PROGRAMACION GENERAL NO-LINEAL CON RESTRICCIONES. ALGORITMOS APLICABLES (I)

L. F. ESCUDERO

En este trabajo se describen algunos de los algoritmos de programación general de uso más generalizado. Estos algoritmos son: MAP (método de programación aproximada), GR (gradiente reducido con condiciones lineales), GRG (gradiente reducido generalizado), GRG2 (gradiente reducido generalizado 2), MINOS (sistema modular de optimización no-lineal con condiciones lineales) y CGAP (programación aproximada en base al gradiente conjugado). Las características comunes más sobresalientes son: (a) no precisan para su convergencia que el problema a optimizar sea convexo aunque si no lo fuese, el óptimo local no garantiza que sea el óptimo global, (b) combinan los métodos de optimización sin restricciones con los métodos de programación lineal, (c) utilizan el gradiente reducido y la matriz hessiana reducida de la función objetivo para determinar la dirección de las variables, y (d) tienen su validación utilizando problemas reales.

Así mismo se describen las diferencias entre estos algoritmos y con respecto a los algoritmos SAL (optimización secuencial de la función de Lagrange Aumentada).

Finalmente, se recoge la tendencia en la investigación futura en este campo.

0. INTRODUCCION

En este trabajo se describen algunos de los métodos de optimización no-lineal con restricciones (PNL) de uso más generalizado. Sus características comunes más sobresalientes son las siguientes: (a) No precisan para su convergencia que el problema a optimizar sea convexo, aunque si no lo fuese el óptimo local obtenido no garantiza que sea el óptimo global, (b) Combinan los métodos de programación sin restricciones (PSR) con los métodos de programación lineal (PL), de tal forma que en las sucesivas etapas de estos métodos se optimizan subproblemas de PL; Wolfe /62/ indica que un problema de PNL implícita ó explícitamente entraña un problema de PSR completado con sucesivas iteraciones que constituyen problemas de PL, (c) Utilizan el gradiente reducido de la función objetivo para determinar la dirección en la que el vector de las variables no-básicas debe moverse en el siguiente paso de optimización, y (d) Tienen una validación utilizando problemas reales.

Escudero /16/ recoge una panorámica general de las técnicas de programación matemática, con una amplia bibliografía sobre los métodos de PSR, PL y PNL, glosando sus principales características. A grosso modo, se puede indicar que los métodos de PNL aquí descritos son variantes del método simplex de PL. Gill y Murray /26/ recogen otros métodos de programación no-lineal con restricciones además de algunos aquí descritos, así como la operativa general de estas técnicas. No obstante, los aquí recogidos son los que hoy día gozan de mayor utilización. Para una exposición clara de los métodos PSR, ver Brodlie /9/.

Wolfe /62/ describe los criterios en base a los cuales se analiza la convergencia de un algoritmo de PNL.

La estructura de este trabajo es la siguiente. La sección 1 recoge el algoritmo MAP (Method of Approximating Programming) que se basa en sucesivas linealizaciones de la función objetivo y condiciones de un problema de PNL. La sección 2 recoge métodos de PNL que, basados en el gradiente reducido, resuelven un problema con función objetivo no-lineal y condiciones lineales. La sección 3 recoge el algoritmo GRG (Generalized Reduced

- L.F. Escudero de IBM-España, Paseo de la Castellana, 4, Madrid - 1.
- Article rebut el març de 1978.

Gradient) que amplia el tipo de problemas no -lineales a tratar incluyendo condiciones no -lineales. La sección 4 recoge el algoritmo GRG2 que basado en el algoritmo GRG efectúa fuertes modificaciones internas fundamenta-- les (a) en la determinación del gradiente reducido, (b) en la determinación de la amplitud del desplazamiento del vector de soluciones, (c) en la forma en que se obtiene el -- vector de variables básicas y, (d) en las -- condiciones bajo las cuales se efectúa el -- cambio de base. La segunda parte de este trabajo (QÜESTIIÓ, número siguiente) comienza - con la sección 5 que recoge el algoritmo MINOS el cual, basado en el algoritmo GRG2, -- trata el problema de PNL con condiciones lineales y matriz de poca densidad. La sección 6 recoge el algoritmo CGAP (Conjugate Gradient Approximation Programming) que trata - problemas de PNL en general, cuya principal diferencia con respecto a los anteriores consiste (a) en utilizar un nuevo método de PSR basado en el gradiente conjugado, (b) en utilizar un nuevo método para la determinación del incremento del vector de cierto tipo de variables no-básicas, utilizando la potencia de un sistema de PL de uso generalizado.

1. ALGORITMO MAP (METHOD OF APPROXIMATING -- PROGRAMMING)

La forma más simple de resolver un problema general de PNL a base de una secuencia de -- iteraciones de PL consiste en aproximar cada función no-lineal del problema por medio de una función lineal y resolver el problema correspondiente. Una objeción inmediata a esta aproximación consiste en que las aproximaciones lineales sólo son válidas localmente (es decir, alrededor del punto al que se han aproximado las funciones no-lineales). Por - tanto, es preciso restringir el desplazamiento del vector de variables alrededor de este punto, de tal forma que la optimización se - efectue a pasos pequeños (de ahí el nombre - de métodos small step).

Frank y Wolfe /24/ describen el primer algoritmo de optimización de una función objetivo no-lineal sujeta a condiciones lineales - utilizando técnicas de PL, linealizando la - función objetivo a base de despreciar el segundo término y superiores de su desarrollo en las series de Taylor. Wolfe /62/ estudia su convergencia. El problema fundamental consiste en que la actualización de la aproximación de la función objetivo no se efectúa en cada cambio de base, sino cuando se obtiene el óptimo del particular problema de PL y se pretende obtener una mayor aproximación. Por ello se puede considerar que son Griffith y Stewart /32/ quienes sugieren el primer método de PNL general. Himmemblau /36/ y Avriel /5/ describen con gran detalle este algoritmo. Sea el problema:

$$\min f(X) \qquad (1.1)$$

sujeto a

$$g_i(X) \geq 0 \quad i=1,2,\ldots,m \qquad (1.2)$$

$$h_\ell(X) = 0 \quad \ell=1,2,\ldots,p \qquad (1.3)$$

$$u_j \geq X_j \geq 0 \quad j=1,2,\ldots,n \qquad (1.4)$$

donde f, g_i, h son funciones continuas diferenciables y u_j es el límite superior de la - variable X_j. Si ésta tiene un límite inferior de cero, la acotación de las variables se puede transformar en (1.4) mediante un -- simple cambio de variables.

Sea $\bar{X}$ una solución factible del modelo (1.1) a (1.4). Zoutendijk /65/, /66/, recoge diversos métodos para obtener soluciones factibles. Expandiendo en series de Taylor las -- funciones no-lineales y despreciando los segundos términos y superiores (es decir, linealizando estas funciones) resulta:

$$f(X) \simeq \bar{f}(X,\bar{X}) = f(\bar{X}) + \nabla f(\bar{X})(X-\bar{X}) \qquad (1.5)$$

$$g_i(X) \simeq \bar{g}_i(X,\bar{X}) = g_i(\bar{X}) + \nabla g_i(\bar{X})(X-\bar{X}) \qquad (1.6)$$

$$i = 1,2,\ldots,m$$

$$h_\ell(X) \simeq \bar{h}_\ell(X,\bar{X}) = h_\ell(\bar{X}) + \nabla h_\ell(\bar{X})(X-\bar{X}) \qquad (1.7)$$

$$\ell = 1,2,\ldots,p$$

donde $\nabla f(\bar{X}) = (\delta f/\delta X_1, \delta f/\delta X_2, \ldots, \delta f/\delta X_n)$, -- $\nabla g_i(\bar{X})$ y $\nabla h_\ell(\bar{X})$ son los gradientes respectivos de las funciones no-lineales f, g_i y h_ℓ evaluados en el vector $\bar{X}$.

Definiendo

$$Y = X - \bar{X} \qquad (1.8)$$

y considerando que $-\infty \leq Y \leq \infty$, se puede sustituir este vector por

$$Y_j = Y_j^+ - Y_j^- \qquad j = 1,2,\ldots,n \qquad (1.9)$$

donde

$$Y_j^+ \geq 0, \; Y_j^- \geq 0 \qquad j = 1,2,\ldots,n \qquad (1.10)$$

y

$$Y_j^+ = \begin{cases} Y_j \text{ si } Y_j \geq 0 \\ 0 \text{ si } Y_j < 0 \end{cases} \quad (1.11)$$

$$Y_j^- = \begin{cases} -Y_j \text{ si } Y_j \leq 0 \\ 0 \text{ si } Y_j > 0 \end{cases} \quad (1.12)$$

El valor $|Y_j|$ representa la distancia de X_j al valor $\bar{X}_j$. Para evitar inexactitudes excesivas en la aproximación lineal, es preciso limitar la distancia $|Y_j|$ tal que

$$|Y_j| \leq m_j \qquad j = 1,2,\ldots,n \quad (1.13)$$

donde m_j es una constante, de cuyo valor depende que el método sea lento (si es muy pequeño) ó tarde en converger produciendo gran inexactitud (si es grande). Dado que el vector X está limitado por (1.4), utilizando -- (1.4) y (1.8), la inegualdad (1.13) se con-- vierte en:

$$0 \leq Y_j^+ \leq \min\{m_j, u_j, -\bar{X}_j\} \quad (1.14)$$

$$0 \leq Y_j^- \leq \min\{m_j, \bar{X}_j\} \quad (1.15)$$

de donde resulta que la aproximación lineal al problema (1.1) a (1.4) será:

$$\min \bar{f}(X,\bar{X}) = f(\bar{X}) + \nabla f(\bar{X}) Y^+ - \nabla f(\bar{X}) Y^- \quad (1.16)$$

sujeto a

$$\bar{g}_i(X,\bar{X}) = g_i(\bar{X}) + \nabla g_i(\bar{X}) Y^+ - \nabla g_i(\bar{X}) Y^- \geq 0 \quad (1.17)$$
$$i = 1,2,\ldots,m$$

$$\bar{h}_\ell(X,\bar{X}) = h_\ell(\bar{X}) + \nabla h_\ell(\bar{X}) Y^+ - \nabla h_\ell(\bar{X}) Y^- = 0 \quad (1.18)$$
$$\ell = 1,2,\ldots,p$$

y también sujeto a los límites (1.14) y -- (1.15).

El algoritmo MAP en esencia consiste en una secuencia iterativa que partiendo del punto $\bar{X}$ resuelve los problemas subsiguientes de PL definidos por (1.14) a (1.18). Si la solu-- ción óptima X_L^* del problema lineal es factible en el problema no-lineal (1.1) a (1.4), sirve de punto de partida $\bar{X}$ (para el cual -- hay que recalcular $\nabla f(\bar{X})$, $g_i(\bar{X})$, y $h_\ell(\bar{X})$, etc. para el siguiente problema lineal, utilizando los mismos límites (1.14) y (1.15). Si - X_L^* es una solución imposible del problema - no-lineal (es decir, si no cumple las condiciones (1.2) y (1.3), es preciso repetir el problema lineal resuelto pero reduciendo los valores m_j. Se considera que X_L^* es óptimo en el problema (1.1) a (1.4) cuando cumple las condiciones de éste, al menos con un cierto grado de tolerancia y además cumple ciertas condiciones de convergencia (Wolfe, /61/) en los sucesivos valores de X y f(X). Estas con-- diciones serían:

$$g_i(X_L) \geq -\varepsilon \qquad i=1,2,\ldots,m \quad (1.19)$$

$$|h_\ell(X_L^*)| = \delta \qquad \ell=1,2,\ldots,p \quad (1.20)$$

$$||X_L^{(i+1)^*} - X_L^{(i)^*}|| \leq \gamma \quad (1.21)$$

$$||f^{(i+1)}(X_L^*) - f^{(i)}(X_L^*)|| \leq \sigma \quad (1.22)$$

donde $X_L^{(i)^*}$ y $f^{(i)}(X^{(i)^*})$ recogen, respectivamente, los valores de la solución óptima - lineal y de la función objetivo no-lineal en la iteración (optimización del problema li-- neal) i. $\varepsilon, \gamma, \delta$ y σ son parámetros de convergencia.

Ya se ha indicado que los límites m_j tienen una gran influencia en el tiempo de CPU en - la optimización. Si los límites son muy pe-- queños, $|Y_j|$ también lo será y por tanto la convergencia es lenta. Por otro lado, valo-- res altos de m_j pueden producir soluciones - imposibles en las condiciones (1.2) y (1.3). En este caso se reduce m_j y se efectúa de -- nuevo la optimización lineal que produjo solución imposible en las condiciones (1.2) y (1.3).

Dado que en cada nueva optimización, la va-- riable Y_j no tiene gran relación con los valores de Y_j en la optimización anterior y -- además las funciones $\bar{g}_i(X,\bar{X})$ (1.17) y $\bar{h}_\ell(X,\bar{X})$ (1.18) son completamente distintas, -- hay que resolver completamente el problema - de PL, sin poder partir de la solución anterior.

Aunque no se ha probado la convergencia del algoritmo MAP, el método normalmente converge a un óptimo local. Se ha utilizado el algoritmo MAP principalmente en problemas de - optimización de PL, pero que le convierten - en no-lineal porque la función objetivo es - no-lineal ó porque hay algunas condiciones - no-lineales. No obstante, debido a la difi-- cultad en la elección de valores adecuados -

para m_j, la convergencia es lenta y su utilización ha decrecido. Su utilidad estriba en que ha sido el primer algoritmo de PNL general, en cuya filosofía se han basado los demás algoritmos de amplia utilización.

2. ALGORITMOS DE PNL CON CONDICIONES LINEALES. GRADIENTE REDUCIDO

Los métodos recogidos en esta sección combinan métodos PSR (programación no-lineal sin restricciones) con la operativa del método simplex de PL (Dantzig, /13/). Sólo contemplan el caso de función objetivo no-lineal y condiciones lineales. Tienen una gran aplicación dado el amplio campo de la PL (en la cual frecuentemente la función objetivo es no-lineal). Así mismo son muy útiles en la resolución del problema dual de programación geométrica. Escudero /16/ recoge una amplia bibliografía a este respecto.

Estos algoritmos se basan en el gradiente reducido de la función objetivo. Sea el problema:

$$\min f(X) \qquad (2.1)$$

sujeto a:

$$AX = b \qquad (2.2)$$
$$X \geq 0 \qquad (2.3)$$

donde f es una función contínua diferenciable, A es una matriz m*n, b es un vector de orden m y $m \leq n$. Al igual que en PL se puede descomponer el vector X de variables en dos subvectores tal que $X = (X^B, X^N)$, donde $X^B = (x_1^B, x_2^B, \ldots, x_m^B)^t$ es el vector de variables básicas ó dependientes y $X^N = (x_1^N, x_2^N, \ldots, x_{n-m}^N)^t$ es el vector de variables no-básicas ó independientes. Así mismo, la matriz A se descompone en $A = (B,C)$. Se supone que las primeras m columnas de A corresponden a las variables básicas, siendo B esta matriz m*m (que además es no-singular). Por tanto, al igual que en el método simplex de PL:

$$BX^B + CX^N = b \qquad (2.4)$$
$$X^B = B^{-1}b - B^{-1}CX^N \qquad (2.5)$$

Se supone además que las variables básicas son factibles y no-degeneradas. Por tanto, $X^B > 0$. Las variables no-básicas se llaman también independientes ya que asignándoles valores resulta una solución única en el sistema de ecuaciones (2.4).

La idea central de los métodos del gradiente reducido consiste en eliminar las variables básicas X^B, sustituyendo su formulación (2.5) en la función objetivo (2.1) y efectuar una optimización (PSR) de dicha función a base de las variables X^N. Los métodos del gradiente reducido más utilizados, y cuyas diferencias sólo se basan en el código correspondiente, son los algoritmos constrained derivatives debido a Wilde y Beightler(/58/), reduced gradient de Wolfe (/60, /61/) y convex-simplex de Zangwill (/64/).

De forma análoga al método simplex de PL, el gradiente reducido será:

$$\frac{d f(X)}{d X^N} = \nabla_{X^N} f(X) - \nabla_{X^B} f(X)\,(B^{-1}C) \qquad (2.6)$$

donde $\nabla_{X^N} f(X) = (\delta f/\delta X_1^N, \delta f/\delta X_2^N, \ldots, \delta f/\delta X_{n-m}^N)$ y $\nabla_{X^B} f(X) = (\delta f/\delta X_1^B, \delta f/\delta X_2^B, \ldots, \delta f/\delta X_m^B)$ son el gradiente respectivamente de las variables no-básicas y básicas. En programación lineal serían, respectivamente, P^N y P^B siendo $P_1^N, P_2^N, \ldots, P_{n-m}^N$ y $P_1^B, P_2^B, \ldots, P_m^B$ los respectivos coeficientes de la función objetivo.

Si se efectúa un pequeño desplazamiento en el valor del vector X^N en la dirección del negativo del gradiente reducido (2.6) y no se viola la condición de no-negatividad (2.3) de las variables X, se puede obtener una disminución en la función objetivo. La operativa sería la siguiente:

Dado un vector factible X^k en la iteración k se obtiene la dirección del desplazamiento de las variables X_j^N $(j=1,2,\ldots,n-m)$:

$$z_j^{N,k+1} = \begin{cases} 0 \text{ si } x_j^{N,k}=0 \text{ y } d f(X)/d x_j^{N,k} > 0 \\ -d f(X)/d x_j^{N,k} \text{ en caso contrario} \end{cases} \qquad (2.7)$$

Y según (2.5), será:

$$z^{B,k+1} = -B^{-1}C\, z^{N,k+1} \qquad (2.8)$$

Por tanto, la dirección del desplazamiento será $z^{k+1} = (z^{B,k+1}, z^{N,k+1})$. Los vectores z^{k+1}, $z^{B,k+1}$, $z^{N,k+1}$, x^{k+1}, $x^{B,k+1}$ y $x^{N,k+1}$

son vectores columna. El siguiente punto

$$X^{k+1} = (X^{B,k+1}, X^{N,k+1}) \text{ será}$$
$$X^{k+1} = X^k + \alpha^*_{k+1} Z^{k+1} \qquad (2.9)$$

donde la amplitud α^*_{k+1} del desplazamiento será aquella tal que (a) el vector X^{k+1} (2.9) cumpla la condición de no-negatividad (2.3) de las variables y (b) minimice la función - objetivo (2.1) que en este caso ya será un - problema de PSR con un solo argumento (α^*_{k+1}). A diferencia del algoritmo MAP, se podría -- considerar que estos algoritmos son del tipo large step dado que $X^{k+1}-X^k=\alpha^*_{k+1} Z^{k+1}$ no está sujeto a ninguna limitación exogena en el desplazamiento de las variables. Por tanto, α^*_{k+1} vendrá dado por:

$$\alpha^1_{k+1}=\max\{\alpha_{k+1}: X^{B,k}+\alpha_{k+1} Z^{B,k+1} \geq 0\} \qquad (2.10)$$

$$\alpha^2_{k+1}=\max\{\alpha_{k+1}: X^{N,k}+\alpha_{k+1} Z^{N,k+1} \geq 0\} \qquad (2.11)$$

y

$$f(X^k+\alpha^*_{k+1} Z^{k+1}) = \min_{(\alpha_{k+1})}\{f(X^k+\alpha_{k+1} Z^{k+1}):$$
$$0 \leq \alpha_{k+1} \leq \min(\alpha^1_{k+1}, \alpha^2_{k+1})\} \qquad (2.12)$$

Escudero /16/ recoge amplia bibliografia sobre la optimización de una función de un sólo argumento. Los métodos más conocidos son el de Fibonacci y el denominado círculo de - oro (golden circle). Ver Wilde /57/. Si

$\alpha^*_{k+1} < \alpha^1_{k+1}$, X^{k+1} será (2.9). En caso contrario, $X^{B,k}_i + \alpha^1_{k+1} Z^{B,k+1}_i = 0$ (2.13)

para algún valor de i (i=1,2,...,m). En este caso, la variable X^B_i sale de la base, siendo la variable entrante X^N_j (j = 1,2,...,n-m) -- aquella entre las no-básicas que tenga el mayor valor positivo; por tanto el pivotaje se efectúa en (i,j). El algoritmo termina cuando $||Z^{k+1}|| \leq \varepsilon$, donde $\varepsilon \geq 0$ es un valor dado. Si este no es el caso, se recalculan de nuevo los gradientes $\nabla_{X^B} f(X)$ y $\nabla_{X^N} f(X)$ para los vectores básico X^B y no-básico X^N resultantes en la iteración k+1, así mismo el gradiente reducido d f(X)/d X^N (2.6) y se comienza de nuevo la operativa del algoritmo.

El método del gradiente reducido tal como se han descrito sus pasos esenciales, puede no converger a un punto en el que se cumplan -- las condiciones de optimalidad de Kuhn-Tucker (mínimo local) (d f(X)/d $X^N_j \geq 0$ si $X^N_j = 0$ y d f(X)/d $X^N_j = 0$ si $X^N_j > 0$) y de no-negatividad de las variables ($X \geq 0$) (2.3) debido al peligro de la convergencia en zig-zag del -- punto X obtenido en la serie de iteraciones. Zoutendijk (1970, 1976) sugiere varias técnicas que evitan este inconveniente. La más -- sencilla consiste en modificar la obtención de la dirección $Z^{N,k+1}_j$ (2.7) en el sentido - de hacerla cero si $X^{N,k}_j \leq \varepsilon$ y el gradiente - reducido es d f(X)/d $X^{N,k}_j > 0$, donde $\varepsilon > 0$ - es un valor dado.

El método del gradiente reducido ha sido la base para el desarrollo de los métodos que - se recogen a continuación, tal que generalizan su aplicación al caso de condiciones no-lineales ó incrementan fuertemente su convergencia en el caso de matrices de condiciones con poca densidad. De todos modos, los métodos que se recogen a continuación añaden nuevas características que les hacen tener una mayor aplicación dado que incrementan su convergencia.

3. ALGORITMO GRG (GENERALIZED REDUCED GRADIENT

El algoritmo GRG básicamente es una extensión del método del gradiente reducido de -- Wolfe /60/, /61/ al caso de condiciones no-lineales. Es debido a Abadie y Carpentier -- /2/, /3/, cuya versión se denomina GRG66. -- Abadie /1/ describe alguna de sus últimas -- aplicaciones y Abadie y Guigou /4/ describen las diferencias de la versión GRG66 con respecto a la versión GRG69, efectuando un análisis comparativo con los principales algoritmos de PNL. Este estudio confirma las conclusiones obtenidas en el ya clásico trabajo de Colville /12/, en el sentido de que el algoritmo GRG66 ocupa el primer lugar (de los treinta analizados) respecto a velocidad y - precisión. Abadie y Guigou /4/ indican que - el tiempo de CPU tiene una relación de 1 a - 3.26, respectivamente, en los algoritmos -- GRG69 y GRG66. Con posterioridad se han producido numerosas versiones (ver Drud /15/ y otros), siendo uno de los algoritmos con mayor utilización en problemas reales. Las dimensiones típicas para las que el tiempo de CPU es todavía aceptable son m = 150 condiciones y n = 250 variables. Para el caso de

condiciones lineales, se han tratado problemas con m = 300 y n = 1000. Faure y Huard --/17/ desarrollan un código que implementa el algoritmo GRG para condiciones lineales.

Las diferencias fundamentales del algoritmo GRG sobre el método del gradiente reducido - de Wolfe son las siguientes:

a) Obtención del gradiente reducido; evitando efectuar demasiados cambios de base.

b) Se obtienen los valores de las variables dependientes a base de resolver un sistema - de ecuaciones no-lineales con el método denominado seudo-Newton.

Sea al problema

$$\min f(X) \tag{3.1}$$

sujeto a

$$h_i(X)=0 \quad i=1,2,\ldots,m \tag{3.2}$$

$$\ell \leq X \leq u \tag{3.3}$$

donde f y h_i son funciones continuas diferenciables para $X \in R^n$ y $m \leq n$.

Al igual que en casos anteriores se descompone el vector X en el vector X^B de variables básicas y X^N de variables no-básicas ó independientes, siendo $X^B \in R^m$ y $X^N \in R^{n-m}$. Se - supone que la solución básica factible es no -degenerada ($X^B > 0$). $X = (X^B, X^N)$. De forma análoga $\ell = (\ell^B, \ell^N)$ y $u = (u^B, u^N)$ tal que

$$\ell^B \leq X^B \leq \mu^B \tag{3.4}$$

siendo linealmente independientes, los vectores

$$\nabla_{X^B} h_i(X) = (\delta h_i/\delta X_1^B, \delta h_i/\delta X_2^B, \ldots, \delta h_i/\delta X_m^B) \tag{3.5}$$

para i = 1,2,...,m. Por tanto, la matriz m*m $\Delta^B(X)$, cuyas filas son los vectores $\nabla_{X^B} h_i(X)$, es no-singular para los valores del vector - X^B que se hayan obtenido en la iteración correspondiente. Las condiciones de optimalidad de Kuhn-Tucker (ver Wilde y Beightler -- /58/, Zangwill /64/, Mangasarian /43/, Avriel /5/ y Lootsma /42/ entre otros) del problema (3.1) a (3.3) son

$$\nabla_{X^N} f(X) - \mu\Delta^N(X) - \lambda = 0 \tag{3.6}$$

$$\nabla_{X^B} f(X) - \mu\Delta^B(X) = 0 \tag{3.7}$$

donde $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_j, \ldots, \lambda_{n-m})$ es el gradiente reducido, $\Delta^N(X)$ la matriz m*(n-m) cuyas filas son los vectores $\nabla_{X^N} h_i(X)$ (de or-- den n-m), y el vector $\mu = (\mu_1, \mu_2, \ldots, \mu_i, \ldots, \mu_m)$ recoge los coeficientes multiplicadores de Lagrange, siendo éstos, de forma análoga a los correspondientes en PL,

$$\mu = \nabla_{X^B} f(X) \; (\Delta^B(X))^{-1} \tag{3.8}$$

y

$$\lambda = \nabla_{X^N} f(X) - \nabla_{X^B} f(X) (\Delta^B(X))^{-1} \Delta^N(X) \tag{3.9}$$

El valor del gradiente reducido λ (3.9) debe ser tal que:

$$\lambda_j \geq 0 \text{ si } X_j^N = \ell_j^N \tag{3.10}$$

$$\lambda_j = 0 \text{ si } \ell_j^N < X_j^N < u_j^N \tag{3.11}$$

$$\lambda_j \leq 0 \text{ si } X_j^N = u_j^N \tag{3.12}$$

Se puede observar que el gradiente reducido (3.9) de las variables no-básicas ó inde-- pendientes X^N es análogo al recogido en (3.6) para el caso de condiciones lineales.

3.1 Dirección del desplazamiento. Vector $Z^N = (Z_1^N, Z_2^N, \ldots, Z_j^N, \ldots Z_{n-m}^N)^t$.

El algoritmo GRG se basa en una secuencia de iteraciones; a partir de una solución factible X^o se van efectuando una serie de desplazamientos X^1, X^2, etc., tal que en la iteración ℓ, el vector X^ℓ es factible en el modelo (3.1) a (3.3) y el vector λ (3.9) cumple las condiciones de Kuhn-Tucker (3.10) a (3.12) En caso de que no se cumplan estas condiciones, e.g. en la iteración k, se modifica el vector no-básico X^N y se sustituye por

$$X^{N,k+1} = X^{N,k} + \alpha_{k+1}^* Z^{N,k+1}, \quad \alpha_{k+1}^* \geq 0 \tag{3.13}$$

de tal forma que la dirección $Z^{N,k+1}$ del desplazamiento de $X^{N,k}$ a $X^{N,k+1}$ será:

$$Z_j^{N,k+1} = \begin{cases} 0 \mid \lambda_j^k = 0 \\ 0 \mid X_j^{N,k} = \ell_j^N, \; \lambda_j^k > 0 \\ 0 \mid X_j^{N,k} = u_j^N, \; \lambda_j^k < 0 \\ -\lambda_j^k \text{ ó } 0 \text{ otros casos} \begin{cases} \ldots \\ X_j^{N,k} = \ell_j^N, \; \lambda_j^k < 0 \\ \ldots \begin{cases} X_j^{N,k} = u_j^N, \; \lambda_j^k > 0 \\ \ell_j^N < X_j^{N,k} < u_j^N, \; \lambda^k \neq 0 \end{cases} \end{cases} \end{cases} \tag{3.14}$$

según las diversas versiones utilizadas en el algoritmo GRG:

a) En la versión GRG, $z_j^{N,k+1}=-\lambda_j^k$. En el caso de condiciones lineales, el valor de $z_j^{N,k+1}$ es idéntico al recogido en (2.7) para el algoritmo del gradiente reducido de Wolfe /60/ /61/.

b) En la versión GRGS, $z_j^{N,k+1}$ será:

$$z_j^{N,k+1}=\begin{cases} -\lambda_s^k & j=s \\ 0 & j\neq s \quad j=1,2,\ldots,n-m \end{cases} \tag{3.15}$$

donde el índice s es tal que:

$$|\lambda_s^k|=\max_{(j)}\{|\lambda_j^k|\} \tag{3.16}$$

Este criterio en el caso de función objetivo y condiciones lineales convertiría el algoritmo en el método simplex.

c) En la versión GRGC, $z_j^{N,k+1}$ se obtendrá de una forma cíclica tal que para la iteración $k = 1,2,\ldots,n-m$:

$$z_j^{N,k+1}=\begin{cases} -\lambda_j^k & j=k \\ 0 & j\neq k \end{cases} \tag{3.17}$$

Cuando k = n+1, se sustituye por k = 1 y comienza el ciclo en (3.17).

La obtención de la dirección Z^N tal como se ha recogido arriba es la forma clásica del algoritmo GRG. No obstante, también admite la posibilidad de obtener la dirección Z^N a base del método del gradiente conjugado de Fletcher y Reeves /22/, etc. Es decir, en principio permite la utilización de los métodos de programación no-lineal sin restricciones (PSR). Escudero /16/ recoge una bibliografía muy amplia de métodos PSR. En el caso del gradiente (en este caso, reducido), la dirección Z^N sería:

$$z_j^{N,k+1}=\begin{cases} 0 \mid \lambda_j^k = 0 \\ 0 \mid x_j^{N,k} = \ell_j^N,\ \lambda_j^k > 0 \\ 0 \mid x_j^{N,k} = u_j^N,\ \lambda_j^k < 0 \end{cases} \tag{3.18}$$

$$r_j^{N,k}\begin{cases} x_j^{N,k}=\ell_j^N, \lambda_j^k < 0 \\ x_j^{N,k}=u_j^N, \lambda_j^k > 0 \\ \ell_j^N < x_j^{N,k} < u_j^N,\ \lambda^k\neq 0 \end{cases}$$

donde el gradiente reducido conjugado $r^{N,k}$ para el caso de un problema de minimización será (Gill y Murray, /26/, p. 134):

$$r^{N,k} = -\lambda^{N,k} + \beta^k r^{N,k-1} \tag{3.19}$$

donde el coeficiente β^k para el caso del método de Fletcher y Reeves /22/ será:

$$\beta^k=\frac{||\lambda^{N,k}||^2}{||\lambda^{N,k-1}||^2} \tag{3.20}$$

de tal forma que solo se utilizan del vector $r^{N,k}$ aquellos valores para los que $z_j^{N,k+1} = r_j^{N,k}$ (3.18). Para la iteración $k = 1, \beta^k = 0$. Así mismo se reinicializa el procedimiento ($\beta^k = 0$) para $k=(n-m)+1, 2(n-m)+1$, etc. y cuando hay un cambio de base y, por tanto, la composición del gradiente reducido conjugado es diferente.

De forma análoga a (3.13), el nuevo vector $x^{B,k+1}$ de las variables básicas será:

$$x^{B,k+1}=x^{B,k}+\alpha_{k+1}^* z^{B,k+1} \tag{3.21}$$

Debido a la linealización de las condiciones $h_i(X)$ (desarrollándolas en series de Taylor y despreciando el segundo término y superiores), resulta que:

$$H(x^{B,k+1},x^{N,k+1})=H(x^{B,k},x^{N,k})+\Delta^B(X)(x^{B,k+1}-x^{B,k})+\Delta^N(X)(x^{N,k+1}-x^{N,k}) \tag{3.22}$$

donde $H(x^{B,k},x^{N,k})$ y $H(x^{B,k+1},x^{N,k+1})$ recogen el vector de los valores de $h_i(X)$ para las iteraciones k y k+1. Por definición deben ser cero. $\Delta^B(X)$ y $\Delta^N(X)$ están evaluados para $x^B = x^{B,k}$ y $x^N = x^{N,k}$. Por tanto,

$$x^{B,k+1}=\Delta^B(X)^{-1}(\Delta^B(X)x^{B,k}+\Delta^N(X)x^{N,k})-\Delta^B(X)^{-1}\Delta^N(X)x^{N,k+1} \tag{3.23}$$

de donde:

$$x^{B,k+1}=x^{B,k}-\Delta^B(X)^{-1}\Delta^N(X)(x^{N,k+1}-x^{N,k})$$

$$=x^{B,k}+\alpha_{k+1}^*(-\Delta^B(X)^{-1}\Delta^N(X)z^{N,k+1}) \tag{3.24}$$

Por tanto, combinando (3.21) y (3.24) resulta

$$z^{B,k+1}=-\Delta^B(X)^{-1}\Delta^N(X)z^{N,k+1} \tag{3.25}$$

3.2 Amplitud α^*_{k+1} del desplazamiento.

Una vez obtenida la dirección $Z^{k+1}=(Z^{B,k+1}, Z^{N,k+1})$ del desplazamiento (3.25 y 3.14) del vector X^k a X^{k+1}, es preciso determinar la amplitud α^*_{k+1} de este desplazamiento de tal forma que en base a (3.13) y (3.21) se obtenga el nuevo vector X^{k+1}. Existen numerosas versiones para su obtención, pero en principio dado que $\alpha^*_{k+1} \geq 0$, vendrá dado por:

$$\alpha^1_{k+1}=\min\{\min_{(i)} \{ \frac{X^{B,k}_i-\ell^B_i}{-Z^{B,k+1}_i} | Z^{B,k+1}_i<0\},$$

$$\min\{\frac{u^B_i-X^{B,k}_i}{Z^{B,k+1}_i} | Z^{B,k+1}_i>0\}\} \qquad (3.26)$$

para i = 1,2,...,m

$$\alpha^2_{k+1}=\min\{\min_{(j)}\{\frac{X^{N,k}_j-\ell^N_j}{-Z^{N,k+1}_j} | Z^{N,k+1}_j<0\},$$

$$\min\{\frac{u^N_j-X^{N,k}_j}{Z^{N,k+1}_j} | Z^{N,k+1}_j>0\} \} \qquad (3.27)$$

para j = 1,2,...,m

$$f(X^k+\alpha^*_{k+1}Z^{k+1})= \min_{(\alpha_{k+1})} \{f(X^k+\alpha_{k+1}Z^{k+1}):$$

$$0\leq\alpha_{k+1} \leq\min\{\alpha^1_{k+1},\alpha^2_{k+1}\}\} \qquad (3.28)$$

por alguno de los métodos de PSR unidimensionales. Los códigos GRG66 y GRG69 utilizan (ver la serie de trabajos de Abadie, Carpentier y Guigou referenciados más arriba y en especial Abadie /1/) la técnica siguiente:

(a) Elección de un primer valor α_{k+1} factible ($0\leq\alpha_{k+1}\leq\min\{\alpha^1_{k+1},\alpha^2_{k+1}\}$).

(b) Si $H(X)\neq0$ y se desvía fuertemente (ver siguientes párrafos) se sustituye α_{k+1} por $\alpha_{k+1}/2$.

(c) Si la función objetivo se incrementa (ver siguientes párrafos) también se sustituye α_{k+1} por $\alpha_{k+1}/2$.

(d) Si el algoritmo converge en las iteraciones seudo-Newton ($H(X)=0$) y se disminuye la función objetivo, se incrementa α_{k+1} tal que queda sustituido por $2\alpha_{k+1}$ y se continua el procedimiento al igual que con el primer valor α_{k+1} factible.

3.3 Cumplimiento de las condiciones del problema: $H(X) = 0$

Es posible que las condiciones $h_i(X) = 0$ -- $i=1,2,...,m$ (3.2) no se cumplan para el punto X^{k+1} debido a que en su obtención se ha efectuado una linealización (3.22) de las mismas. En este caso, permaneciendo fijo el vector no-básico $X^{N,k+1}$, hay que modificar el vector básico $X^{B,k+1}$ hasta que (3.2) se cumpla. Hay numerosas formas de hacerlo. La más utilizada es el procedimiento denominado seudo-Newton que basado en la formulación (3.2) obtiene iterativamente el valor modificado $X^{B,k+1}$. Por tanto, se supone que en el punto $(X^{B,k+1},X^{N,k+1})$, $H(X^{B,k+1},X^{N,k+1}) \neq 0$. Si se linealizan estas condiciones manteniendo fijo $X^{N,k+1}$ resulta:

$$H(X^{B,k+1},X^{N,k+1})+\Delta^B(X)(\bar{X}^{B,k+1}-X^{B,k+1}) \qquad (3.29)$$

donde $\bar{X}^{B,k+1}$ es el vector columna de los variables básicas con cuyo valor la formulación (3.29) debería ser 0. Considerando que $X^{B,k+1}$ se ha obtenido en la anterior $(\ell-1)$ iteración del método seudo-Newton, se puede sustituir esta notación por $X^{B,\ell-1}$ (y para $\ell=1$, $X^{B,0}=X^{B,k+1}$). De forma análoga se puede sustituir $\bar{X}^{B,k+1}$ por $X^{B,\ell}$. Por tanto,

$$X^{B,\ell}=X^{B,\ell-1}-\Delta^B(X)^{-1}H(X^{B,\ell-1},X^{N,k+1}) \qquad (3.30)$$

Resolviendo el sistema de ecuaciones lineales (3.30), se sustituye $X^{B,\ell}$ en la formulación (3.2) y si $H(X)\neq0$, se efectúa una nueva $\ell=\ell+1$ iteración en el método seudo-Newton, aplicando de nuevo la formulación (3.30), donde $H(X^{B,\ell-1},X^{N,k+1})$ es el vector para el cual $H(X)\neq0$.

Hemos denominado seudo-Newton al método recogido en el algoritmo GRG para obtener el punto X^{k+1} tal que $H(X)=0$, ya que el método Newton recalcula la matriz $\Delta^B(X)$ para cada nuevo punto $(X^{B,\ell},X^{N,k+1})$ obtenido en la utilización iterativa de la formulación (3.30). Se seguirá utilizando esta formulación hasta que ocurra una de las siguientes situaciones:

a) En sucesivas iteraciones se vaya incrementando la norma $||H(X^{B,\ell},X^{N,k+1})||$. Para evitar que se siga desviando el vector de soluciones, se reduce α_{k+1} tal que se satisfaga

$$0 \leq \alpha_{k+1} \leq \min\{\alpha^1_{k+1}, \alpha^2_{k+1}\} \qquad (3.31)$$

y se obtiene de nuevo (3.13) y (3.21); si -- $H(X) \neq 0$ se aplica nuevamente el método seudo -Newton con la formulación (3.30).

b) El valor de la función objetivo sea tal - que

$$f(X^{B,\ell}, X^{N,k+1}) > f(X^{B,\ell-1}, X^{N,k+1}) \qquad (3.32)$$

Se actúa igual que en el caso anterior.

c) En alguna iteración ℓ, el punto $X^{B,\ell}$ cae fuera de los límites definidos en (3.3). En este caso es preciso encontrar un punto $\bar{X}^{B,\ell}_r$ en el segmento que une los puntos $X^{B,\ell}$ y -- $X^{B,\ell-1}$ tal que $\bar{X}^{B,\ell}_r = \ell^B_r$ ó $\bar{X}^{B,\ell}_r = u^B_r$ para algún r (uno ó varios) y $\ell^B_s < \bar{X}^{B,\ell}_s < u^B_s$ para s = 1,2 ...,m, $s \neq r$. Es decir en este caso es preciso efectuar un cambio de base de forma análoga a como se opera en el método simplex -- (Dantzig, /13/); pero en este caso las matrices $\Delta^B(X)$ y $\Delta^N(X)$ y los vectores $\nabla_{X^B} f(X)$ y - $\nabla_{X^N} f(X)$ se evaluan con el vector $(X^{B,\ell}, X^{N,k+1})$. A continuación se efectúan los pivotajes necesarios para obtener una solución - factible; y con estos valores de las matri-- ces se inicia nuevamente el procedimiento en el algoritmo GRG.

d) Algún $X^{B,\ell}_r = \ell^B_r$ ó $X^{B,\ell}_r = u^B_r$. Se actúa -- igual que en el caso anterior. Los algorit-- mos GRG2 y MINOS que se recogen en las si-- guientes secciones no efectúan el cambio de base aunque se den las condiciones recogidas en c) y d) hasta que el procedimiento converja (caso e). Ya que en el caso e) puede ocurrir que no sea preciso efectuar ningún pivotaje, pues no tiene forzadamente que incum-- plirse la condición (3.3) para la convergencia del método de Newton.

e) El procedimiento converja. Es decir,

$$||H(X^{B,\ell^*}, X^{N,k+1})|| \leq \varepsilon \qquad (3.33)$$

para alguna iteración ℓ^*, donde ε es un va-- lor dado. Si

$$\ell^B < X^{B,\ell^*} < u^B \qquad (3.34)$$

el nuevo punto será $X^{k+1} = (X^{B,\ell^*}, X^{N,k+1})$ y se comienza de nuevo con el método seudo-Newton sustituyendo α_{k+1} por 2 α_{k+1}. Si algún componente X^{B,ℓ^*}_r toma el valor de alguno de sus - límites, se actúa como en los casos anteriores con el correspondiente cambio de base.

3.4 Criterios de terminación.

El algoritmo GRG termina cuando se cumplan - las condiciones de optimalidad de Kuhn y Tucker (3.10 a 3.12) y del problema (3.2) y (3.3) No obstante, se considera que existe un cuasi-óptimo local cuando alternativamente:

$$a)\ |Z^{N,k+1}_j| < \varepsilon_Z |Z^{N,1}_j| \quad j=1,2,\ldots,n-m \qquad (3.35)$$

donde ε_Z es un valor dado, ya que se considera en este caso que la nueva dirección del - desplazamiento es nula.

$$b)\ f(X)^k - f(X^{k+1}) < \varepsilon_f \qquad (3.36)$$

en varias iteraciones sucesivas, donde ε_f es un valor dado.

$$c)\qquad \gamma < \gamma_{max} \qquad (3.37)$$

donde

$$\gamma_j = \begin{cases} Z^{N,k+1}_j (u^N_j - X^{N,k}_j) \,|\, Z^{N,k+1}_j > 0 \\ Z^{N,k+1}_j (\ell^N_j - X^{N,k}_j) \,|\, -Z^{N,k+1}_j < 0 \end{cases} \qquad (3.38)$$

$$\gamma = \max_{(j)} \{\gamma_j\} \qquad (3.39)$$

para j = 1,2,...,n-m, y γ_{max} es un valor dado.

Los criterios de terminación de un algoritmo (y sus valores ε_Z, ε_f y γ_{max}) tienen gran importancia en su convergencia. Ver Wolfe /62/. Son decisivos en la comparación de diferen-- tes algoritmos.

3.5 Técnicas para reducir el tiempo de optimización.

Dentro de los pasos generales que hemos contemplado en el algoritmo GRG se pueden in-- cluir numerosas variantes. Muy frecuentemente el ritmo de convergencia depende de estas variantes. Las más fundamentales son las siguientes:

a) En lugar de obtener $\Delta^B(X)^{-1}$ cada vez que

se precise, se puede aproximar de la forma siguiente:

Sea $\Delta^B(X)^{(k)}$ la matriz básica en la iteración (k) a partir de la cual se aproxima $\Delta^B(X^{(\ell)})^{-1}$ en la iteración (ℓ). Esta aproximación será:

$$\Delta^B(X^{(\ell)})^{-1} \simeq 2\Delta^B(X^{(k)})^{-1} - \Delta^B(X^{(k)})^{-1}\Delta^B(X^{(\ell)})\Delta^B(X^{(k)})^{-1} \qquad (3.40)$$

Abadie, Carpentier y Guigou en la serie de trabajos referenciados al principio de esta sección describen los tests en base a los cuales se obtiene directamente la inversa de la matriz básica en la iteración (ℓ). El tiempo ahorrado es muy considerable y la aproximación muy buena.

La posibilidad (3.40) se ha recogido en el código GRG66 pero no en GRG69.

b) La amplitud α^*_{k+1} tiene un límite adicional al recogido en (3.31) a base de (3.26) y (3.27), tal que una vez obtenido se sustituye por α^*_{k+1}/K si K > 1 donde

$$K=\max_{(i)}\{\sqrt{|h_i(x^k+\alpha^*_{k+1}z^{k+1})|}/\varepsilon_i\} \qquad (3.41)$$

Para i = 1,2,...,m y ε_i es un valor dado. De esta forma probablemente se evita que el método seudo-Newton dé lugar a una no-convergencia en H (X) = 0.

c) Se pueden utilizar (ver secciones siguientes) todas las modernas técnicas variantes del método simplex por lo que respecta a la Forma Producto de la Inversa (FPI), descomposición LU, etc. Escudero /16/ recoge una panorámica muy actualizada. El código GRG66 utiliza el método FPI y el código GRG69 utiliza el procedimiento normal de la inversa de una matriz.

d) Utilización de técnicas anti-zig-zag. La más sencilla consiste en hacer cero la dirección $z_j^{N,k+1}$ si γ_j (3.38) es inferior a un valor dado. No obstante, Zoutendijk /65/, /66/ describe técnicas más sofisticadas.

e) Obtención de una solución factible. Ver Zoutendijk /66/ a este respecto.

El algoritmo GRG es uno de los sistemas de PNL general que tiene todavía una gran aceptación, debido fundamentalmente a la rapidez en su resolución. En las siguientes secciones se recogen otros algoritmos que, basados en éste, recogen los principales avances efectuados en las técnicas de programación matemática.

4. ALGORITMO GRG2

En la sección anterior se ha descrito el algoritmo GRG en las versiones originales debidas a Abadie, Carpentier y Guigou. Debido a la operatividad de este algoritmo, recientemente se han desarrollado otras versiones, ej. debido a a Heltne y Liitschwager /34/, Cohen /11/ y Drud /15/.

En esta sección se describe el algoritmo debido a Lasdon y Warren /40/ denominado GRG2 que adquiere sustantividad propia debido a las innovaciones técnicas que introduce. Vamos a respetar la terminología de la sección anterior y describiremos solamente las diferencias más fundamentales con respecto al algoritmo GRG.

4.1 Variables básicas, superbásicas y no-básicas.

En el algoritmo GRG2 se descompone el vector columna X en el vector de variables básicas X^B (al igual que el algoritmo GRG), en el vector de variables superbásicas X^S que son aquellas variables no-básicas cuyo valor está estrictamente entre sus límites y en el vector X^N de las n-m-s no-básicas, tal que $X = (X^B, X^S, S^N)$. El gradiente reducido $\lambda^N = (\lambda^N_1, \lambda^N_2, \ldots, \lambda^N_j, \ldots, \lambda^N_{n-m-s})$ de las variables no-básicas sirve para analizar si éstas deben dejar de ser no-básicas y transformarse en superbásicas, ya que de lo contrario estas variables quedarían fijas a sus respectivos límites. La dirección del desplazamiento de las variables superbásicas y no-básicas se forma con la dirección $z^S_1, z^S_2, \ldots, z^S_r, \ldots, z^S_s$ de las variables que en la iteración correspondiente sean no-básicas.

El algoritmo GRG2 consiste también en una secuencia de iteraciones; a partir de una solución factible X^O se va efectuando una serie de desplazamientos X^1, X^2, etc. hasta que en una iteración el vector X sea factible en el modelo (3.1) a (3.3) y los vectores fila λ^S y λ^N cumplan las condiciones de Kuhn y Tucker.

Los vectores λ^S y λ^N se obtienen de forma análoga al vector λ (3.9) en el algoritmo GRG. Por tanto,

$$\lambda^S = \nabla_{X^S} f(X) - \nabla_{X^B} f(X) (\Delta^B(X))^{-1} \Delta^S(X) \qquad (4.1)$$

$$\lambda^N = \nabla_{X^N} f(X) - \nabla_{X^B} f(X) (\Delta^B(X))^{-1} \Delta^N(X) \qquad (4.2)$$

donde $\nabla_{X^N} f(X)$ y $\Delta^N(X)$ tienen el mismo significado que en el algoritmo GRG y

$\nabla_{X^S} f(X)$ y $\Delta^S(X)$ representan para las variables superbásicas lo mismo que el vector $\nabla_{X^N} f(X)$ y la matriz $\Delta^N(X)$ para las variables no-básicas. Las condiciones de optimalidad de Kuhn y Tucker son:

$$\lambda_j^N \geq -\varepsilon \text{ si } X_j^N = \ell_j^N \quad j=1,2,\ldots,n-m-s \qquad (4.3)$$

$$\lambda_j^N \leq \varepsilon \text{ si } X_j^N = u_j^N \quad j=1,2,\ldots,n-m-s \qquad (4.4)$$

$$-\varepsilon < \lambda_r^S < \varepsilon \quad r=1,2,\ldots,s \qquad (4.5)$$

donde ε es un valor dado que por defecto el algoritmo GRG2 no considera que es cero, sino 10^{-4}.

El algoritmo GRG2 considera alternativamente que se ha obtenido el óptimo local si después de un número determinado (3 por defecto) de iteraciones consecutivas,

$$\frac{f(X^k) - f(X^{k+1})}{f(X^k)} < \varepsilon \qquad (4.6)$$

donde ε es un valor dado (por defecto, 10^{-4}).

Si en la iteración k, X^k no es óptimo (ya que una vez actualizados para X^k los vectores $\nabla_{X^B} f(X)$, $\nabla_{X^S} f(X)$ y $\nabla_{X^N} f(X)$, matrices $\Delta^B(X)$, $\Delta^S(X)$ y $\Delta^N(X)$ y vector $\lambda=(\lambda^S,\lambda^N)$ (4.1 y 4.2), éste no cumple en (4.3) a (4.5) las condiciones de Kuhn y Tucker, se determina la dirección del desplazamiento siendo $Z^{S,k+1}$ para las variables superbásicas y $Z^{N,k+1}=0$ para las variables no-básicas. Una vez obtenido $Z^{S,k+1}$ y $Z^{N,k+1}$, se obtiene la dirección del desplazamiento en las variables básicas, cuyo vector $Z^{B,k+1}$ de forma análoga a (3.25) será:

$$Z^{B,k+1} = -(\Delta^B(X))^{-1} \Delta^S(X) Z^{S,k+1} \qquad (4.7)$$

A continuación es preciso determinar la amplitud máxima del desplazamiento que se puede efectuar en las variables superbásicas sin tener que cambiar de base. Esta cuantía α_{k+1}^1 se obtiene con la formulación (3.26) de forma análoga a como se efectúa en el algoritmo GRG. Si $\alpha_{k+1}^1 < \varepsilon$, donde ε es un valor dado, se considera que la base es degenerada y por tanto se procede a efectuar un cambio de base con las reglas recogidas en el apartado 4.5.

Si la base no es degenerada, después de obtener la dirección del desplazamiento $Z^{S,k+1}$ (apartado 4.2) y $Z^{N,k+1}=0$, se procede a determinar la amplitud α_{k+1}^* de este desplazamiento (apartado 4.3) a base del método seudo-Newton, a obtener el vector $X^{B,k+1}$ de soluciones básicas. Si no se puede determinar un valor para la amplitud α_{k+1}^* para el que el método seudo-Newton converja hacia un punto en el que se disminuya la función objetivo, se trasladan algunas variables no-básicas al vector de variables superbásicas y, por tanto, el vector del gradiente reducido superbásico λ^S se ve incrementado con los valores λ_j^N de las variables trasladadas. Estas variables serán aquellas para las que

$$\lambda_j^N < -\varepsilon \text{ si } X_j^N = \ell_j^N \quad j=1,2,\ldots,n-m-s \qquad (4.8)$$

$$\lambda_j^N > \varepsilon \text{ si } X_j^N = u_j^N \quad j=1,2,\ldots,n-m-s \qquad (4.9)$$

donde ε es un valor dado.

También se trasladarán al vector superbásico aquellas variables no-básicas para las que se cumpla (4.8) y (4.9) si para todas las variables superbásicas se cumple (4.5).

4.2 Dirección del desplazamiento. Vector $Z^S = (Z_1^S, Z_2^S, \ldots, Z_r^S, \ldots, Z_s^S)^t$.

La dirección del desplazamiento $Z^{S,k+1}$ desde el punto superbásico $X^{S,k}$ a $X^{S,k+1}$ (considerando naturalmente que el punto X^k no cumple las condiciones de optimalidad de Kuhn y Tucker) tiene dos tipos de métodos en su obtención en el algoritmo GRG2. Si el número s de variables superbásicas es menor que un valor dado (n, por defecto), la técnica utilizada se basa en los métodos Quasi-Newton para optimizar una función multivariante sin restricciones. En caso contrario se utilizan métodos adaptados del método del gradiente con

jugado ya que precisan menor capacidad de almacenamiento y son normalmente más rapidos. Escudero /16/ recoge una amplia bibliografía sobre estos tipos de métodos en PSR.

a) Método Quasi-Newton

El método Quasi-Newton en el algoritmo GRG2 efectúa una aproximación a la matriz hessiana de la función objetivo f(X) en la que las variables básicas se expresan en función de las variables superbásicas, y las variables no-básicas permanecen fijas. Se parte de la base de que se aproxima la función f(X) (3.1) de una forma cuadrática desarrollándola en series de Taylor y despreciando el tercer -- término y superiores. Por tanto,

$$f(X^{k+1}) = f(X^k) + \nabla_{X^N} f(X)\,(X^{N,k+1} - X^{N,k}) + \nabla_{X^B} f(X)\,(X^{B,k+1} - X^{B,k}) + \nabla_{X^S} f(X)\,(X^{S,k+1} - X^{S,k}) + (X^{k+1} - X^K)^t G_T (X^{k+1} - X^k)/2 \qquad (4.10)$$

donde G_T es la matriz Hessiana $\delta^2 f(X)/\delta X^2$ de orden n*n de todas las variables (X^B, X^S, X^N) del problema.

Al objeto de facilitar la lectura, sea el siguiente cambio de terminología:

$$g = (\nabla_{X^B} f(X), \nabla_{X^S} f(X)) \qquad (4.11)$$

$$d = \begin{pmatrix} d_B \\ d_S \end{pmatrix} = \begin{pmatrix} \alpha^*_{k+1} Z^{B,k+1} \\ \alpha^*_{k+1} Z^{S,k+1} \end{pmatrix} \qquad (4.12)$$

$$B_B = \Delta^B(X) \qquad (4.13)$$

$$B_S = \Delta^S(X) \qquad (4.14)$$

Dado que $X^{N,k+1} - X^{N,k} = 0$ y que $X^{B,k+1} - X^{B,k} = d_B$ y $X^{S,k+1} - X^{S,k} = d_S$, resulta que:

$$f(X^{k+1}) = f(X^k) + \nabla_{X^B} f(X)\, d_B + \nabla_{X^S} f(X)\, d_S + d^t G d/2 \qquad (4.15)$$

donde G es la matriz $\delta^2 f(X)/\delta X^2$ de orden -- (m*s)+(m*s) de las variables básicas y superbásicas. Lasdon y Warren /40/ se basan en -- Murtagh y Saunders /46/ para obtener en el algoritmo GRG2 la dirección d_S (en la cual también está incluida la amplitud aunque no tiene mayor importancia) del desplazamiento de las variables superbásicas X^S. A su vez, estos autores se basan en la operativa desarrollada por Gill y Murray (/26/ pp.67-71), y validada por los mismos autores en una serie de trabajos referenciados en dicho libro para obtener la dirección del desplazamiento en el caso de condiciones lineales. Esta operativa se basa en la ortogonalidad de cierta matriz T con la matriz de condiciones tal como se recoge a continuación. Gill y Murray -- /29/ presentan una excelente panorámica de estos trabajos.

Los nuevos valores de las variables X^B y X^S deben ser tal que la dirección d (4.12) cumpla las condiciones (3.2) del problema. Por tanto, y según la linealización (3.22), resulta que d debe ser tal que:

$$(B_B,\ B_S)\, d = (B_B, B_S) \begin{pmatrix} d_B \\ d_S \end{pmatrix} = 0 \qquad (4.16)$$

Ahora bien, como d_B depende de d_S, sea T la matriz (m+s)*s para la cual

$$\begin{pmatrix} d_B \\ d_S \end{pmatrix} = T\, d_S \qquad (4.17)$$

Por tanto, si la matriz T fuese ortogonal a la matriz de condiciones, es decir si

$$(B_B,\ B_S)\ T = 0 \qquad (4.18)$$

resultaría que la dirección d cumpliría las condiciones (3.2). Ahora bien para que se -- cumpla (4.18), T debe ser

$$T = \begin{pmatrix} -B_B^{-1}\, B_S \\ I \end{pmatrix} \qquad (4.19)$$

donde $-B_B^{-1}\, B_S$ es una matriz de orden m*s e I es la matriz unidad s*s. Si T es (4.19) -- resulta que

$$(B_B, B_S)\, T = (B_B, B_S) \begin{pmatrix} -B_B^{-1}\, B_S \\ I \end{pmatrix} = -B_B\, B_B^{-1}\, B_S + B_S = 0 \qquad (4.20)$$

que confirma la formulación (4.7).

De donde, y según (4.1), (4.15) será:

$$f(X^{k+1}) = f(X^k) + gd + d^t\, G\, d =$$

$$= f(X^k) + (\nabla_{X^B} f(X), \nabla_{X^S} f(X))\, T d_S + (T d_S)^t G T d_S =$$

$$=f(X^k)+d_S^t(\lambda^S)^t+d_S^t\ T^t\ G\ T\ d_S/2 \qquad (4.21)$$

Por tanto, la dirección d_S que minimice esta función será aquella para la cual sus primeras derivadas sean cero:

$$T^t\ G\ T\ d_S = -(\lambda^S)^t \qquad (4.22)$$

y la matriz Hessiana, que en este caso será la matriz T^tGT que denominaremos B, sea positiva definida. Tradicionalmente se ha obtenido d_S:

$$d_S = -\ H\ (\lambda^S)^t \qquad (4.23)$$

a base de aproximar iterativamente la matriz inversa H de la hessiana B por los métodos Quasi-Newton tales como los denominados DFP, BFGS y otros de rango 1 y 2. Escudero /16/ recoge una bibliografía muy amplia sobre estas técnicas.

Gill y Murray (/26/pp. 74-76) indican que de esta forma, y debido a errores de cálculo, la matriz resultante B de esta aproximación puede dejar de ser positiva definida, en cuyo caso el método puede no converger. Sustitutivamente sugieren aproximar directamente la matriz B a base de un procedimiento que se recoge más abajo. Así mismo, estos autores sugieren un procedimiento (recogido más abajo) para obtener d_S a partir de la fórmula:

$$L\ D\ L^t\ d_S = -(\lambda^S)^t \qquad (4.24)$$

donde L es una matriz triangular inferior de orden s*s con diagonal unitaria y D es una matriz diagonal, dado que

$$B = T^t\ G\ T \qquad (4.25)$$

es una matriz simétrica y, por tanto, según la factorización de Cholesky se puede expresar a través de (4.24). Wilkinson /59/ describe claramente esta factorización. Ver también Brodlie /9/.

Por tanto, la operativa en el método Quasi-Newton para obtener la dirección d_S del desplazamiento es la siguiente:

1) Dada una solución factible $X^k=(X^{B,k},X^{S,k},X^{N,k})$ se obtiene el gradiente g (4.11), las matrices B_B (4.13) y B_S (4.14) y, por tanto, la matriz T (4.19).

2) Obtención según (4.1) del gradiente reducido λ^S.

3) Obtención de la matriz hessiana $B(=T^tGT)$. En la iteración k se obtendrá directamente a base de la información existente en la iteración k-1, según el procedimiento sugerido por Gill y Murray(/25/ y /26/ pp. 74-76). Utilizan el método denominado complementary DFP formula surgido de un análisis de la fórmula de Davidon /14/ mejorada por Fletcher y Powell /21/ y de la fórmula BFGS debida a Broyden /10/, Fletcher /20/, Goldfarb /31/ y Shanno /56/. Esta fórmula es la siguiente:

$$B^k=B^{k-1}+\frac{(\lambda^{S,k}-\lambda^{S,k-1})^t(\lambda^{S,k}-\lambda^{S,k-1})}{\alpha_k^*\ z^{S,k}\ (\lambda^{S,k}-\lambda^{S,k-1})^t}+\frac{(\lambda^{S,k-1})^t\lambda^{S,k-1}}{z^{S,k}(\lambda^{S,k-1})^t} \qquad (4.26)$$

tal que $B^o = T^{ot}\ G^o\ T^o$ donde T^o y G^o se obtienen a base de la solución inicial factible $(X^{B,o},X^{S,o})$. Es interesante observar que en la formulación (4.26) la modificación de B^{k-1} para obtener B^k no exige manipulación de matrices. La matriz resultante será siempre positiva definida si los elementos de la matriz diagonal D (en 4.24) son positivos. Esta es una condición más fácil de cumplir.

Dado que en B^o se ha efectuado la factorización LDL^t (ver Wilkinson /59/) la actualización (4.26) no modifica esta factorización si se sigue la operativa descrita por Gill y Murray /29/. Gill y Murray /26/ y /29/ describen la posibilidad de utilización de la factorización QR para obtener los multiplicadores de Lagrange (3.8) y la matriz T (4.19). Así mismo describen la posibilidad de modificar las factorizaciones LDL^t y QR (Jennings /38/ y Gill y Murray, /30/) sin precisar obtenerlas de nuevo en el caso de un cambio de base. Aunque este procedimiento es muy adecuado para el algoritmo de programación cuadrática con condiciones lineales que ellos proponen (Gill y Murray, /29/), podría utilizarse con ligeras modificaciones en algoritmos generales de PNL.

4) Obtención de la dirección d_S del desplazamiento a base de resolver el sistema de ecuaciones (4.22). En lugar de obtener d_S a base

de (4.23) lo que precisaría obtener la inversa H de B, Gill y Murray /26/ sugieren un procedimiento más rápido y que además evitará errores de cálculo. Avriel (/5/ p. 352) - efectúa una exposición clara y precisa de esta técnica. Este procedimiento se basa en -- que la matriz B es simétrica, y la actualización efectuada en (4.26) no le ha cambiado - su carácter de factorización LDL^t. A partir de (4.24), d_S se obtiene de la siguiente forma, donde todos los vectores y matrices tienen el superíndice k excepto d_S que corresponde al superíndice k+1:

a) Obtención del vector columna Y resolviendo el sistema de ecuaciones

$$L\,Y = -(\lambda^S)^t \qquad (4.27)$$

Por tanto,

$$Y_1 = -\lambda_1^S$$

$$Y_r = -\lambda_r^S - \sum_{i=1}^{r-1} L_{ri}\,Y_i \qquad r=2,\ldots s \qquad (4.28)$$

b) Dado que

$$D\,L^t\,d_S = Y \qquad (4.29)$$

resulta que

$$L^t\,d_S = D^{-1}\,Y \qquad (4.30)$$

de donde

$$d_{Ss} = \frac{Y_s}{D_{ss}} \qquad (4.31)$$

$$d_{Sr} = \frac{Y_r}{D_{rr}} - \sum_{i=r+1}^{s} L_{ir}\,d_{Si} \qquad (4.32)$$

$$r = 1,2,\ldots,s-1$$

5) Obtención de la dirección d_B del desplazamiento de las variables básicas X^B. Según -- (4.17), se aplica directamente la fórmula -- (4.7). Contemplando (4.12) se puede observar que lo que se ha venido llamando dirección d es en realidad la dirección y la amplitud -- del desplazamiento. Realmente no tiene mayor problema ya que el siguiente paso (apartado 4.3) consiste en determinar la amplitud α^*_{k+1} sobre d_S y d_B.

b) Método del Gradiente Reducido Conjugado

Aunque el método Quasi-Newton tiene en esta versión una gran mejora sobre los tradicionales Quasi-Newton, si se utiliza la técnica - del gradiente reducido conjugado se obtiene una convergencia análoga y, por otra parte, ahorra tiempo de optimización y precisa menor capacidad de almacenamiento. Se utiliza la metodología recogida en el algoritmo GRG. Por tanto, de forma análoga a las formulaciones (3.18) a (3.20),

$$z^{S,k+1} = -(\lambda^{S,k})^t + \beta^k z^{S,k} \qquad (4.32)$$

donde el escalar β^k dependerá de la formulación utilizada. El algoritmo GRG2 utiliza -- las formulaciones β^k_{FP} debida a Fletcher y -- Reeves /22/, β^k_{PR} debida a Polak y Ribière -- /50/ y β^k_P debida a Perry /48/, siendo éstas:

$$\beta^k_{FP} = ||\lambda^{S,k}||^2 / ||\lambda^{S,k-1}||^2 \qquad (4.33)$$

$$\beta^k_{PR} = \lambda^{S,k}(\lambda^{S,k} - \lambda^{S,k-1})^t / ||\lambda^{S,k-1}||^2 \qquad (4.34)$$

$$\beta^k_P = \lambda^{S,k}((\lambda^{S,k} - \lambda^{S,k-1})^t - \alpha^*_k z^{s,k}) / (\lambda^{S,k} - \lambda^{S,k-1}) z^{s,k} \qquad (4.35)$$

En la iteración k = 1, $\beta^k = 0$. Así mismo se reinicializa el procedimiento (β^k=0) para -- k=s+1, 2s+1,...,etc. En estos casos la dirección $z^{S,k+1}$ será $-(\lambda^{S,k})^t$, es decir $\beta^k = 0$. También se efectúa cuando cambia el status - del vector X tal que difiera la composición de los vectores X^B, X^S y X^N. Powell /52/ estudia diversas formas de reinicializar la dirección del desplazamiento. Murtagh y Saunders /46/ recogen la operativa correspondiente.

4.3 Amplitud α_{k+1} del desplazamiento.

Existen numerosas versiones para su obtención y el detalle de cada una de ellas es muy prolijo. Ver Murtagh y Saunders /46/ y Lasdon y Warren /40/ para mayor precisión. Dado que - $\alpha^*_{k+1} \geq 0$, el objetivo consiste en cumplir la condición en (3.28) basada en (3.26) y (3.27). Los pasos fundamentales son los siguientes:

1) Se parte de un primer valor de α_{k+1}. Si - converge el algoritmo denominado seudo-Newton (apartado 4.4) al determinar las correspondientes variables básicas X^B y, por tanto, - se cumple la condición (3.2), no se violan - sus límites (3.4) y la función objetivo se - ha reducido, se sustituye α_{k+1} por 2 α_{k+1} y se comienza de nuevo el algoritmo seudo-Newton (apartado 4.4).

2) Si en el paso 1, se cumplen todos sus requisitos pero la función objetivo es mayor - que la obtenida previamente de tal forma que

se hayan obtenido las siguientes amplitudes A, B y C, tal que:

$$0 \leq A \leq B \leq C \qquad (4.36)$$

$$f(X^k+AZ^{k+1}) \geq f(X^k+BZ^{k+1}) \leq f(X^k+CZ^{k+1}) \qquad (4.37)$$

entonces el intervalo (A,C) contiene un mínimo local de $F(X^k+\alpha^*_{k+1}Z^{k+1})$. Entonces se ajusta una función cuadrática en α_{k+1} a los puntos A, B, C y sus correspondientes funciones (4.37). El valor mínimo será α^*_{k+1} y se comienza de nuevo el algoritmo GRG2.

3) Sea el caso recogido en el paso 1, pero en el que el algoritmo seudo-Newton (apartado 4.4) no converge o precisa más de 6 iteraciones. En este caso, se termina la búsqueda del valor óptimo α^*_{k+1} y se considera que éste es el último obtenido si ha producido una reducción en la función objetivo con respecto al valor previo de α_{k+1}. Si no ha obtenido esta mejora se sustituye α_{k+1} por $\alpha_{k+1}/2$ y comienza de nuevo el algoritmo seudo-Newton. También se actúa de la misma forma si el algoritmo seudo-Newton no converge para el primer valor de α_{k+1}.

4) Si el algoritmo seudo-Newton converge, pero se viola alguno de los límites de las variables básicas, el caso se trata en el apartado 4.5.

4.4 Cumplimiento de las condiciones del problema H (X) = . Método seudo-Newton.

Esencialmente se sigue el mismo procedimiento del método seudo-Newton utilizado en el algoritmo GRG, considerando fijas a sus valores las variables superbásicas $X^{S,k+1}$ y no-básicas $X^{N,k+1}$. Al igual que en GRG también en GRG2 la matriz básica $\Delta^B(X)$, que también hemos llamado B_B, sólo se recalcula al principio de cada iteración y no se recalcula en cada iteración ℓ con los valores $X^{B,\ell}$ como se efectúa en el algoritmo de Newton.

Si en la iteración ℓ del algoritmo seudo-Newton se violan los límites de X^B, no se cambia de base como hace el algoritmo GRG, sino que se continúa el procedimiento hasta que converja, ya que puede ocurrir que entonces no sea preciso efectuar ningún pivotaje.

4.5 Cambio de base.

Si el algoritmo seudo-Newton converge, pero se viola alguno de los límites (3.4) de las variables básicas, la amplitud causante es reducida a un valor en el que no haya ningún límite violado y al menos una variable tome el valor de uno de sus límites. Se determina este valor de la amplitud a base de efectuar una interpolación lineal entre los valores previos y los valores actuales de los variables básicas cuyos límites se han violado. Para la amplitud resultante (se hubieran o no previamente violado los límites) se obtienen los valores definitivos de las variables básicas $X^{B,\,k+1}$ y superbásicas $X^{S,k+1}$ y se actualizan los vectores $\nabla_B f(X)$, $\nabla_S f(X)$, $\nabla_N f(X)$ y las matrices $\Delta^B(X)$, $\Delta^S(X)$ y $\Delta^N(X)$.

A partir de estos nuevos datos se procede a determinar las variables básicas de la nueva iteración, que si hay cambio de base, es como si de nuevo (con una solución mejor) se comenzará el problema.

La determinación de las nuevas variables básicas X^B (que pueden ser algunas de las antiguas) y, por tanto, la obtención de la matriz $\Delta^B(X)^{-1}$ se efectúa de la forma siguiente. Después de cada pivotaje, por cada columna no-pivotada se determina el elemento con mayor valor absoluto entre las filas no-pivotadas y se obtiene, sea M, el máximo de estos elementos. La columna pivote será aquella entre las no-pivotadas cuya variable tenga la máxima diferencia entre su valor (proveniente de la determinación de la amplitud α^*_{k+1}) y su límite más cercano, siempre que el valor absoluto del elemento pivote sea mayor que εM, donde ε es un valor dado (generalmente, $\varepsilon=0.1$). El elemento pivote en esta columna será el de mayor valor absoluto.
Una vez seleccionadas las m variables básicas, se obtiene la matriz $\Delta^B(X)^{-1}$, y se comienza de nuevo el algoritmo GRG2. Esta forma de seleccionar el vector X^B reducirá el número de cambios de base a efectuar.

Lasdon y Warren /40/ reconocen que el algoritmo GRG2 no está todavía totalmente validado, pero dada la técnica empleada y los resultados preliminares recogidos en el trabajo referenciado, este algoritmo puede ser uno de los sistemas básicos en programación no-lineal general.