TRABAJOS DE
INVESTIGACION OPERATIVA
Vol. 1. Núm. 1, 1986, pp. 23 a 49

UN ALGORITMO DE PROGRAMACION GEOMETRICA BASADO EN FUNCIONES PENALIDAD-MULTIPLICADORAS (*)

Eduardo Ramos Méndez
Cátedra de Investigación Operativa
Facultad de Químicas y Matemáticas
Universidad de Murcia

RESUMEN

El trabajo presenta un nuevo algoritmo para la resolución de un problema de programación geométrica primal transformado. El método se basa en las técnicas de tipo Lagrangiano aumentado y utiliza como penalidad funciones derivadas de la exponencial para las restricciones con un único término, y de la pérdida cuadrática para las restricciones con más de un término. El problema resultante se resuelve por medio de un método lagrangiano con iteración de tipo Newton, y los parámetros de penalización se actualizan mediante una fórmula inspirada en las condiciones de optimalidad de primer orden. Se incluye alguna experiencia computacional.

Palabras clave: Programación Geométrica, Algoritmos, Langragianos Aumentados.
Clasificación AMS: 90C30.

SUMMARY

This paper presents a new algorithm for the resolution of a transformed primal geometric programming problem. The method is based on augmented Langrangian function techniques and as penalty uses modifications of the exponential penalty function for the constraints with a single term and of the

(*) Recibido, abril 1985. Aceptado, octubre 1985.

quadratic penalty function for the constraints with more than one term. The penalized problem is solved by a lagrangian method with Newton's iteration and the penalty parameters are updated according to a formula inspired on the first order optimality conditions. Some computational experience is included.

Key words: Geometric Programming, Algorithms, Augmented Lagrangian Functions.

1. INTRODUCCION

La Programación Geométrica (PG) se ha utilizado ampliamente en los últimos años como herramienta para la formulación de numerosos modelos de optimización que surgen en diferentes campos de aplicación. (Véase, por ejemplo: Beightler y Phillips, 1976; Rijckaert y Martens, 1978b; Smeers y Tyteca, 1984.) Consecuentemente se han desarrollado y comparado varios métodos numéricos de resolución de problemas de PG (Dembo, 1976, 1978, 1979; Rijckaert y Martens, 1978a; Sarma y otros, 1978; Fattler y otros, 1982; Ecker, 1984).

En un trabajo anterior (Ramos, 1985) se ha introducido una amplia familia de algoritmos de PG que se basan en la idea de aplicar los métodos del Lagrangiano aumentado de programación no lineal general a los problemas de PG. Esta familia depende fundamentalmente de la formulación del problema geométrico utilizada, de la función penalidad-multiplicador elegida, del método de resolución empleado para resolver el problema resultante y del procedimiento de actualización de los multiplicadores y de los parámetros de penalización.

En este trabajo se describe completamente un algoritmo de esta familia. El formato de programación geométrica utilizado será el problema primal transformado (posinómico); como funciones penalidad-multiplicadoras se emplearán las basadas en la función exponencial para las restricciones con un sólo término (monomios) y en la función pérdida cuadrática para las restricciones con más de un término; el problema de minimización de la función no lineal resultante sometida a un conjunto de restricciones lineales se resolverá utilizando un método Lagrangiano, con iteración de tipo Newton, para calcular puntos estacionarios; y, finalmente, los parámetros de penalización se actualizarán utilizando una regla inspirada en las condiciones de optimalidad de primer orden.

En la sección 2 se desarrollan por completo las fórmulas iterativas de algoritmo; en la sección 3 se comentan algunas experiencias computacionales para finalizar, en la sección 4 con una discusión sobre los resultados y posibles extensiones.

2. EL ALGORITMO

El problema de programación geométrica (posinómico) se presenta usualmente bajo el formato (Duffin, Peterson y Zener, 1967; Peterson, 1978):

PROBLEMA PRIMAL (PP)

$$\begin{array}{l} \text{Minimizar } g_0(t) \\ \text{sujeto a } g_k(t) \leqslant 1 \quad k = 1, \ldots, p \end{array}$$

donde

$$g_k(t) = g_k(t_1, \ldots, t_m) = \sum_{j \in J(k)} c_j \prod_{i=1}^{m} t_i^{a_{ij}}$$

$$J(k) = \{m_k, \ldots, n_k\} \quad (k = 0, 1, \ldots, p)$$

$$m_0 = 1;\ m_{k+1} = n_k + 1, (k = 0, \ldots, p-1);\ n_p = n$$

$$t_i > 0 \quad (i = 1, \ldots, m)$$

a_{ij} números reales cualquiera $(i = 1, \ldots, m; j = 1, \ldots, n)$

c_j números reales positivos $(j = 1, \ldots, n)$

Se denotará $A = (a_{ij})$ la matriz exponente y $c^t = (c_1, \ldots, c_n)$ el vector de coeficientes de los términos.

Otra posible formulación que se utilizará en este trabajo para desarrollar el algoritmo es la siguiente:

PROBLEMA PRIMAL TRANSFORMADO (PPT)

$$\text{Minimizar } f_0(\omega) = \sum_{j \in J(0)} \exp(\omega_j)$$

$$\begin{array}{l} \text{sujeto a } f_k(\omega) \leqslant 0 \quad k = 1, \ldots, p \\ C(\omega - \log c) = 0 \end{array}$$

donde

$$f_k(\omega) = f_k(\omega_1, ..., \omega_n) = \sum_{j \in J(k)} \exp(\omega_j) - 1$$

$$\log c = (\log c_1, ..., \log c_n)^t$$

C es una matriz $(n - m) \times n$ cuyas filas generan el espacio nulo de la matriz A, es decir,

$$CA^t = 0$$

Los problemas PP y PPT son equivalentes cuando A tiene rango máximo. Esto puede comprobarse mediante el cambio (Duffin, Peterson y Zener, 1967):

$$t_i = \exp(z_i)$$
$$\omega = A^t z + \log c$$

En base a las ideas expuestas en Ramos, 1985, el método de resolución que vamos a emplear consiste en introducir en la función objetivo las restricciones no lineales del PPT convenientemente penalizadas de forma que se transforme en un tipo de Lagrangiano aumentado. El problema de programación no lineal con restricciones lineales se resolverá según un esquema de Newton-Lagrange.

Funciones penalidad-multiplicadoras

Sea $K_1 = \{k | m_k = n_k\} \subset \{1, 2, ..., p\}$, el conjunto de índices de restricciones con un sólo término y sea $K_2 = \{k | m_k < n_k\} \subset \{1, 2, ..., p\}$ el conjunto de índices de restricciones con más de un término. Supondremos por simplicidad en las notaciones y sin pérdida de generalidad que:

$$K_1 = \{1, 2, ..., q\} \quad \text{y} \quad \# K_1 = q$$
$$K_2 = \{q + 1, ..., p\} \quad \text{y} \quad \# K_2 = p - q$$

Denotando

$\omega = (\omega_1, \omega_2, ..., \omega_n)^t$ al vector de variables primales
$\sigma = (\sigma_1, \sigma_2, ..., \sigma_p)^t$ al vector de variables duales
$\alpha = (\alpha_1, \alpha_2, ..., \alpha_p)^t$ al vector de parámetros de penalización

la contribución al término de penalidad de cada restricción se define de la siguiente manera:

a) Si $k \in K_1$

$$\Pi_k(\omega, \sigma, a) = \frac{\sigma_k^2}{\alpha_k} h_k \tag{1}$$

donde hemos denotado: $h_k = h_k(\omega) = \exp(\alpha_k^2 \omega_{m_k}) - 1$.

b) Si $k \in K_2$

$$\Pi_k(\omega, \sigma, \alpha) = \begin{cases} -\dfrac{\sigma_k^2}{4\alpha_k} & \text{si } f_k \leqslant -1/2\alpha_k \\ \alpha_k \sigma_k^2 f_k + \sigma_k^2 f_k & \text{si } f_k > -1/2\alpha_k \end{cases} \tag{2}$$

donde $f_k = f_k(\omega)$.

Las funciones anteriores son del tipo penalidad-multiplicador estudiadas por Ramos, 1979, 1981, en donde puede encontrarse una amplia bibliografía sobre el tema. Con estas funciones penalidad-multiplicador el Lagrangiano aumentado viene definido de la manera usual.

Lagrangiano aumentado

$$L(\omega, \sigma, \alpha) = f_0(\omega) + \sum_{k=1}^{p} \Pi_k(\omega, \sigma, \alpha) \tag{3}$$

Podemos formular ahora el problema siguiente:

PROBLEMA PRIMAL TRANSFORMADO PENALIZADO (PPTP)

$$\begin{array}{ll} \text{Minimizar} & L(\omega, \sigma, \alpha) \\ \text{sujeto a} & C(\omega - \log c) = 0 \end{array} \tag{4}$$

siendo la minimización con respecto a las variables primales.

Lagrangiano ordinario del PPTP

Denotando $u = (u_1, u_2, ..., u_{n-m})^t$ al vector de variables duales asociadas a las restricciones de igualdad (4) y $b = C \log c$, el lagrangiano ordinario del PPTP puede escribirse como:

$$F(\omega, \sigma, u, \alpha) = L(\omega, \sigma, \alpha) + u^t\left[(C\;0)\begin{pmatrix}\omega\\ \sigma\end{pmatrix} - b\right] \tag{5}$$

Iteración de tipo Newton

Llamando ∇F y $\nabla^2 F$, respectivamente, al gradiente y matriz hessiana de F respecto a todas las variables de que depende, una iteración típica del método de Newton para calcular puntos estacionarios de F vendrá dada por:

$$\begin{bmatrix}\omega'\\ \sigma'\\ u'\end{bmatrix} = \begin{bmatrix}\omega\\ \sigma\\ u\end{bmatrix} - (\nabla^2 F)^{-1}\,\nabla F \tag{6}$$

donde los vectores ω, σ, u denotan el punto actual y los vectores ω', σ', u' denotan el nuevo punto.

Proposición 1

La iteración (6) es equivalente a resolver el siguiente sistema de ecuaciones lineales:

$$H(\omega' - \omega) + G(\sigma' - \sigma) + C^t(u' - u) = -(V + C^t u) \tag{7}$$

$$G^t(\omega' - \omega) + M(\sigma' - \sigma) = -W \tag{8}$$

$$C(\omega' - \omega) = 0 \tag{9}$$

en donde V es un n-vector y W un p-vector definidos por:

$$V = (\partial L/\partial\omega_1, ..., \partial L/\partial\omega_n)^t \quad ; \quad W = (\partial L/\partial\sigma_1, ..., \partial L/\partial\sigma_p)^t$$

y las matrices $H(n \times n)$, $G(n \times p)$, $M(p \times p)$ son:

$$
\begin{aligned}
H &= (\partial^2 L/\partial\omega_i\partial\omega_j)_{i,j=1,\dots,n} \\
G &= (\partial^2 L/\partial\omega_j\partial\sigma_k)_{j=1,\dots,n;\,k=1,\dots,p} \\
M &= (\partial^2 L/\partial\sigma_k\partial\sigma_l)_{k,l=1,\dots,p}
\end{aligned}
$$

DEMOSTRACION

Es inmediata al tener en cuenta que:

$$
\nabla F = \begin{bmatrix} V + C^t u \\ W \\ (C\,0)\begin{pmatrix}\omega \\ \sigma\end{pmatrix} - b \end{bmatrix}
$$

$$
\nabla^2 F = \begin{bmatrix} H & G & (C\,0)^t \\ G^t & M & 0 \\ (C\,0) & 0 & 0 \end{bmatrix}
$$

Proposición 2

La solución del sistema de ecuaciones lineales (7)-(9), supuesto que existan las matrices inversas indicadas es:

$$
\begin{aligned}
u' &= \{C[H^{-1}G(M - G^tH^{-1}G)^{-1}G^tH^{-1} + H^{-1}]C^t\}^{-1} \cdot \\
&\quad \cdot \{C[H^{-1}G(M - G^tH^{-1}G)^{-1}(W - G^tH^{-1}V) - H^{-1}V]\} \qquad (10) \\
\sigma' &= \sigma + (M - G^tH^{-1}G)^{-1}[G^tH^{-1}(V + C^tu') - W] \qquad (11) \\
\omega' &= \omega - H^{-1}[V + G(\sigma' - \sigma) + C^tu'] \qquad (12)
\end{aligned}
$$

DEMOSTRACION

Simplificando la expresión (7) y despejando ω' se obtiene la fórmula (12). Llevando el valor de ω' dado por (12) a la ecuación (8) ésta puede escribirse de la manera siguiente:

$$
G^t\{-H^{-1}[V + G(\sigma' - \sigma) + C^tu']\} + M(\sigma' - \sigma) = -W
$$

y de aquí al despejar σ' se obtiene la expresión (11). Finalmente combinando (9) y (12) se tiene:

$$C\{-H^{-1}[V + G(\sigma' - \sigma) + C^t u']\} = 0$$

y esta ecuación junto con (11) puede escribirse

$$C\{-H^{-1}[V + G(M - G^t H^{-1} G)^{-1}[G^t H^{-1}(V + C^t u') - W] + C^t u']\} = 0$$

y de aquí al despejar u' se obtiene la fórmula (10).

Las expresiones (10)-(12) nos dan las fórmulas iterativas para, a partir de un punto inicial (ω^0, σ^0) encontrar un punto (ω^*, σ^*) que sea estacionario para el Lagrangiano F y, por tanto, ω^* es la solución del PPT. A pesar de la aparente complejidad de las fórmulas anteriores, la estructura especial del PPT junto con las propiedades de las funciones penalidad-multiplicadoras que estamos utilizando, permitirá utilizar buena parte de los cálculos analíticamente y se simplificarán notablemente las mismas.

En las proposiciones siguientes, Proposiciones 3, 4 y 5, se van a obtener expresiones para las matrices que aparecen en (10)-(12). Para $k \in K_2$ se utilizará para $\Pi_k(\omega, \sigma, k)$ el valor dado en (2) cuando $f_k > -1/2\alpha_k$ ya que si $f_k \leqslant -1/2\alpha_k$ las matrices de derivadas segundas se hacen singulares y no es posible aplicar (10)-(12). Este es un inconveniente típico de este tipo de lagrangiano aumentado. La solución a adoptar en el caso de que se obtenga un punto tal que $f_k \leqslant -1/2\alpha_k$ es manipular convenientemente el parámetro α_k a fin de que pueda obtenerse la desigualdad contraria; esto es especialmente útil en las primeras iteraciones.

Proposición 3

a) El vector V tiene la forma

$$V = (V_0, V_1, \ldots, V_q, V_{q+1}, \ldots, V_p)^t$$

siendo: i) V_0 un n_0-vector cuya j-sima componente es $\exp(\omega_j)$, $j = 1, \ldots, n_0$.

ii) $V_k = \alpha_k \sigma_k^2 (h_k + 1) \quad k \in K_1$.

iii) $V_k = \xi_k \theta_k \quad k \in K_2$.

donde $\xi_k = \sigma_k(2\alpha_k f_k + 1)$ es un escalar y $\theta_k = (\exp(\omega_{m_k}), ..., \exp(\omega_{n_k}))^T$ un $(n_k - m_k + 1)$-vector.

b) El vector W tiene la forma

$$W = (W_1, ..., W_q, W_{q+1}, ..., W_p)$$

siendo: i) $W_k = 2(\sigma_k/\alpha_k)h_k \quad k \in K_1$.

ii) $W_k = 2\sigma_k f_k(\alpha_k f_k + 1) \quad k \in K_2$.

DEMOSTRACION

Inmediata.

Lema 1

(Fórmula de Sherman-Woodbury-Morrison) (McCormick, 1983, página 70).

Sea D una matriz $(s \times s)$ no singular, a un s-vector y $k_1 \neq 0$, k_2 números reales. Entonces se verifica que

$$(k_1 D + k_2 aa^t)^{-1} = \frac{1}{k_1}\left(D^{-1} - \frac{k_2/k_1}{1 + (k_2/k_1)a^t D^{-1} a} D^{-1} aa^t D^{-1}\right)$$

Proposición 4

a) La matriz H^{-1} tiene la forma

$$H^{-1} = \begin{bmatrix} H_0^{-1} & 0 & \cdot & 0 & 0 & \cdot\cdot & 0 \\ 0 & H_1^{-1} & \cdot & & & \cdot\cdot & 0 \\ 0 & & & H_q^{-1} & 0 & \cdot\cdot & 0 \\ 0 & & & 0 & H_{q+1}^{-1} & \cdot\cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\cdot & \cdot \\ 0 & 0 & \cdot & 0 & 0 & \cdot\cdot & H_p^{-1} \end{bmatrix}$$

siendo:

i) H_0^{-1} una matriz diagonal ($n_0 \times n_0$) cuyo j-simo elemento es $\exp(-\omega_j)$, $j = 1, ..., n_0$.

ii) $H_k^{-1} = [\alpha_k^3 \sigma_k^2 (h_k + 1)]^{-1} \quad k \in K_1$.

iii) $H_k^{-1} = \lambda_k (D_k^{-1} - \mu_k e_k e_k^t) \quad k \in K_2 \quad f_k > -1/2\alpha_k$.

en donde: D_k es una matriz diagonal $((n_k - m_k + 1) \times (n_k - m_k + 1))$ cuyo j-simo elemento vale $\exp(\omega_j)$, $j = m_k, ..., n_k$.

$$\lambda_k = 1/\sigma_k^2 \lambda_k' \quad \text{con} \quad \lambda_k' = 2\alpha_k f_k + 1$$
$$\mu_k = 2\alpha_k / \mu_k' \quad \text{con} \quad \mu_k' = 4\alpha_k f_k + 2\alpha_k + 1$$
$$e_k = (1, ..., 1) \quad \text{un } (n_k - m_k + 1)\text{-vector}$$

b) La matriz G tiene la forma

$$G = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ G_1 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & G_p \end{bmatrix}$$

siendo:

i) $G_k = 2\alpha_k \sigma_k h_k \quad k \in K_1$.

ii) $G_k = 2\sigma_k (2\alpha_k f_k + 1) \theta_k \quad k \in K_2 \quad f_k > -1/2\alpha_k$.

c) La matriz M tiene la forma

$$M = \begin{bmatrix} M_0 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ 0 & M_q & \cdots & 0 \\ 0 & & M_{q+1} & 0 \\ 0 & \cdot & \cdots & M_p \end{bmatrix}$$

siendo:

i) $M_k = 2h_k/\alpha_k \quad k \in K_1$.

ii) $M_k = 2f_k(\alpha_k f_k + 1) \quad k \in K_2 \quad f_k > -1/2\alpha_k$.

DEMOSTRACION

a) La estructura diagonal de la matriz H^{-1} se deduce fácilmente de la definición de la función L. Asimismo son inmediatas las expresiones de H_0^{-1} y de H_k^{-1}, $k \in K_1$. Sea $k \in K_2$; entonces

$$H_k = \sigma_k^2(2\alpha_k f_k + 1)D_k + 2\alpha_k\sigma_k^2 0_k 0_k^t$$

Observando que

$$D_k^{-1}0_k = e_k \tag{13}$$

$$0_k^t D_k^{-1} 0_k = 0_k^t e_k = f_k + 1 \tag{14}$$

la parte iii) es consecuencia de aplicar el lema 1 a la matriz H_k anterior.

b) y *c*) Inmediato.

Las proposiciones anteriores nos dan las expresiones de los vectores V y W y de las matrices H, G y M que aparecen en las fórmulas (10)-(12). Vamos ahora a efectuar las operaciones matriciales indicadas en las fórmulas iterativas. Calculamos en primer lugar la matriz y el vector del lado derecho del sistema de ecuaciones lineales cuya solución proporciona la iteración en u.

Proposición 5

a) La matriz $n \times n$

$$U = H^{-1}G(M - G^t H^{-1} G)^{-1} G^t H^{-1} + H^{-1}$$

tiene la forma

$$U = \begin{bmatrix} U_0 & 0 & \cdots & 0 \\ 0 & U_1 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & U_p \end{bmatrix}$$

siendo:

i) U_0 una matriz diagonal $(n_0 \times n_0)$ cuyo j-simo elemento vale $\exp(-\omega_j)$, $j = 1, ..., n_0$.

ii) $U_k = -h_k/(\alpha_k^3\sigma_k^2(h_k + 1)(h_k + 2)) \quad k \in K_1$.

iii) $U_k = \lambda_k(D_k^{-1} - \rho_k e_k e_k^t) \quad k \in K_2 \quad f_k > -1/2\alpha_k$ con

$$\rho_k = \frac{\rho_k'}{\rho_k''} = \frac{6\alpha_k^2 f_k^2 + 6\alpha_k f_k + 2}{4\alpha_k^2 f_k^3 + (6\alpha_k^2 + 3\alpha_k)f_k^2 + (6\alpha_k + 1)f_k + 2}$$

b) El vector

$$d = H^{-1}G(M - G^t H^{-1}G)^{-1}(W - G^t H^{-1}V) - H^{-1}V$$

tiene la forma:

i) $d_j = -1$ si $j \in J(0)$.

ii) $d_j = -h_k/(\alpha_k^2(h_k + 2))$ si $j \in J(k), k \in K_1$.

iii) $d_j = -(2\alpha_k^2 f_k^3 + 3\alpha_k^2 f_k^2 + f_k)/\rho_k''$ si $j \in J(k), k \in K_2; f_k > 1/2\alpha_k$.

DEMOSTRACION

a) Teniendo en cuenta las expresiones de H^{-1}, G y M es sencillo ver que la matriz U tiene la forma diagonal indicada.

i) Puede comprobarse fácilmente que U_0 es la misma submatriz que H_0^{-1} y de ahí su expresión.

ii) Sea $k \in K_1$. Entonces se tiene:

$$H_k^{-1}G_k = 2/\alpha_k^2\sigma_k$$
$$(M_k - G_k^t H_k^{-1}G_k)^{-1} = -\alpha_k/2(h_k + 2)$$
$$H_k^{-1}G_k(M_k - G_k^t H_k^{-1}G_k)^{-1}G_k^t H_k^{-1} = -2/(\alpha_k^3\sigma_k^2(h_k + 2))$$

y finalmente

$$\begin{aligned} U_k &= H_k^{-1}G_k(M_k - G_k^t H_k^{-1}G_k)^{-1}G_k^t H_k^{-1} + H_k^{-1} \\ &= -h_k/(\alpha_k^3\sigma_k^2(h_k + 1)(h_k + 2)) \end{aligned}$$

iii) Sea $k \in K_2$. Entonces

$$H_k^{-1}G_k = \lambda_k(D_k^{-1} - \mu_k e_k e_k^t)2\sigma_k\lambda_k'\theta_k = 2\frac{\lambda_k'}{\sigma_k\mu_k}e_k$$

en donde al efectuar los cálculos hemos tenido en cuenta (13) y (14).

$$(M_k - G_k^t H_k^{-1}G_k)^{-1} = \left(2f_k(\alpha_k f_k + 1) - 2\sigma_k\lambda_k'\theta_k\frac{2\lambda_k'}{\sigma_k\mu_k}e_k\right)^{-1} =$$

$$= -\frac{1}{2}\frac{\mu_k'}{\rho_k''}$$

habiendo usado de nuevo (14).

$$H_k^{-1}G_k(M_k - G_k^t H_k^{-1}G_k)^{-1}G_k^t H_k^{-1} = -\frac{2(\lambda_k')^2}{\sigma_k^2\mu_k'\rho_k''}e_k e_k^t$$

y finalmente

$$U_k = H_k^{-1}G_k^t(M_k - G_k^t H_k^{-1}G_k)^{-1}G_k^t H_k^{-1} + H_k^{-1} =$$

$$= -\frac{2(\lambda_k')^2}{\sigma_k^2\mu_k'\rho_k''}e_k e_k^t + \lambda_k(D_k^{-1} - \mu_k e_k e_k^t) =$$

$$= \lambda_k\left[D_k^{-1} - \left(\mu_k + \frac{2(\lambda_k')^2}{\sigma_k^2\mu_k'\rho_k''\lambda_k}\right)e_k e_k^t\right]$$

Efectuando operaciones se encuentra que

$$\mu_k + \frac{2\lambda_k'^2}{\sigma_k^2\mu_k'\rho_k''\lambda_k} = \rho_k$$

con lo que se obtiene la expresión del enunciado.

b) El vector d puede escribirse como

$$d = H^{-1}G(M - G^t H^{-1}G)^{-1}W - UV$$

i) En virtud de la parte *a*), i) y de las proposiciones 3 y 4 es inmediato comprobar que $d_j = -1, j \in J(0)$.

ii) Sea $k \in K_1$. Entonces teniendo en cuenta *a*) ii) y las proposiciones 3 y 5

$$H_k^{-1}G_k(M_k - G_k^t H_k^{-1} G_k)^{-1} W_k = -\frac{2h_k}{\alpha_k^2(h_k + 2)}$$

$$U_k V_k = -\frac{h_k}{\alpha_k^2(h_k + 2)}$$

Restando ahora las dos últimas expresiones se obtiene el resultado deseado.

iii) Sea $k \in K_2$. Teniendo de nuevo en cuenta *a*) iii) y las proposiciones 3 y 5

$$H_k^{-1}G_k(M_k - G_k^t H_k^{-1} G_k)^{-1} W_k = -\frac{2\lambda_k' f_k(\alpha_k f_k + 1)}{\rho_k''} e_k$$

$$\begin{aligned} U_k V_k &= \lambda_k(D_k^{-1} - \rho_k e_k e_k^t)\sigma_k^2(2\alpha_k f_k + 1)\theta_k = \\ &= e_k - \rho_k(f_k + 1)e_k = (1 - \rho_k(f_k + 1))e_k \end{aligned}$$

en donde hemos utilizado (13) y (14). Restando ahora las dos expresiones anteriores y efectuando los cálculos se llega fácilmente a la expresión de los $d_j\ j \in J(k)$.

Una vez calculados la matriz U y el vector d vamos a simplificar las expresiones que proporcionan las iteraciones en las variables.

Teorema 1. (Iteración en *u*)

La iteración en u consiste en encontrar la solución u' del siguiente sistema lineal de $(n - m)$ ecuaciones con $(n - m)$ incógnitas:

$$(CUC^t)u = Cd \tag{15}$$

DEMOSTRACION

Inmediata a partir de (10).

Teorema 2. (Iteración en σ)

Las fórmulas que dan la iteración en σ son:

$$\sigma_k' = \frac{\sigma_k}{h_k + 2}\left(h_k + 1 - \frac{C_k u'}{\alpha_k \sigma_k^2}\right) \quad k \in K_1 \tag{16}$$

$$\sigma_k' = \frac{\sigma_k(\lambda_k')^2}{\rho_k''}(f_k + 1 - \lambda_k e_k^t C_k u') \quad k \in K_2 \tag{17}$$

siendo $C_k = (C_{m_k}, ..., C_{n_k})$ submatriz de C formada por las columnas correspondientes a los términos de la restricción k.

DEMOSTRACION

De (11) se deduce

$$\sigma' = \sigma + (M - G^t H^{-1} G)^{-1} G^t H^{-1} V + (M - G^t H^{-1} G)^{-1} G^t H^{-1} C^t u' - \\ - (M - G^t H^{-1} G)^{-1} W$$

Es inmediato comprobar que las matrices producto que aparecen en la fórmula anterior presentan una estructura por cajas.

Sea $k \in K_1$. Entonces

$$(M_k - G_k^t H_k^{-1} G_k)^{-1} G_k^t H_k^{-1} V_k = -\frac{h_k + 1}{h_k + 2}\sigma_k \tag{18}$$

$$(M_k - G_k^t H_k^{-1} G_k)^{-1} G_k^t H_k^{-1} C_k^t u' = -\frac{1}{\alpha_k \sigma_k (h_k + 2)} C_k^t u' \tag{19}$$

$$(M_k - G_k^t H_k^{-1} G_k)^{-1} W_k = -\frac{h_k}{h_k + 2}\sigma_k \tag{20}$$

Reuniendo los tres resultados anteriores (18)-(20) se obtiene la expresión de σ_k' buscada.

Sea $k \in K_2$. Entonces

$$(M_k - G_k^t H_k^{-1} G_k)^{-1} G_k^t H_k^{-1} V_k = -\frac{(\lambda_k')^2 (f_k + 1)}{\rho_k''} \sigma_k \tag{21}$$

$$(M_k - G_k^t H_k^{-1} G_k)^{-1} G_k^t H_k^{-1} C_k^t u' = -\frac{\lambda_k'}{\rho_k''} \frac{e_k^t C_k^t u'}{\sigma_k} \tag{22}$$

$$(M_k - G_k^t H_k{}^{-1} G_k)^{-1} W_k = -\frac{\mu_k'}{\rho_k''} f_k(\alpha_k f_k + 1)\sigma_k \tag{23}$$

Reuniendo (21)-(23) se llega tras unos sencillos cálculos a la expresión (17).

Teorema 3. (Iteración en ω)

Las fórmulas que dan la iteración en las variables ω son:

$$\omega_j' = \omega_j - 1 - \exp(\omega_j) C_j^t u' \quad j \in J(0) \tag{24}$$

$$\omega_j' = \omega_j - \frac{h_k}{\alpha_k(h_k + 2)} \left(1 - \frac{C_j^t u'}{\alpha_k \sigma_k (h_k + 1)}\right) \quad j \in J(k) \quad ; \quad k \in K_1 \tag{25}$$

$$\omega_j' = \omega_j - d_j - \lambda_k(\eta_k - \rho_k e_k^t) C_k^t u' \quad j \in J(k) \quad ; \quad k \in K_2 \tag{26}$$

siendo $\eta_k = (0, ..., \exp(-\omega), ..., 0)$ con $n_k - m_k + 1$ componentes.

DEMOSTRACION

De (12)

$$\omega' = \omega - H^{-1}(V + G(\sigma' - \sigma) + C^t u')$$

y al sustituir aquí el valor de $(\sigma' - \sigma)$ dado por (11)

$$\begin{aligned} \omega' &= \omega - H^{-1}[V + G\{(M - G^t H^{-1} G)^{-1}[G^t H^{-1}(V + C^t u') - W]\} + C^t u'] \\ &= \omega - (H^{-1} + H^{-1} G(M - G^t H^{-1} G)^{-1} V + H^{-1} G(M - G^t H^{-1} G)^{-1} W \\ &\quad - [H^{-1} + H^{-1} G(M - G^t H^{-1} G)^{-1} G^t H^{-1}] C^t u' \\ &= \omega + d - U C^t u' \end{aligned} \tag{27}$$

en donde hemos empleado la proposición 5.

Las fórmulas (24)-(26) no son más que la expresión por componentes de la fórmula (27).

Junto con las fórmulas iterativas en las variables primales y duales los algoritmos basados en Lagrangianos aumentados suelen utilizar también algún mecanismo de actualización de los parámetros de penalización. Las reglas que se emplean en este sentido son heurísticas. Inspirados en las condiciones de optimalidad que prescriben la anulación del lagrangiano ordinario del PPTP proponemos las fórmulas que se expresan a continuación porque experimentalmente han manifestado un mejor comportamiento computacional.

Actualización de los multiplicadores

a) Sea $k \in K_1$. Se define

$$\beta_k = \begin{cases} (1/\omega_{m_k}) \log\left(-\dfrac{C_k^t u'}{\sigma_k^2 \alpha_k} \right) & \text{si} \quad C_k^t u' > 0 \\ 0 & \text{en otro caso} \end{cases}$$

y entonces

$$\alpha_k' = \begin{cases} \sqrt{\beta_k} & \text{si} \quad \beta_k > 0 \\ \alpha_k & \text{en otro caso} \end{cases} \tag{28}$$

b) Sea $k \in K_2$. Se define

$$\beta_k = -(1/2f_k)\left(1 + \frac{e_k^t C_k^t u'}{\sigma_k^2 (f_k + 1)}\right)$$

y entonces

$$\alpha_k' = \begin{cases} \beta_k & \text{si} \quad \beta_k > 0 \\ \alpha_k & \text{en otro caso} \end{cases} \tag{29}$$

El algoritmo puede enunciarse en los términos siguientes:

ALGORITMO

A) Datos: $m; p; n; J(k), k = 1, ..., p; A, c$.

B) Paso inicial

— Calcular la matriz C.

— Calcular un punto ω^0 que verifique el sistema de restricciones lineales del PPTP.

— Dar estimaciones iniciales σ^0, α^0.

C) Paso General

— Resolver el sistema (15) en u.

— Iterar σ según las fórmulas (16)-(17).

— Iterar ω según las fórmulas (24)-(26).

— Comprobar las condiciones de convergencia. Si se verifican reconstruir la solución primal usando la expresión $\omega = Az + \log c$ y finalizar.

— En otro caso modificar α según las expresiones (28)-(29).

— Repetir el paso general.

3. EXPERIENCIAS COMPUTACIONALES

El algoritmo descrito en el párrafo anterior se ha probado resolviendo algunos problemas test tomados de Rijckaert y Martens, 1978a. Los cálculos se han realizado con un microordenador HP-87.

El algoritmo comienza calculando una matriz C generadora del espacio de A. Existen varios procedimientos para efectuar esta operación. Aquí simplemente se ha elegido una base B de entre las columnas de A de forma que $A = (B\,R)$, donde denota la submatriz de A formada por las columnas que no están en B y se ha tomado:

$$C = \begin{vmatrix} -B^{-1} & R \\ & I \end{vmatrix}$$

La estructura de C permite encontrar inmediatamente un punto ω inicial cumpliendo las restricciones lineales del PPTP dado por:

$$\omega^B = 0$$
$$\omega^R = C \log c$$

Experimentalmente se ha comprobado que la elección de la matriz B que determina el punto inicial ω tiene un papel influyente en la evolución de las iteraciones. Dado que las variables ω asociadas a términos de restricciones deben tomar en el óptimo un valor menor o igual que cero se ha utilizado la estrategia de elegir como matriz B, una matriz con el mayor número posible de columnas correspondientes a términos de restricciones; esta estrategia ha dado normalmente buenos resultados. La matriz B sirve también para proporcionar los valores de las variables primales t resolviendo el sistema lineal en z

$$(\omega^*)^B = Bz \tag{30}$$

donde $(\omega^*)^B$ denota el subvector de ω^* formado por las componentes cuyos índices corresponden a las columnas de B, y exponenciando a continuación el resultado

$$t^* = \exp(z^*)$$

donde z^* es la solución de (30).

Los valores iniciales de σ se han tomado usualmente iguales a uno y como α inicial se han usado normalmente potencias de diez que ponderen las restricciones en su contribución al objetivo total.

El sistema lineal (15) se ha resuelto usando una subrutina basada en el método de Gauss, pivotando sobre el elemento de mayor valor absoluto.

Como criterios de convergencia se han utilizado:

1) $f_k(\omega^r) \leqslant 1 \cdot E - 05 \quad k = 1, ..., p.$
2) $|(f_0(\omega^r) - f_0(\omega^{r-1}))/f_0(\omega^{r-1})| \leqslant 1 \cdot E - 05.$
3) $\|\omega^r - \omega^{r-1}\| \leqslant 1 \cdot E - 04.$
4) $\|\sigma^r - \sigma^{r-1}\| \leqslant 1 \cdot E - 04.$

en donde ω^r, σ^r denotan la iteración actual, deteniéndose el proceso cuando se satisfacen simultáneamente los cuatro criterios.

Como se ha visto el mayor esfuerzo computacional del algoritmo consiste en el cálculo de la matriz CUC^t y la resolución del sistema (15), ya que las actualizaciones de ω y σ son sencillas. Entonces una buena idea del comportamiento del algoritmo viene dada por el número de iteraciones necesarias para resolver el problema, ya que de él pueden extrapolarse otros criterios tales como tiempo de CPU, número de evaluaciones de funciones, etc. En el apéndice se muestran algunas características de los problemas, junto con el número de iteraciones necesarias para resolverlos.

4. CONCLUSIONES

En este trabajo se materializa un algoritmo para la programación geométrica que responde al esquema general descrito en Ramos 85. El método se analiza por completo justificando las fórmulas iterativas propuestas. Entre las principales ventajas del procedimiento cabe señalar la de presentar un esquema iterativo muy simple, la de permitir obtener estimaciones de segundo orden de los multiplicadores de Kuhn-Tucker a través de las variables σ (Ramos, 1979, 1981) y la de tener una razón de covergencia cuadrática derivada del método de Newton en que se basa.

El algoritmo es especialmente indicado cuando el «grado de dificultad» del problema no es muy alto. Alternativamente queda abierta la posibilidad de investigar la aplicación de las técnicas de programación en gran escala para la actualización de la matriz que proporciona la iteración en u vía factorización o algún esquema análogo.

REFERENCIAS

BEIGHTLER, C. S., y PHILLIPS, D. T. (1976): *Applied geometric Programming,* J. Wiley.

DEMBO, R. S. (1976): «A set of geometric programming test problems and their solutions», *Mathematical Programming,* 10, págs. 192-213.

DEMBO, R. S. (1978): «Current state of the art of algorithms and computer software for geometric programming», *Journal Optimization Theory and Applications,* 26, núm. 2, págs. 149-183.

DEMBO, R. S. (1979): «Second order algorithms for the posynomial geometric programming dual. Part I: Analysis», *Mathematical Programming,* 17, págs. 156-175.

DUFFIN, R. J.; PETERSON, E. L., y ZENER, C. M. (1967): *Geometric Programming,* J. Wiley.

ECKER, J. G.; KUPFERSCHMID, M., y SACHER, R. S. (1984): «Comparison of special purpose algorithm with general purpose algorithm for solving geometric programming codes», *Journal Optimization Theory Applications*, 43, núm. 2, págs. 237-263.

FATTLER, J. E.; SIN, Y. T.; ROOT, R. R.; RADSDELL, K. M., y REKLAITIS, G. V. (1982): «On the computational utility of posynomial geometric programming solution methods», *Mathematical Programming,* 29, págs. 163-201.

GILL, P. E.; MURRAY, W., y WRIGTH, M. H. (1981): *Practical Optimization,* Academic Press.

McCORMICK, G. P. (1983): *Nonlinear programming: theory, algorithms and applications,* J. Wiley.

PETERSON, E. L. (1978): *Geometric programming,* en Moder, J. J., y S. E. Elmaghraby (eds.): «Handbook of operations research, 1», Van Nostrand, págs. 207-244.

RAMOS, E. (1979): *Métodos de las penalizaciones y métodos de los multiplicadores para la programación matemática,* Tesis Doctoral, Univ. Santiago de Compostela.

RAMOS, E. (1981): «Funciones penalidad y lagrangianos aumentados», *Trabajos de Estadística y de Investigación Operativa,* 32, núm. 1, págs. 94-115.

RAMOS, E. (1985): «Métodos de lagrangiano aumentado aplicados a la programación geométrica», *Actas de las X Jornadas Hispano-Lusas de Matemáticas,* Murcia.

RIJCKAERT, M. J., y MARTENS, X. M. (1978a): «Comparison of generalized geometric programming algorithms», *Journal Optimization Theory Applications,* 26, núm. 2, págs. 205-242.

RIJCKAERT, M. J., y MARTENS, X. M. (1978b): «Bibliografical note on geometric programming», *Journal Optimization Theory Applications,* 26, núm. 2, págs. 325-337.

SARMA, P. V. L. N.; MARTENS, X. M.; REKLAITIS, G. V., y RIJCKAERT, M. J. (1978): «A comparison of computational strategies for geometric programming», *Journal Optimization Theory Applications*, 26, núm. 2, págs. 185-203.

SMEERS, Y., y TYTECA, D. (1984): «A geometric programming model for the optimal design of wastewater treatment plants», *Operations Research,* 32, núm. 2, págs. 314-342.

APENDICE

Problema 1

		A'											
		1	2	3	4	B	c	ω^0	ω^*	σ^0	σ^*	α^0	α^*
f_0	1	−1.0	0.0	0.0	0.0		1.0000	−4.6052	−4.4143				
f_1	2	1.0	−1.0	0.0	0.0	*	1.0000	0.0000	−0.0622	1.0000	0.1135	100.00	101.39
	3	0.0	0.0	−1.0	0.0		0.5000	−8.2940	−2.8076				
f_2	4	0.0	0.0	1.0	−1.0	*	0.0100	0.0000	−2.8076	1.0000	0.1135	0.01	0.22
	5	0.0	1.0	0.0	0.0	*	0.0100	0.0000	−0.1286				
	6	0.0	1.0	0.0	1.0	*	0.0005	0.0000	−2.8076				
t^*		82.6222	87.9296	8.2847	1.3727		$f_1^* = 0.0121$						

Problema 2

		A'										
		1	2	3	B	c	ω^0	ω^*	σ^0	σ^*	α^0	α^*
f_0	1	1.0	0.0	0.0		5.	2.9957	6.2983				
	2	1.0	0.0	0.0		50000.	9.4335	6.1309				
	3	0.0	1.0	0.0		20.	6.4615	7.4397				
	4	0.0	−1.0	0.0		72000.	7.7187	6.7403				
	5	0.0	0.0	1.0		10.	7.0901	7.6223				
	6	0.0	0.0	−1.0		144000.	7.0901	6.5579				
f_1	7	−1.0	0.0	0.0	*	4.	0.0000	−3.3026	1.0000	47.7393	$1.0E+6$	$3.3E+9$
	8	0.0	−1.0	0.0	*	32.	0.0000	−0.9784				
	9	0.0	0.0	−1.0	*	120.	0.0000	−0.5322				

t^* 108.7347 85.1262 204.3246 $f_0^* = 6299.8424$

Problema 3

		A^t 1	2	3	4	B	c	ω^0	ω^*	σ^0	σ^*	α^0	α^*
f_0	1	0.65	0.00	0.00	0.00		592	10.4230	10.6865				
	2	0.39	0.00	0.00	0.00		582	8.7902	8.9482				
	3	0.52	0.00	0.00	0.00		1200	10.3217	10.5324				
	4	0.22	−0.22	0.00	0.00		370	6.7741	7.8532				
	5	0.40	0.00	−0.40	0.00		250	7.0863	8.0175				
	6	0.62	0.00	−0.62	0.00		210	7.7726	9.2160				
	7	0.40	0.00	0.00	−0.40		250	7.0863	7.6770				
	8	0.85	0.00	0.00	−0.85		200	8.6235	9.8789				
f_1	9	−1.00	0.00	0.00	0.00	*	500	0.0000	−0.4053	1.0000	−276.4346	1.00	1.56
	10	−1.00	1.00	0.00	0.00	*	50	0.0000	−4.9049				
	11	−1.00	0.00	1.00	0.00	*	50	0.0000	−2.3282				
	12	−1.00	0.00	0.00	1.00	*	50	0.0000	−1.4769				
t^*		749.8949	0.1111	1.4619	3.4248		$f_0^* = 126303.1780$						

Problema 4

		A'												
		1	2	3	4	B	c	ω^0	ω^*	σ^0	σ^*	α^0	α^*
f_0	1	1.0	1.0	0.0	0.0		168.00000	12.6232	12.6886				
	2	1.0	1.0	−1.0	0.0		3651.20000	11.4733	11.5714				
	3	1.0	0.0	0.0	0.0		3651.20000	11.9316	11.9643				
	4	0.0	0.0	0.0	−1.0		40000.00000	7.0910	10.4950				
f_1	5	1.0	−1.0	0.0	0.0	*	1.04250	0.0000	0.0000	1.0000	−655.7930	1.00	1.00
f_2	6	1.0	0.0	1.0	0.0	*	0.00035	0.0000	0.0000	1.0000	−325.6124	1.00	1.00
f_3	7	−1.0	0.0	0.0	1.0	*	1.25000	0.0000	−3.4367	1.0000	−1059.8133	1.00	0.25
	8	−1.0	0.0	0.0	0.0	*	41.63000	0.0000	−0.0327				
t^*		43.0138	44.8418	66.4239	1.1070		$f_0^* = 623249.8170$						

Problema 5

		A^t																
		1	2	3	4	5	6	7	8	B	c	ω^0	ω^*	σ^0	σ^*	α^0	α^*
f_0	1	0.9	−1.5	−3.	0.	0.	0.	0.	0.		2.0	2.3165	2.7434				
	2	0.	0.	0.	−0.3	2.6	0.	0.	0.		5.0	0.7048	1.6881				
	3	0.	0.	0.	0.	0.	−1.8	−0.5	1.		4.7	1.8911	2.1140				
f_1	4	2.3	1.7	4.5	0.	0.	0.	0.	0.	*	10.0	0.0000	0.0000	1.0000	2.1452	1.0	1.0
f_2	5	0.	0.	0.	−2.1	0.4	0.	0.	0.		0.6	−3.3322	0.0000	1.0000	−0.2971	1.0	9.9
f_3	6	0.	0.	0.	0.	0.	4.5	−2.7	−0.6	*	6.2	0.0000	0.0000	1.0000	1.8121	1.0	1.0
f_4	7	1.6	0.4	−3.8	0.	0.	0.	0.	0.	*	3.1	0.0000	0.0000	1.0000	0.9753	1.0	1.0
f_5	8	0.	0.	0.	5.4	1.3	0.	0.	0.	*	3.7	0.0000	0.0000	1.0000	1.4221	1.0	1.0
f_6	9	0.	0.	0.	0.	0.	−1.1	7.3	−5.6	*	0.3	0.0000	0.0000	1.0000	1.3846	1.0	1.0
f_7	10	−3.8	2.2	4.3	0.	0.	0.	0.	0.	*	7.2	0.0000	−0.9660	1.0000	4.2472	1.0	6.3
	11	0.	0.	0.	−0.7	−1.6	0.	0.	0.	*	0.5	0.0000	−0.5268				
	12	0.	0.	0.	0.	0.	4.3	−1.9	8.5	*	0.2	0.0000	−3.5452				
t^*		0.9689	0.1989	1.1213	0.7844	1.0022	0.7010	1.0941	0.9724		$f_0^* = 29.2295$						

Problema 6

		A' 1	2	3	4	5	6	7	8	B	c	ω^0	ω^*	σ^0	σ^*	α^0	α^*
f_0	1	0.9	−1.5	−3.	0.	0.	0.	0.	0.		2.0	2.3165	2.7366				
	2	0.	0.	0.	−0.3	2.6	0.	0.	0.		5.0	0.7048	1.7086				
	3	0.	0.	0.	0.	0.	−1.8	−0.5	1.		4.7	1.8911	2.1127				
f_1	4	2.3	1.7	4.5	0.	0.	0.	0.	0.	*	10.0	0.0000	0.0000	1.0000	2.1379	1.	1.
f_2	5	0.	0.	0.	−2.1	0.4	0.	0.	0.		0.2	−1.4318	−1.0917	1.0000	−0.0000	1.	1.
f_3	6	0.	0.	0.	0.	0.	4.5	−2.7	−0.6	*	6.2	0.0000	0.0000	1.0000	1.8109	1.	1.
f_4	7	1.6	0.4	−3.8	0.	0.	0.	0.	0.	*	3.1	0.0000	0.0000	1.0000	0.9720	1.	1.
f_5	8	0.	0.	0.	5.4	1.3	0.	0.	0.	*	3.7	0.0000	0.0000	1.0000	1.2817	1.	1.
f_6	9	0.	0.	0.	0.	0.	−1.1	7.3	−5.6	*	0.3	0.0000	0.0000	1.0000	1.3837	1.	1.
f_7	10	−3.8	2.2	4.3	0.	0.	0.	0.	0.	*	7.2	0.0000	−0.9507	1.0000	4.2008	1.	12568.
	11	0.	0.	0.	−0.7	−1.6	0.	0.	0.	*	0.5	0.0000	−0.5377				
	12	0.	0.	0.	0.	0.	4.3	−1.9	8.5	*	0.2	0.0000	−3.5246				
t^*		0.9668	0.1998	1.1207	0.7830	1.0100	0.7020	1.0962	0.9745		$f_0^* = 29.2266$						

PROBLEMA	m	n	p	q	iter
1	4	6	2	0	11
2	3	9	1	0	25
3	4	12	1	0	13
4	4	8	3	2	40
5	8	12	7	6	12
6	8	12	7	6	6

m = Número de variables primales.
n = Número de términos.
p = Número de restricciones primales.
q = Número de restricciones primales con un único término.
iter = Número de iteraciones.