots E_1 L^T$$

y se ha fijado previamente el valor  $\sigma$  que determina si una variable ha cambiado considerablemente.

Sea  $p$  el número de actualizaciones que son necesarias realizar en una iteración cualquiera:

1. Inicialización (Factorización de Cholesky).

$$\begin{aligned} Q &= P_0 P_0^T \\ h &= 0 \end{aligned}$$

2. Resolver el sistema

$$P_h v_h = A_h$$

3. Calcular el valor  $\mu_h$  según (1014).

$$E_h = (I + \mu_h v_h v_h^T)$$

4. Actualizar

$$P_{h+1} = P_h E_h$$

Si  $p$  es mayor que  $h$ , entonces  $h = h + 1$  e ir a 2.

En caso contrario, FIN.

El proceso se completa con un procedimiento de reinversión. Conforme el método progresa, el número de matrices de actualización  $E_k$  aumenta y el paso 2 del algoritmo se va haciendo más lento e inexacto y, por lo tanto, es conveniente cada cierto número de iteraciones del algoritmo de optimización, eliminar el archivo de matrices  $E_k$ , realizar una descomposición triangular de la matriz  $Q$  completa y reiniciar el proceso. La aplicación del teorema del “Paso Espaciado” (Luenberger, 1973) es suficiente para demostrar que el algoritmo actualizado converge si converge el algoritmo base.

## 6.2 Segundo Método: Aplicación del Algoritmo LSQR

En contraste con el método anterior se propone un método de carácter iterativo. Los métodos iterativos generan una secuencia de soluciones aproximadas que convergen a la solución final.

El método del gradiente conjugado es apropiado en la resolución del problema (13) y (16). Desde el punto de vista computacional tiene una serie de atractivos que le diferencian de otros métodos:

- (1) Se puede trabajar directamente con la matriz  $A$  sin necesidad de formar la matriz  $Q$ . Esto supone un ahorro de número de operaciones considerable frente a los métodos directos.
- (2) Se consigue un gran aprovechamiento de la dispersión de la matriz  $A$ , debido a que en este método sólo se realizan productos de la forma  $Av$  y  $A^T u$  para distintos vectores  $u$  y  $v$ .
- (3) Desde el punto de vista computacional es de destacar el ahorro de memoria central que supone la aplicación de este método frente a otros, debido a que únicamente es necesario almacenar unos pocos vectores en el área de trabajo.
- (4) La aplicación del método del gradiente conjugado a la resolución de sistemas lineales de ecuaciones converge a la solución teóricamente en  $n$  iteraciones como máximo (Gill *et al.*, 1981).
- (5) Es posible, a diferencia del método aproximado anterior, aplicarlos al problema degenerado (cuando la matriz  $A$  no tiene rango máximo).

El método tiene la siguiente propiedad (Luenberger, 1973): si se suponen exactas las operaciones aritméticas realizadas, la convergencia se produce en un número de iteraciones igual al número de valores propios distintos de  $Q$ . Por tanto, el método del gradiente conjugado puede ser considerado como un método directo. Sin embargo, en la práctica los errores de redondeo causan rápidamente que las direcciones calculadas dejen de ser conjugadas y el método se comporte como un algoritmo iterativo. Se puede concluir, que el método converge a la solución en muy pocas iteraciones si los valores propios de  $Q$  en (16) están agrupados en grupos de aproximadamente el mismo valor; y debido a los errores de redondeo, el método convergerá en considerablemente más de  $m$  iteraciones si los valores propios de  $Q$  son muy distintos.

Con las razones anteriores se puede concluir que la aplicación del algoritmo del gradiente conjugado no es el sistema más eficiente para resolver el problema



planteado. Es necesario diseñar un algoritmo que explote las ventajas mencionadas y a su vez elimine los inconvenientes propios del método. Gay (1987) y Gill *et al.*, (1986) han propuesto la resolución del problema de mínimos cuadrados que sus respectivos algoritmos plantean empleando el método LSQR de Paige y Saunders (1982). El algoritmo del Elipsoide Interior en este caso tiene el siguiente esquema.

0. Inicialización.

$a_0$  es un punto estrictamente interior.

$$r_0 = c$$

$$D_0 = \text{diag}(a_{01}, \dots, a_{0n})$$

$$k = 0.$$

1. Cálculo del Precondicionador  $H_k$ .

Calcular  $H_k$  (triangular superior) tal que

$$AD_k^2 A^T \approx H_k H_k^T$$

2. Resolución del Problema de Mínimos Cuadrados mediante LSQR.

$$\delta_{k+1} : \text{mínimo } \|D_k r_k - D_k A^T H_k^{-1} \delta_{k+1}\|$$

3. Actualización de  $r_{k+1}$ .

$$r_{k+1} = r_k - A^T \delta_{k+1}$$

4. Nueva dirección de Avance.

$$d_{k+1} = D_k r_{k+1}$$

5. Cálculo del paso de avance ( $\alpha_{k+1}$ ).

6. Nuevo Punto.

$$a_{k+1} = a_k + \alpha_{k+1} d_{k+1}$$

7. Criterio de Terminación.

Si  $(r_{k+1}) \geq -\epsilon$  FIN

En caso contrario

$$D_{k+1} = \text{diag}(a_{k+1})$$

Hacer  $k = k + 1$  e ir a 1.

El primer paso del algoritmo solo debe efectuarse cuando el preconditionador haya perdido sus propiedades (ver Paige y Saunders 1982, Golub y Van Loan, 1982).

## 7. RESULTADOS

El objetivo final de los métodos de cálculo numérico es resolver los problemas reales de manera veloz y fiable. La teoría de la optimización y, más en general, de los métodos numéricos no ha ofrecido hasta el momento un resultado concluyente que permita evaluar el comportamiento práctico de los algoritmos. Buena prueba de ello se obtiene si se intentara juzgar la eficiencia del método del simplex a través de su complejidad. Admitiendo los muchos inconvenientes que representa –desde el punto de vista de rigor matemático– el extraer conclusiones de naturaleza empírica, éste es hasta el momento el único camino posible.

Con esta finalidad se han desarrollado dos aplicaciones informáticas con las que se han resuelto un conjunto de programas lineales test generados aleatoriamente. Los resultados obtenidos permiten extraer algunas conclusiones respecto a la efectividad de los métodos expuestos. La primera corresponde al algoritmo del elipsoide interior e incluye los dos procedimientos analizados para resolver el problema de mínimos cuadrados. La segunda, el simplex, se empleará como patrón de referencia. La versión del simplex utilizada, simplex revisado disperso con factorización LU de la base y actualización, es la más adecuada para resolver problemas lineales de dimensión media. Las dos aplicaciones se han programado en FORTRAN 77 y se han ejecutado en un ordenador IBM PC AT.

Se han resuelto dos tipos distintos de problemas lineales, programas generales o sin estructura especial en la matriz de restricciones y problemas cuya matriz de restricciones tiene estructura escalonada, también denominados dinámicos. En los dos casos los problemas se generan en su forma estándar (9). En la Tabla 1 se definen las características de los problemas del primer tipo.

**Tabla 1**  
**Características del Problema.**  
(sin estructura especial)

PROBLEMA	Num. de Filas	Num. de Columnas	Elementos no nulos
Gen 1	27	50	229
Gen 2	50	80	466
Gen 3	60	77	490
Gen 4	70	105	599
Gen 5	80	100	465
Gen 6	90	110	579
Gen 7	100	125	487
Gen 8	120	140	645
Gen 9	150	220	1553
Gen 10	190	430	3239

En las Tablas 2 y 3 se presentan los resultados obtenidos empleando las correspondientes aplicaciones informáticas. En ellas se puede observar la gran diferencia en cuanto a número de iteraciones entre el método del simplex y el método del elipsoide interior. A pesar de esto, el primero alcanza la solución óptima en menos tiempo que el segundo. Las razones que justifican este hecho es el elevado número de operaciones que el método del elipsoide interior debe efectuar en cada iteración.

**Tabla 2**  
**Resultados del Método del Simplex.**  
 (sin estructura especial)

PROBLEMA	Iter. FASE I	Iter. TOTAL	Tiempo (s) de Cálculo	Objetivo
Gen 1	50	69	27	$.24217245 \times 10^2$
Gen 2	128	166	71	$.44616823 \times 10^3$
Gen 3	159	171	98	$.51814465 \times 10^2$
Gen 4	327	393	499	$.37096256 \times 10^2$
Gen 5	249	275	372	$.16807412 \times 10^1$
Gen 6	312	355	766	$.20241443 \times 10^3$
Gen 7	405	414	673	$.21508856 \times 10^3$
Gen 8	357	480	1332	$.35668732 \times 10^2$
Gen 9	573	622	1601	$.67543098 \times 10^3$
Gen 10	876	993	3281	$.72366551 \times 10^3$

**Tabla 3**  
**Resultados del Algoritmo del Elipsoide Interior.**  
 (sin estructura especial)

PROBLEMA	Iter. FASE I	Iter. TOTAL	Tiempo (s) de Cálculo	Objetivo
Gen 1	2	25	28	$.24218124 \times 10^2$
Gen 2	4	78	243	$.44616543 \times 10^3$
Gen 3	4	42	204	$.51814431 \times 10^2$
Gen 4	4	113	549	$.37096432 \times 10^2$
Gen 5	4	44	277	$.16808654 \times 10^1$
Gen 6	4	39	441	$.20242345 \times 10^3$
Gen 7	3	69	813	$.21508466 \times 10^3$
Gen 8	7	181	2212	$.35659789 \times 10^2$
Gen 9	5	221	7342	$.67487634 \times 10^3$
Gen 10	6	244	9863	$.72366198 \times 10^3$

En la Tabla 4 se presentan las características que definen los problemas en escalera resueltos y en las Tablas 5 y 6 los resultados de aplicar los dos algo-

ritmos que se comparan. Se puede comprobar que los tiempos de cálculo son sensiblemente inferiores para el algoritmo del elipsoide interior.

**Tabla 4**  
**Características del problema.**  
 (Estructura en escalera)

PROBLEMA	Núm. de	Filas p.	Columnas	Filas	Columnas	Element. no nulos
	Periodos	Periodo	p. Perio	Total	Total	
Esc 1	10	6	8	60	80	459
Esc 2	20	3	5	60	100	525
Esc 3	15	4	6	60	90	507
Esc 4	6	12	20	72	120	899
Esc 5	4	25	30	100	120	1753
Esc 6	10	12	15	120	150	1538
Esc 7	12	12	15	144	180	1865
Esc 8	20	8	15	160	300	2619
Esc 9	24	8	17	192	408	3853
Esc 10	30	6	15	180	450	3561

**Tabla 5**  
**Resultados del Método del Simplex.**  
 (Estructura en escalera)

PROBLEMA	Iter.	Iter.	Tiempo (s)	Objetivo
	FASE I	TOTAL	de Cálculo	
Esc 1	91	94	81	$.22179923 \times 10^2$
Esc 2	71	79	49	$.56425972 \times 10^1$
Esc 3	85	87	68	$.64514443 \times 10^2$
Esc 4	64	106	106	$.16440282 \times 10^2$
Esc 5	218	257	837	$.10867934 \times 10^3$
Esc 6	232	267	860	$.20980271 \times 10^2$
Esc 7	287	328	1280	$.14475962 \times 10^2$
Esc 8	345	356	1436	$.64343312 \times 10^1$
Esc 9	516	539	1258	$.33214943 \times 10^1$
Esc 10	361	385	912	$.18706354 \times 10^1$

**Tabla 6**  
**Resultados del Algoritmo del Elipsoide Interior.**  
(Estructura en escalera)

PROBLEMA	Iter. FASE I	Iter. TOTAL	Tiempo (s) de Cálculo	Objetivo
Esc 1	10	15	55	$.22172923 \times 10^2$
Esc 2	3	7	31	$.56426919 \times 10^1$
Esc 3	5	7	38	$.64514445 \times 10^2$
Esc 4	8	17	61	$.16440214 \times 10^2$
Esc 5	4	18	391	$.10867943 \times 10^3$
Esc 6	5	19	338	$.20980276 \times 10^2$
Esc 7	5	14	477	$.14476045 \times 10^2$
Esc 8	5	12	407	$.64344103 \times 10^1$
Esc 9	4	16	851	$.33214865 \times 10^1$
Esc 10	6	19	644	$.18707550 \times 10^1$

Una ventaja que presentan los métodos de puntos interiores frente al método del simplex, de carácter combinatorio, es que en aquéllos es posible obtener una buena aproximación a la solución óptima si se detiene el proceso antes de terminar, mientras que la finalización anticipada del método de Dantzig no supone necesariamente que el valor obtenido sea una buena aproximación a la solución óptima. Este hecho sugiere el siguiente procedimiento mixto:

- (I) Utilizar el algoritmo de elipsoide interior hasta conseguir una solución admisible suficientemente próxima al óptimo. La mayor parte de las iteraciones de los métodos de puntos interiores transcurren por puntos próximos al vértice óptimo del politopo. La medida más sencilla de la proximidad de una solución a la solución óptima, si se conoce ésta es el valor de la función objetivo. Con menos de la mitad de las iteraciones, el algoritmo del elipsoide interior calcula una solución con, al menos, las dos primeras cifras significativas de este valor iguales que las del óptimo. Otro indicador del grado de optimalidad de una solución es (Gill *et al.*, 1986),

$$\frac{\|D(c - A^T \pi)\|}{\|x\| \|\pi\|}$$

- (II) Una vez calculada una “buena” solución admisible, el siguiente paso es su conversión a una solución básica. Al definir una base se asigna el valor cero a las variables no básicas y se obtiene el vector de variables básicas como solución de un sistema de ecuaciones lineales. Se han desarrollado distintos procedimientos que partiendo de una solución no básica obtienen una solución básica con un valor de la función objetivo igual o menor. El procedimiento es conocido como “inserción de la solución” (Benichou *et al.*, 1977).
- (III) Si la solución básica obtenida no es la óptima del programa lineal, se aplica el algoritmo del simplex. El óptimo se alcanza en muy pocas iteraciones.

## 8. CONCLUSIONES

En este artículo se deduce y analiza un algoritmo para la resolución de programas lineales de tipo general. El principal resultado es la definición misma del método y la obtención de la fórmula del elipsoide interior a un polítopo. Se trata de un procedimiento de puntos interiores que se ha inspirado en el algoritmo de Karmarkar. El contenido geométrico del algoritmo es notable y en su construcción se emplean nociones elementales de Álgebra lineal. Su conexión con el método de mínimos cuadrados es evidente, y cabe esperar futuras interpretaciones de su contenido en este contexto. Otro resultado de este estudio es la incorporación al esquema del algoritmo del elipsoide interior de dos métodos de mínimos cuadrados: el primero de ellos basado en la actualización incompleta de la descomposición triangular de Cholesky de una matriz simétrica y el segundo aplicando el procedimiento LSQR de Paige y Saunders (1982).

De los experimentos realizados con las aplicaciones informáticas se puede concluir que el método del elipsoide interior es un procedimiento menos eficiente que el algoritmo del simplex cuando se resuelven programas lineales generales, aunque la distancia que los separa no es tan grande (Tabla 1,2,3) que no pueda ser superada.

Los problemas para los que el método del elipsoide es superior al simplex son aquellos en los que el problema de mínimos cuadrados puede ser resuelto rápidamente, en particular para aquellos programas lineales cuya matriz de restricciones tiene estructura en escalera, es en este caso cuando los métodos de puntos interiores son más prometedores.

## 9. BIBLIOGRAFÍA

- [1] **Barnes, E.R.** (1986). “A variation on Karmarkar’s algorithm for solving linear programming problems”. *Mathematical Programming*, 36, pp. 174-182
- [2] **Benichou, M. et al.** (1977). “The efficient solution of large -scale linear programming problems- some algorithmic techniques and computational results”. *Mathematical Programming*, vol. 13, pp. 280-322.
- [3] **Dantzig, G.B.** (1951). “Maximization of a linear function of variables subject to linear inequalities”. En *Activity Analysis of Production and allocation*, editado por Koopmans T.C., pp. 339-347, Wiley, New York.
- [4] **Fiacco, A.V. y McCormick, G.P.** (1968). “Nonlinear Programming: Sequential Unconstrained Minimization Techniques”. John Wiley and Sons, Nueva York y Toronto.
- [5] **Gay, D.M.** (1987). “Pictures of Karmarkar’s Linear Programming Algorithm”. *Computing Science Technical Report* Núm. 136, AT & T Bell Laboratories, New Jersey.
- [6] **George, J.A. y Wright, M.H.** (1981). *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, New Jersey.
- [7] **Gill, P.E., Murray, W. y Wright, M.H.** (1981). “Practical Optimization”. Academic Press, San Francisco.
- [8] **Gill, P.E., Murray, W., Saunders, M.A., Tomlin, J.A. y Wright, M.H.** (1986). “On projected Newton Barrier Methods for Linear Programming and an Equivalence to Karmarkar’s Projective Method”. *Technical Report*, SOL 85-11R.
- [9] **Gill, P.E., Murray, W., Saunders, M.A. y Wright, M.H.** (1986). “A note on nonlinear approaches to linear programming”. *Report*, SOL 86-7, Department of Operations Research, Stanford University.
- [10] **Gill, P.E., Golub, H.G., Murray, W. y Saunders, M.A.** (1974). “Methods for Modifying Matrix Factorizations”. *Mathematics of Computation*, V. 28, Núm. 126, pp. 505-535.
- [11] **Gill, H.G., Murray, W. y Saunders, M.A.** (1985). “Methods for Computing and Modifying the LDV Factors of a Matrix”. *Mathematics of Computation*, V. 29, Núm. 132, pp. 1051-1077, 1975.
- [12] **Golub, G.H. y Van Loan, C.F.** (1982). “Matrix Computations”. North Oxford Academic, Oxford.
- [13] **Heath, M.T.** (1984). “Numerical Methods for Large Sparse Linear Least Squares Problems”. *SIAM J. on Scientific and Statistical Computing*, 5, pp. 497-513.



- [14] **Kallio, M. y Orchard-Hays, W.** (1981). “Experiments with the reduced gradient method for general and dynamic linear programming”. En *Large scale linear programming*, Dantzig et al. Eds, IIASA Collaborative Proceedings series, CP-81-51, IIASA, Lasenburg, Austria.
- [15] **Karmarkar, N.** (1984). “A new polynomial-time algorithm for linear programming”. *Combinatorica*, 4, pp. 373-395.
- [16] **Luenberger, D.G.** (1973). “Introduction to linear and nonlinear programming”. Addison Wesley, Reading, Massachusetts.
- [17] **Paige, C.C. y Saunders, M.A.** (1982). “LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Square”. *ACM Transactions on Mathematical Software*, vol. 8, Núm.1.
- [18] **Vanderbei, R.J., Meketon, M.S. y Freedman, B.A.** (1985). “A modification of Karmarkar’s linear programming algorithm”. *Algorithmica*, 1, pp. 395-407.

