

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

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.

5. ALGORITMO MINOS (MODULAR IN-CORE NON-LINEAR OPTIMIZATION SYSTEM)

El algoritmo MINOS (Modular In-core Non-Linear Optimization System) es un algoritmo de tipo GRG diseñado para soportar problemas de PNL con condiciones lineales, un gran número de variables y matriz de poca densidad. Su campo de aplicación es amplio: problemas típicos de planificación multiperíodo, etc. de PL con función objetivo no-lineal. Ha sido diseñado por Murtagh y Saunders /46/. El algoritmo GRG2 toma del MINOS la mayoría de las técnicas utilizadas. No obstante, el MINOS está expresamente adaptado al problema descrito.

Las características fundamentales que le diferencian del algoritmo GRG2 son las siguientes:

- 1) No precisa el algoritmo iterativo quasi-Newton para obtener el vector X^B dados X^S y X^N ya que las condiciones son lineales.
- 2) Las matrices B_B y B_S son fijas por la misma razón; de donde sólo se cambiará la matriz B_B cuando sea preciso efectuar un

cambio de base. Una consecuencia adicional consiste en que ahora es posible actualizar la matriz inversa B_B^{-1} con la técnica LU y Forma de Eliminación de la Inversa, en lugar de obtener la inversa por el clásico de Gauss-Jordan como se efectúa en el algoritmo GRG2. Esta posibilidad es muy interesante dado que la matriz de condiciones tiene una densidad muy pequeña. Asimismo, se simplifica enormemente la obtención de la matriz hessiana B y del gradiente reducido λ^S ya que sólo es preciso calcular los gradientes

$$\nabla_{X^B} f(X) \quad \text{y} \quad \nabla_{X^S} f(X)$$

para cada punto X . Si se cambia de base, entonces también habría que actualizar B_B^{-1} .

- 3) Utiliza las técnicas de quasi-Newton y del gradiente reducido conjugado utilizadas por el método GRG2 en la determinación de la dirección del desplazamiento. Murtagh y Saunders /46/ efectúan un estudio con numerosos casos reales comparativos sobre estas técnicas y otras de tipo ran-

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

go 1. En la nueva versión de MINOS (Murtagh y Saunders, /47/, se observa una convergencia más rápida en la utilización de las técnicas de quasi-Newton sobre el gradiente reducido conjugado. En cambio, de los numerosos casos analizados no se puede deducir cual de las técnicas del gradiente reducido conjugado es mejor. Para más de $s=200$ variables superbásicas, la capacidad de memoria requerida hace conveniente utilizar las segundas técnicas sobre las primeras.

- 4) Utiliza la técnica denominada delinesearch debida a Gill y Murray, /28/, que, basada en sucesivas interpolaciones cúbicas, determina la amplitud del desplazamiento.
- 5) Utiliza todas las modernas técnicas de almacenamiento y manejo de datos utilizadas en los códigos generales de PL. Escudero, /16/, describe con bastante detalle estas innovaciones. Así, utilizan las técnicas de preparación de datos del sistema MPSX IBM, /37/, con lo cual hacen compatible el modelo con un gran número de sistemas, las técnicas de representación y almacenamiento de una matriz de Kalan, /39/, las técnicas de la Forma de Eliminación de la Inversa (FEI) en la inversión B_B^{-1} de Hellerman y Rarick, /33/, complementada con la técnica LU en la creación del vector η de cada pivotaje debida a Forrest y Tomlin, /23/, las técnicas empleadas en la fase 1 de los sistemas de PL, etc.
- 6) El cambio de base (y subsiguiente actualización de B_B^{-1}) se efectúa sólo sobre una variable. Esta es aquella para la que su valor ha tomado el de uno de sus límites en la fase de determinación de la amplitud del desplazamiento. El pivotaje se efectúa con las técnicas sofisticadas de PL.
- 7) Al seleccionar la variable entrante en la base, en el pivotaje se da preferencia a las variables superbásicas sobre las variables no-básicas. Murtagh y Saunders, /46/, describen el conjunto de tests tal que si las variables X^S no los cumplen, el pivotaje se efectúa sobre las variables X^N . Estos tests se basan en el ritmo de cambio de X^S y $f(X)$ en las iteraciones anteriores, en el valor $\|\lambda^S\|$ obtenido a ba-

se del gradiente reducido de X^S , etc.

El algoritmo MINOS tiene una validación muy exhaustiva con una gran gama de problemas -- reales (problemas químicos, de planificación energética, etc.) algunos con dimensiones de $m=1000$ y $n=2000$ a 2500. La densidad de la matriz oscila entre el 1 y 1,3%.

El algoritmo LSGRG trata problemas con función objetivo y condiciones no-lineales, pero con matrices de poca densidad. Es una extensión del algoritmo MINOS, pero utiliza -- las técnicas del algoritmo GRG2 para acomodar las condiciones no-lineales.

6. ALGORITMO CGAP (CONJUGATED GRADIENT APPROXIMATION PROGRAMMING)

Beale, /7/, describe por primera vez el algoritmo CGAP (Conjugated Gradient Approximation Programming) como una variante del algoritmo MAP de Griffith y Stewart, /32/, aunque más certestamente se podría considerar como variante del algoritmo del gradiente reducido de Wolfe, /60/, /61/, y de la generalización de GRG de Abadie y Carpentier, (/2/, /3/) y -- otros. Batchelor y Beale, /6/, describen una nueva versión del algoritmo CGAP y Beale, /8/, recoge sus características más fundamentales. En este trabajo nos remitiremos a glosar las principales características del algoritmo CGAP. Para detalles adicionales nos remitimos a los trabajos anteriores, principalmente Batchelor y Beale /6/. El detalle del algoritmo es muy prolijo y hace un uso extensivo de parámetros al controlar la secuencia de iteraciones.

6.1 Variables lineales y no-lineales. Diagrama del proceso.

Una de las principales características del algoritmo CGAP consiste en la distinción entre variables lineales y no-lineales. Esta distinción ya aparece en el trabajo de Griffith y Stewart, /32/, pero es en el CGAP donde tiene su mayor aplicación. Sea el problema

$$\min f(X,Y) \tag{6.1}$$

sujeto a

$$g_i(X, Y) = g_i, \quad i=1, 2, \dots, m \quad (6.2)$$

$$0 \leq X \leq U^X \quad (6.3)$$

$$L \leq Y \leq U \quad (6.4)$$

tal que se considera que la variable Y_r -- ($r=1, 2, \dots, R$) es no-lineal si dando un valor fijo a las variables del vector

$$Y = (Y_1, Y_2, \dots, Y_r, \dots, Y_R)^t$$

el problema (6.1) a (6.4) se convierte en un problema de programación lineal con las variables del vector

$$X = (X_1, X_2, \dots, X_j, \dots, X_n)^t$$

Esta es la razón por la cual X_j será una variable lineal.

Siendo fijo el vector Y , el problema será

$$\min Z = b_0 + aX \quad (6.5)$$

sujeto a

$$AX = b \quad (6.6)$$

$$0 \leq X \leq U^X \quad (6.7)$$

donde a es un vector $1 \times n$, A la matriz de condiciones $m \times n$, U^X el vector $n \times 1$ de los límites del vector de variables X , b un vector $m \times 1$ que recoge el RHS (6.2) del problema primitivo así como los términos de la condición (6.2) en los que sólo intervienen variables no-lineales y b_0 el coeficiente de la función objetivo en el que sólo intervienen variables no-lineales. Por tanto, el problema será

$$\min Z = b_0 + \sum_{j=1}^n a_{0j} X_j \quad (6.8)$$

sujeto a

$$\sum_{j=1}^n a_{ij} X_j = b_i, \quad i=1, 2, \dots, m \quad (6.9)$$

$$0 \leq X_j \leq U_j, \quad j=1, 2, \dots, n \quad (6.10)$$

Los coeficientes b_i y a_{ij} para $i=0, 1, 2, \dots, m$ y $j=1, 2, \dots, n$ son constantes ó dependen de -

las variables no-lineales Y ; por tanto $b_i(Y)$ y $a_{ij}(Y)$.

A grandes rasgos se puede indicar que los pasos del algoritmo CGAP para cada iteración k son los siguientes:

- 0) Se parte de una solución inicial factible X e Y .
- 1) Determinación de las variables no-lineales que se considerarán independientes. - Otros autores (e.g. en los algoritmos -- GRG2, MINOS y LSGRG) las denominan superbásicas, aunque no hacen la distinción entre variables lineales y no-lineales. A este paso se le denomina exploratory step. En la iteración $k=1$, se obtienen las variables independientes a base de linealizar la función objetivo y condiciones -- (por tanto, se desarrollan en series de Taylor despreciando el segundo término y siguientes) alrededor de la solución inicial X e Y y resolver el correspondiente subproblema de programación lineal (PL). En las demás iteraciones se sustituye la solución inicial, por la solución obtenida en la iteración anterior. Se considera variables independientes a aquellas variables no-lineales no-básicas que reúnan -- ciertas condiciones que se glosan más abajo.
- 2) Actualización (reducción) de las tolerancias de las variables no-lineales.
- 3) Determinación del gradiente reducido de las variables independientes. Para ello se efectúa una nueva linealización de la función objetivo y condiciones alrededor de la solución óptima X e Y obtenida en el paso anterior, reduciendo a cero las tolerancias de las variables independientes. Se resuelve el correspondiente subproblema de PL y, por tanto, se obtiene el gradiente reducido de las variables independientes. A este paso se le denomina Base step.
- 4) Determinación de la dirección del desplazamiento de las variables independientes. Se obtiene en primer lugar la escala o -- normalización del gradiente reducido y, -- según sea el caso, el gradiente reducido conjugado. El negativo del gradiente redu

cido conjugado serán la dirección del desplazamiento de las variables independientes.

5) Amplitud del desplazamiento de las variables independientes. En base a una técnica muy sofisticada se obtiene la amplitud de este desplazamiento.

a) El valor intermedio de la amplitud se obtiene en base al valor máximo de desplazamiento de las variables independientes dadas las tolerancias y los límites de éstas, en orden a no dejar de ser independientes.

b) En base a la diferencia del valor intermedio de la amplitud con respecto al valor intermedio previamente calculado, en base a las derivadas (directional derivatives) de la función objetivo y variables básicas, calculadas en la solución óptima del subproblema de PL inmediatamente optimizado, y en base a los límites de estas variables, se obtiene un nuevo punto X e Y.

c) Alrededor del punto X e Y se vuelve a linealizar y optimizar el correspondiente subproblema de PL, considerando fijas las variables independientes a los valores obtenidos en base al valor intermedio de la amplitud y a la dirección de desplazamiento previamente calculados.

d) Según sea el valor óptimo de la función objetivo en la nueva solución así como el nuevo gradiente de las variables independientes así se decide: (a) continuar la obtención de nuevos valores de la amplitud en la misma dirección del desplazamiento (aunque ahora haya menos variables independientes) para lo cual se bifurca al paso 5b), (b) investigar una nueva dirección y amplitud del desplazamiento de las variables -- que todavía continúan siendo independientes y cuyo nuevo gradiente reducido no es cero para lo cual se bifurca al paso 4, (c) efectuar una nueva iteración k+1 comenzando con el paso exploratorio para lo cual se bifurca al paso 1 (sería el caso en el que ya no quedan variables independientes o el -

gradiente reducido es cero), (d) obtener una nueva amplitud del desplazamiento de las variables independientes -- efectuando una interpolación (lineal -- en la mediana o cúbica) entre los dos últimos valores obtenidos de la amplitud (sería el caso en el que el valor de la función objetivo es superior al valor anterior, todavía hay variables independientes y el nuevo gradiente reducido no es cero) y a continuación se bifurca al paso 5b), ó (e) finalizar el problema si se cumplen los criterios standard de terminación, ya descritos en las secciones anteriores.

En la optimización de cada subproblema lineal, el algoritmo CGAP utiliza el sistema SCICONIC (Scicon, /55/, que, junto con el sistema MPSX (IBM, /37/), es uno de los sistemas de PL que mayores innovaciones técnicas tiene incorporadas tanto en el almacenamiento y tratamiento de la matriz de condiciones, como en la selección del pivote y en la reducción del incremento del número de elementos no-nulos de la matriz actualizada de condiciones. Esta reducción se efectúa a base de las técnicas de factorización triangular en forma de LU de los vectores eta y de la técnica de la Forma de Eliminación de la Inversa (FEI) en la reinversión de la matriz básica, entre otras innovaciones efectuadas en los cinco últimos años. El consiguiente ahorro de tiempo de CPU en la optimización y, por tanto, en el incremento de las dimensiones del problema a optimizar es considerable. Escudero, /16/, describe estas nuevas posibilidades. Utilizando este sistema se han tratado, además de los típicos problemas-test, problemas reales de hasta m=2336 condiciones, n=3984 variables lineales y R=408 variables no-lineales.

6.2 Paso exploratorio

En el paso exploratorio se determinan las variables independientes como resultado de la linealización y optimización del problema no-lineal (6.1) a (6.4). El algoritmo CGAP clasifica las variables X e Y en: (a) variables lineales básicas

$$X^B = (X_1^B, X_2^B, \dots, X_i^B, \dots, X_B^B)^t,$$

(b) variables lineales no-básicas

$$X^N = (X_1^N, X_2^N, \dots, X_j^N, \dots, X_N^N)^t,$$

(c) variables no-lineales básicas

$$Y^B = (Y_1^B, Y_2^B, \dots, Y_i^B, \dots, Y_B^B)^t,$$

(d) variables no-lineales no-básicas

$$Y^N = (Y_1^N, Y_2^N, \dots, Y_j^N, \dots, Y_N^N)^t \text{ y}$$

(e) variables independientes

$$Y^I = (Y_1^I, Y_2^I, \dots, Y_j^I, \dots, Y_I^I)^t.$$

Esta linealización se efectúa alrededor de la solución inicial X e Y ó del punto obtenido en la optimización anterior. Para ello se desarrollan en series de Taylor (despreciando segundos términos y superiores) la función objetivo y condiciones. Las variables X estarán acotadas según (6.3) y las variables Y estarán acotadas, además de por sus límites, por las tolerancias $t=(t_1, t_2, \dots, t_r, \dots, t_R)^t$ impuestas a su desplazamiento. Por tanto, -- considerando que la solución inicial ó la obtenida previamente es X^0 e Y^0 , se obtienen los correspondientes valores a_{ij}^0 y b_i^0 de a_{ij} y b_i , que sólo dependen de Y , así como la linealización de (6.1) y (6.4) (ó (6.5) y (6.6)) que según (6.8) a (6.10), será:

$$Z = b_0^0 + \sum_{j=1}^n a_{0j}^0 X_j^0 + \sum_{j=1}^n a_{0j}^0 (X_j - X_j^0) + \sum_{r=1}^R \frac{\delta b_0}{\delta Y_r} (Y_r - Y_r^0) + \sum_{r=1}^R c_{0r} (Y_r - Y_r^0) \quad (6.11)$$

$$- b_i^0 + \sum_{j=1}^n a_{ij}^0 X_j^0 + \sum_{j=1}^n a_{ij}^0 (X_j - X_j^0) - \sum_{r=1}^R \frac{\delta b_i}{\delta Y_r} (Y_r - Y_r^0) + \sum_{r=1}^R c_{ir} (Y_r - Y_r^0) \quad (6.12)$$

donde

$$c_{ir} = \sum_{j=1}^n \frac{\delta a_{ij}}{\delta Y_r} X_j^0, \quad i=0,1,2,\dots,m \text{ y } r=1,2,\dots,R \quad (6.13)$$

Haciendo un cambio de variables tal que

$$d_{0r}^0 = c_{0r} + \frac{\delta b_0}{\delta Y_r} \quad (6.14)$$

$$d_{ir}^0 = c_{ir} - \frac{\delta b_i}{\delta Y_r}, \quad i=1,2,\dots,m \quad (6.15)$$

$$R_0^0 = b_0^0 - \sum_{r=1}^R d_{0r}^0 Y_r^0 \quad (6.16)$$

$$R_i^0 = b_i^0 + \sum_{r=1}^R d_{ir}^0 Y_r^0, \quad i=1,2,\dots,m \quad (6.17)$$

el problema linealizado será

$$\min Z = R_0^0 + \sum_{j=1}^n a_{0j}^0 X_j + \sum_{r=1}^R d_{0r}^0 Y_r \quad (6.18)$$

sujeto a

$$\sum_{j=1}^n a_{ij}^0 X_j + \sum_{r=1}^R d_{ir}^0 Y_r = R_i^0, \quad i=1,2,\dots,m \quad (6.19)$$

$$0 \leq X_j \leq U_j^X, \quad j=1,2,\dots,n \quad (6.20)$$

$$\max (L_r, Y_r^0 - t_r) \leq Y_r \leq \min (U_r, Y_r^0 + t_r) \quad r=1,2,\dots,R \quad (6.21)$$

de estructura similar el problema (6.8) a (6.10).

Se denominan variables independientes

$$Y^I = (Y_1^I, Y_2^I, \dots, Y_j^I, \dots, Y_I^I)^t$$

a aquellas variables no-lineales que en el óptimo del problema (6.18) a (6.21) sean no-básicas (por tanto, su valor óptimo estará en uno de los límites definidos por (6.21)) y su valor óptimo difiera en más de t_r de sus límites L_r ó U_r . Por tanto, si el valor óptimo de Y en el desplazamiento es Y_r^e , será variable independiente si

$$|Y_r^e - Y_r^0| = t_r, \quad Y_r^e - L_r > t_r \text{ y } Y_r^e - Y_r > t_r.$$

Las otras variables Y_r se considerarán simplemente variables no-lineales no-básicas. Considerando que, al estar estas otras variables muy próximas a sus límites y haberse optimizado en esa dirección, no merece la pena intentar una optimización sobre su posible desplazamiento, el problema se centra en el

desplazamiento de las variables independientes.

Para que la operativa del algoritmo CGAP converja hacia un óptimo local es preciso que el problema (6.1) a (6.4) cumpla las siguientes condiciones:

- a) Las funciones $a_{ij}(Y)$ y $b_i(Y)$, $i=0,1,2,\dots,m$ y $j=1,2,\dots,n$ deben ser continuas y diferenciables respecto a Y .
- b) U_j^X debe ser finito para los valores de j tal que a_{ij} no sea constante y dependa de Y .
- c) Si se linealiza el problema (6.1) a (6.4) alrededor de una solución factible (e.g. X^0, Y^0) y se crea el problema (6.18) a (6.21), este problema debe tener también una solución factible.

El conjunto de variables independientes Y^I permanece constante en toda la iteración k , salvo que la variable e.g. Y_j^I tome en el óptimo de la iteración l en el paso de determinación de la amplitud del desplazamiento un valor tal que su diferencia con respecto a L_j ó U_j sea no mayor que t_j (tolerancia que inmediatamente después del paso exploratorio ya se habrá reducido).

El siguiente paso al paso exploratorio consiste en reducir automáticamente las tolerancias t_r de tal forma que en cada iteración k se vaya obteniendo una solución más precisa.

6.3 Dirección del desplazamiento de las variables independientes

El paso 3 (base step) de la operativa del algoritmo CGAP tiene por objetivo obtener el gradiente reducido $\lambda^{I,k}$ de las variables independientes, cuya descripción teórica es la recogida en las secciones anteriores.

Para ello se fija el vector independiente Y^I a su valor óptimo obtenido en el paso inmediatamente anterior (paso exploratorio); los vectores Y^B e Y^N estarán limitados según (6.21) donde t_r ya se ha actualizado e Y responderá a los valores óptimos obtenidos en el paso exploratorio. Optimizando el subproblema (6.18) a (6.21) donde (X^0, Y^0) es el

óptimo del paso exploratorio, se obtiene el gradiente reducido $\lambda^{I,k}$ (vector fila de orden I) de una forma similar a (4.1). A los efectos de la dirección del desplazamiento se considera que son cero los gradientes reducidos $\lambda_X^{N,k}$ y $\lambda_Y^{N,k}$, respectivamente, de las variables no-básicas lineales y no-lineales.

La dirección $Z^{I,k+1}$ del desplazamiento de las variables independientes tiene tres formulaciones diferentes.

- a) Dirección de retorno. Siendo Y_0^I e Y_1^I los vectores alrededor de los cuales se ha realizado el problema (6.1) a (6.4), respectivamente, en los exploratory y base steps, el escalar f_D será:

$$f_D = \lambda^{I,k}(Y_0^I - Y_1^I) \quad (6.22)$$

tal que si $f_D < 0$ significa que la dirección del desplazamiento debe volver hacia Y_0^I , es decir hacia la solución de partida del paso exploratorio. Por tanto,

$$Z^{I,k+1} = s (Y_0^I - Y_1^I) \quad (6.23)$$

donde $Z^{I,k+1}$, s , Y_0^I e Y_1^I son vectores columna de orden I . El vector de normalización o escala s evita que debido a una serie de circunstancias, entre otras las diferentes unidades de escala en las variables, la dirección del desplazamiento pueda variar erráticamente. El valor que mejor resultado ha dado es:

$$s_j = 1/\sigma_j^2, \quad j=1,2,\dots,I \quad (6.24)$$

donde σ_j^2 recoge la varianza muestral del reduced cost λ_j^I de la variable Y_j^I , obtenida en los sucesivos pasos del algoritmo. La operativa consiste en reinicializar σ_j^2 en cada iteración k , inmediatamente después del paso exploratorio y actualizar su valor cada vez que se obtenga un gradiente reducido λ^I en la misma iteración k . Batchelor y Beale, /6/, describen las formulaciones. Ver también Beale, /7/, y para una descripción clara del problema ver Beale /8/.

- b) Si $f_D \geq 0$ ó en cualquier caso en las siguientes iteraciones a la primera (dentro de la iteración k) en la obtención de una nueva dirección de desplazamiento para el

mismo conjunto Y^I de variables independientes (excepto aquellas que se hayan -- eliminado tal como se ha indicado), la dirección $Z^{I,k+1}$ se obtiene a base del negativo del gradiente reducido normalizado

$$Z^{I,k+1} = -s (\lambda^{I,k})^t \quad (6.25)$$

donde $\lambda^{I,k}$ es un vector fila, tal que

$$Z_j^{I,k+1} = -s_j \lambda_j^{I,k}$$

o del gradiente reducido conjugado

$$Z^{I,k+1} = -s (\lambda^{I,k}) + \beta^k Z^{I,k} \quad (6.26)$$

donde

$$\beta^k = (s^t \lambda^{I,k}) (\Delta d^I)^t / \Delta d^I Z^{I,k} \quad (6.27)$$

tal que Δd^I es un vector fila que recoge la diferencia en el gradiente reducido de los dos últimos puntos (X, Y) obtenidos. Para la primera búsqueda de la dirección $Z^{I,k+1}$ en la iteración k , Δd^I será:

$$\Delta d^I = \lambda^{I,k} - \lambda^{I,k-1} \quad (6.28)$$

En este caso, β^k (6.27) es bastante similar a la formulación (4.35) debida a Perry /48/.

Batchelor y Beale, /6/, recogen las condiciones bajo las cuales se aplican las formulaciones (6.25) ó (6.26). En general, -- se puede indicar que si en la iteración -- anterior (k-1) o en los pasos previos de obtención de la búsqueda de la dirección $Z^{I,k+1}$, el conjunto Y^I no es lo mismo, se aplica (6.25) y no (6.26), o lo que es lo mismo $\beta^k=0$.

6.4 Amplitud del desplazamiento

Los vectores no-básicos X^N e Y^N tienen una -- dirección de desplazamiento nula. El nuevo -- valor de las variables independientes será:

$$Y^{I,k+1} = Y^{I,k} + \alpha_{k+1}^* Z^{I,k+1} \quad (6.29)$$

para la iteración k+1, donde α_{k+1}^* es la amplitud del desplazamiento. Ahora bien, es posible que en el paso 5 haya que obtener -- $Y^{I,k+1}$ a base de iteraciones sucesivas modificando la dirección $Z^{I,k+1}$. De todas formas,

el valor óptimo α_{k+1}^* se obtiene por pasos intermedios. Para clarificar nomenclatura denominaremos simplemente α y α_A , respectivamente, a los valores intermedios de α_{k+1}^* a calcular y previamente calculado, y $Z^{I,k+1}$ a la dirección que se esté contemplando. Y_A^I será de forma análoga el vector de los últimos valores obtenidos para el vector de variables Y^I . De forma análoga X_A^B , Y_A^B , X_A^N e Y_A^N .

El valor máximo α_M para el que todavía las -- variables Y^I seguirían siendo independientes será:

$$\alpha_M = \min \left\{ \min_{(j)} \left\{ \frac{L_j + t_j - Y_{Aj}^I}{Z_j^I}, Z_j^I < 0 \right\}, \min_{(j)} \left\{ \frac{U_j - t_j - Y_{Aj}^I}{Z_j^I}, Z_j^I > 0 \right\} \right\} \quad (6.30)$$

donde t_j está actualizada.

Por otra parte, si T:

$$T = \min_{(j)} (t_j / Z_j^I) \quad (6.31)$$

recogiera la variación $(\alpha - \alpha_A)$, significa que alguna variable independiente cambia su valor de Y_A^I a Y^I con una diferencia igual a su tolerancia y las demás con una diferencia menor.

El criterio seguido por el algoritmo CGAP para obtener el incremento $(\alpha - \alpha_k)$ en la amplitud del desplazamiento de las variables -- independientes consiste en exigir que al menos haya una variable independiente en la -- que el desplazamiento sea como mínimo el de su tolerancia, aunque para ello tuviera que dejar de ser independiente.

Por tanto,

$$\alpha - \alpha_A = \max \{ (\alpha_M - \alpha_A), T \} \quad (6.32)$$

De donde,

$$Y^I = Y_A^I + (\alpha - \alpha_A) Z^I \quad (6.33)$$

siendo

$$Y^N = Y_A^N \quad \text{y} \quad X^N = X_A^N.$$

Los vectores Y^B y X^B se obtienen en base al concepto de directional derivatives. Así la de la función objetivo será:

$$\frac{d f}{d \alpha} = - \sum_{j=1}^I z_j^I \lambda_j^I \quad (6.34)$$

que recoge la disminución de la función objetivo debida al incremento unitario de la amplitud $(\alpha - \alpha_A)$, según (6.33). La correspondiente a la variable lineal básica X_i^B dependerá de la fila i del último cuadro simplex de la optimización del subproblema inmediatamente anterior.

$$\begin{aligned} X_{Ai}^B &= \bar{a}_{i0} + \sum_{j=1}^N \bar{a}_{ij} (-X_{Aj}^N) + \\ &+ \sum_{j=1}^N \bar{a}_{ij} (-Y_{Aj}^N) + \\ &+ \sum_{j=1}^I \bar{a}_{ij} (-Y_j^I) \end{aligned} \quad (6.35)$$

de donde

$$\frac{d X_i^B}{d \alpha} = - \sum_{j=1}^I \bar{a}_{ij} z_j^I, \quad i=1,2,\dots,B \quad (6.36)$$

que recoge la variación en el valor X_{Ai}^B debida al incremento unitario de la amplitud $(\alpha - \alpha_A)$. De forma análoga,

$$\frac{d Y_i^B}{d \alpha} = - \sum_{j=1}^I \bar{a}_{ij} z_j^I \quad (6.37)$$

Si Y^I queda definido por (6.33),

$$Y^N = Y_A^N \quad y \quad X^N = X_A^N,$$

el resto de las variables será:

$$Y_i^B = \max \left\{ L_i, \min \left\{ U_i, \left(Y_{Ai}^B + (\alpha - \alpha_A) \frac{d Y_i^B}{d \alpha} \right) \right\} \right\} \quad (6.38)$$

$$X_i^B = \max \left\{ 0, \min \left\{ U_i^X, \left(X_{Ai}^B + (\alpha - \alpha_A) \frac{d X_i^B}{d \alpha} \right) \right\} \right\} \quad (6.39)$$

Es decir, Y^B y X^B tomarán su valor respectivamente en las direcciones (6.37) y (6.36), estando restringido por sus límites corres-

pondientes.

El punto (X, Y) calculado según (6.33), (6.38) y (6.39) sirve como punto de linealización para el paso 5c. En esta linealización se fijan las variables independientes al vector $-Y^I$ (6.33). Batchelor y Beale /6/ y Beale /8/ describen las condiciones que, basadas en la directional derivative de la función objetivo y en el nuevo gradiente de las variables independientes en el óptimo del subproblema inmediatamente optimizado, determinan que la operativa del algoritmo CGAP:

- Continue en la misma dirección modificando la amplitud.
- Investigue una nueva dirección de las -- que todavía sean variables independientes.
- Determine las nuevas variables independientes en el paso exploratorio de la -- iteración siguiente $(k+1)$.
- Obtenga una nueva amplitud en la misma -- dirección interpolando las anteriores, -- bien a base de una aproximación cúbica -- descrita por Beale /8/.
- Termine el problema si se cumplen los -- criterios standard de terminación descritos en las secciones anteriores.

El algoritmo CGAP incorpora por primera vez a un sistema general de PL las técnicas de optimización no-lineal; está siendo extensamente validado en grandes problemas reales -- fundamentalmente en el campo petrolífero.

7. CONCLUSION

Los algoritmos de programación no-lineal basados en el método simplex de PL tienen, como consecuencia, la característica en común de aproximar las funciones no-lineales: objetivo y de condiciones desarrollándolas en series de Taylor hasta el primer término (linealización) en las condiciones y algún algoritmo (GRG2, MINOS, LSGRG) incluso hasta el segundo término en la función objetivo.

Resuelven el problema a base de optimizar -- iterativamente el subproblema lineal. Se ba-

san en diversas variantes del gradiente reducido para desplazar las variables no-básicas unas veces, superbásicas otras e independientes en el resto. La técnica del gradiente reducido conjugado es la que mejores resultados está proporcionando. En base al desplazamiento de estas variables, se obtiene el correspondiente a las variables básicas o dependientes e iterativamente se ajustan éstas para producir una solución factible que o es la óptima o sirve de base para la iteración-linealización siguiente.

Estos algoritmos obtienen el óptimo local -- sin garantizar que sea el óptimo global. Por tanto, en problemas no-convexos pueden producir alguna sorpresa, aunque en los casos reportados en la literatura el óptimo obtenido es muy aceptable. No obstante, no son algoritmos de programación convexa ya que no precisan esta condición para converger al óptimo, en este caso, local.

Los algoritmos descritos en este trabajo están siendo extensamente validados en problemas reales, aparte de los tradicionales problemas-test.

El futuro de la programación no-lineal con restricciones parece orientarse en el sentido de estos algoritmos, fundamentalmente utilizando el soporte de los sistemas generales de PL aprovechando las consiguientes innovaciones técnicas que se han producido en estos sistemas en los últimos años.

8. INVESTIGACION FUTURA

No obstante, en paralelo a estos algoritmos, hoy día se está trabajando en nuevos algoritmos cuya aplicabilidad no está totalmente demostrada pero, dada la filosofía en que se fundan, es posible que en un próximo futuro sean también algoritmos de gran aplicación. Estos son de dos tipos: algoritmos que optimizan una función no-lineal sujeta a condiciones lineales y algoritmos cuyo problema a optimizar también tiene condiciones no-lineales. En ambos tipos de algoritmos se abandona la técnica de reducción de variables y, por tanto, la técnica de optimización de un problema sin restricciones en el que se optimiza la función en base a las variables no-básicas, independientes o superbásicas.

El primer tipo de algoritmos está basado en los trabajos de Gill y Murray; los más significativos se han referenciado en este trabajo (en las referencias /29/ y /30/ está el resumen fundamental de los mismos). Las técnicas que desarrollan han sido utilizadas -- por los algoritmos GRG2, MINOS y LSGRG. Pero en un problema en el que la matriz de condiciones es del tipo $AX \geq b$, Gill y Murray prefieren utilizar la técnica de, partiendo de una solución factible \bar{X} , analizar las condiciones que son activas (aquellas en las que $A\bar{X}=b$), obtener la dirección del desplazamiento de las variables (ya no hay la distinción entre variables básicas y no-básicas) según el criterio elegido (basado en los algoritmos de Newton, quasi-Newton o gradiente reducido), obtener la amplitud máxima del desplazamiento según la condición inactiva que menor holgura tenga, obtener la amplitud óptima según un ajuste cúbico (con la técnica -- delinsearch debida a ellos) e incluir esta condición en el conjunto de condiciones activas si es preciso. Gill y Murray efectúan -- una actualización de la matriz ortogonal T, tanto al añadir una condición al conjunto de condiciones activas como el sacar alguna de ellas de dicho conjunto si ello redundaría en un menor valor en la función objetivo. En esta eliminación utilizan los multiplicadores de Lagrange estimados a base del gradiente de la función objetivo y de la matriz pseudo-inversa A^+ de la matriz traspuesta A^t de las condiciones activas. Tanto al ampliar como al reducir el conjunto de condiciones activas, actualizan sin necesidad de recalcular la matriz hessiana proyectada T^tGT , pero en forma factorizada LGL^t . Cuando utilizan el algoritmo de quasi-Newton para obtener la dirección de desplazamiento emplean la complementary DFP formula (Gill y Murray, /25/ y /26/ pp. 74-76) a base de actualizar con rango 2 la matriz LGL^t . Este conjunto de algoritmos lo recogen en el sistema NPL. Dicho sistema contiene además rutinas para la minimización de una función sin restricciones -- con una o varias variables, obteniendo analíticamente o no las primeras y/o segundas derivadas, utilizando los algoritmos de Newton, Newton modificado, quasi-Newton, gradiente reducido, etc.

El segundo tipo de algoritmos en los que probablemente se basará la investigación futura en este campo, es aquel en que la función --

objetivo y condiciones son no-lineales, pero abandonando no solo la técnica de reducción de variables, sino que además convierten el problema en una optimización sin restricciones penalizando la función objetivo, no solo convirtiendo ésta en función de Lagrange (a base de utilizar las condiciones afectadas - por un multiplicador de Lagrange), sino que incluso penalizan dicha función de Lagrange con una función que recoge la suma ponderada por un coeficiente penalizador del cuadrado de las condiciones del problema. Este tipo de algoritmos denominados sequential augmented lagrangian (SAL) evitan los problemas de inestabilidad de los algoritmos de penalización tipo SUMT (ver Fiacco y McCormick /18/ y Lootsma /41/) y evitan los problemas de -- los algoritmos de reducción de variables al no cumplirse, en una determinada iteración, las condiciones del problema dado que éstas (ver e.g. apartado 3.3) se han linealizado. El algoritmo SAL, a base de minimizar iterativamente problemas sin restricciones en los que va actualizando los multiplicadores de Lagrange (y si es el caso los coeficientes penalizadores), obtiene unos valores de las variables tal que si en el correspondiente problema de minimización además de cumplirse las condiciones necesarias y suficientes para un mínimo sin restricciones, también se cumplen las condiciones necesarias de Kuhn-Tucker para un mínimo con restricciones (cumplimiento de las condiciones del problema, gradiente cero en la función de Lagrange, -- multiplicadores de Lagrange no-negativos en las condiciones de desigualdad activas y nulos en las inactivas), se demuestra (Pierre y Lowe, /49/, teorema 4.3) que existen condiciones de suficiencia para el mínimo de $f(X)$ sujeto a las condiciones del problema. Hestenes /35/ y Powell /51/ introducen el algoritmo SAL para el caso de condiciones no-lineales de igualdad. Murray /45/, Fletcher /19/, Mangasarian /44/, Rockafellar /54/, Pierre y Lowe /49/ y, recientemente, Wright /63/ y Powell /53/ entre otros extienden la aplicación del algoritmo al caso de condiciones lineales y condiciones no-lineales de desigualdad. El sistema NPL, en la versión de Wright /63/, también recoge el algoritmo SAL. Pierre y Lowe /49/ describen con todo detalle el código LPNLP, pero los problemas resueltos que presentan son casos recogidos en la literatura, de muy poca dimensión y, por el momento, no está validado con problemas reales. Wright

/63/ presenta casos resueltos con el algoritmo SAL con $m=105$ condiciones (no-lineales) y $n=150$ variables.

La filosofía en que se basan los tipos de algoritmos brevemente referenciados en este -- apartado tienen el rigor matemático que no tienen los otros algoritmos recogidos en este trabajo, fundamentalmente por lo que respecta al tratamiento de las condiciones no-lineales. Posiblemente sea así, pero la potencia y grado de sofisticación que hoy día han alcanzado los métodos de reducción de variables (basados en el método simplex de PL) tanto en el almacenamiento y tratamiento de la matriz de condiciones, como en la selección del pivote y en las técnicas de factorización matricial LU y LGL^t entre otras innovaciones efectuadas en los últimos años, hacen que los algoritmos descritos en este trabajo sean los que hoy día se estén aplicando en la resolución de problemas reales. Es preciso una mayor validación de los algoritmos referenciados en este apartado para obtener una conclusión definitiva sobre los mismos.

9. REFERENCIAS

- /1/ ABADIE, J. "Application of the GRG algorithm to optimal control problems". -- J. ABADIE (ed). "Integer and non-linear programming". North-Holland, Amsterdam, 1970, pp. 191-211.
- /2/ ABADIE, J. y CARPENTIER, J. "Généralisation de la méthode du gradient réduit de Wolfe au cas des contraintes non-linéaires". D.B. HERTZ y J. MELESE (eds.). "Proceedings of IV IFORS Conference". Wiley Interscience, New York, 1966, pp. 1041--1053.
- /3/ ABADIE, J. y CARPENTIER, J. "Generalization of the Wolfe reduced gradient method to the case of non-linear constraints". R. FLETCHER (ed.). "Optimization". Academic Press, New York, 1969, pp. 37-47.
- /4/ ABADIE, J. y GUIGOU, J. "Numerical Experiments with the GRG method". J. ABADIE (ed.). "Integer and non-linear programming". North-Holland, Amsterdam, 1970, -- pp. 529-536.

- /5/ AVRIEL, M. "Non-linear programming". -- Prentice-Hall, Englewood Cliffs, N.Y., - 1976.
- /6/ BATCHELOR, A.S.J. y BEALE, E.M.L. "A revised method of conjugate gradient approximation programming". IX Mathematical - Programming Symposiu, Budapest, 1976.
- /7/ BEALE, E.M.L. "A Conjugate Gradient Method of Approximation Programming". R.W. COTTLE y J. KRARUP (eds.). "Optimization Methods". English Universities Press, -- Londres, 1974, pp. 261-277.
- /8/ BEALE, E.M.L. "Non-linear programming -- using a general mathematical programming system". Nato Summer School on Design -- and Implementation of Software Optimization, Urbino (Italia), 1977.
- /9/ BRODLIE, K.W. "Unconstrained minimization". D.A.H. JACOBS (ed.). "The State of the - Art in Numerical Analysis". Academic Press Londres, 1977, pp. 229-268.
- /10/ BROYDEN, C.G. "The convergence of a class of double-rank minimization algorithms: the new algorithm". J. Inst. Math. Applications 6, 1970, pp. 222-231.
- /11/ COHEN, C. "Generalized Reduced Gradient Technique for Non-Linear Programming". - User Writeup, Vogelback Computer Center, Northeastern University, 1974.
- /12/ COLVILLE, A.R. "A comparative study of - non-linear programming codes". IBM New - York Scientific Center, Tech. Rep., 1968, pp. 320-2949.
- /13/ DANTZIG, G.B. "Linear programming and extensions". Princeton, N.H., 1963.
- /14/ DAVIDON, W.C. "Variable-metric method -- for minimization". AEC Res. and Dev. Rep. ANL-5990, 1959.
- /15/ DRUD, A. "Optimization of large scale dynamic systems using the GRG-algorithm" - TMS XXIII, Atenas, 1977.
- /16/ ESCUDERO, L.F. "Panorámica actual de la programación matemática". Seminario de - Programación Matemática, Madrid, 1977.
- /17/ FAURE, P. y HUARD, P. "Résolution des - programmes mathématiques à fonction non linéaire par la méthode du gradient réduit". Revue Française de Recherche Opérationnelle 9, 1965, pp. 167-205.
- /18/ FIACCO, A.V. y McCORMICK, G.P. "Non-linear programming: Sequential Unconstrained Minimization Techniques". John Wiley, New York, 1968.
- /19/ FLETCHER, R. "A class of methods for -- nonlinear programming with termination and convergence properties". J. ABADIE (ed.). "Integer and nonlinear programming". North-Holland, Amsterdam, 1970a. pp. 157-175.
- /20/ FLETCHER, R. "A new approach to variable-metric algorithms". Computer Journal 13, 1970b, pp. 317-322.
- /21/ FLETCHER, R. y POWELL, M.J.D. "A rapidly convergent descent method for minimization". Computer Journal 6, 1963, pp. 163-168.
- /22/ FLETCHER, R. y REEVES, C.M. "Function - minimization by conjugate gradients". Computer Journal 7, 1964, pp. 149-154.
- /23/ FORREST, J.J.H. y TOMLIN, J.A. "Updated triangular factors of the basis to maintain sparsity in the product form simplex method". Mathematical Programming 2, 1972, pp. 262-278.
- /24/ FRANK, M. y WOLFE, P. "An algorithm for quadratic programming". Naval Research Logistics Quarterly 3, 1956, pp. 95-110.
- /25/ GILL, P.E. y MURRAY, W. "Quasi-Newton - methods for unconstrained optimization" J. Inst. Math. Applications 9, 1972, pp. 91-108.
- /26/ GILL, P.E. y MURRAY, W. "Numerical methods for constrained optimization". -- Academic Press, Londres, 1974a.
- /27/ GILL, P.E. y MURRAY, W. "New type methods for unconstrained and linearly -- constrained optimization". Mathematical Programming 7, 1974b, pp. 311-350.

- /28/ GILL, P.E. y MURRAY, W. "Safeguard step-length algorithms for optimization -- using descent methods". Rep. NAC 37, -- National Physical Laboratory, Teddington, 1974c.
- /29/ GILL, P.E. y MURRAY, W. "Linearly constrained problems including linear and quadratic programming". D.A.H. JACOBS (ed.). "The State of the Art in Numerical Analysis". Academic Press, Londres, 1977a, pp. 313-363.
- /30/ GILL, P.E. y MURRAY, W. "Modification of matrix factorizations after a range-one change". D.A.H. JACOBS (ed.). "The State of the Art in Numerical Analysis". Academic Press, Londres, 1977b, pp. 55-83.
- /31/ GOLDFARB, D. "A family of variable-metric methods derived by variational -- means". Mathematics of Computation 24, 1970, pp. 23-26.
- /32/ GRIFFITH, R.E. y STEWART, R.A. "A non-linear programming technique for the optimization of continuous processing systems". Management Science 7, 1961, pp. 379-392.
- /33/ HELLERMAN, E. y RARICK, D.C. "The partitioned preassigned pivot procedure". -- D.J. ROSE y R.A. WILLOUGHBY (eds.). -- "Sparse matrices and their applications" Plenum Press, New York, 1972, pp. 67-76.
- /34/ HELTNE, D.R. y LIITSCHWAGER, J.M. "Users guide for GRG 73 y Technical Appendices to GRG 73". College of Engineering, Universidad de Iowa, 1973.
- /35/ HESTENES, M.R. "Multiplier and gradient methods". Journal of Optimization Theory and Application 4, 1969, pp. 303-320.
- /36/ HIMMEMBLAU, D. "Applied non-linear programming". McGraw-Hill, New York, 1972.
- /37/ IBM. "MPSX: Mathematical Programming -- System Extended/370". IBM, PP 5740-XM3, New York, 1974.
- /38/ JENNINGS, A. "Matrix Computation for Engineering and Scientists". John Wiley, New York, 1977.
- /39/ KALAN, J.E. "Aspects of large-scale in-core linear programming". Proceedings -- of the 1971 Annual Conference of the -- ACM, Chicago, 1971, pp. 304-313.
- /40/ LASDON, L.S. y WARREN, A.D. "Generalized reduced gradient software for linearly and nonlinearly constrained problems" Nato Summer School on Design and Implementation of Optimization Software, Urbino (Italia), 1977.
- /41/ LOOSTMA, F.A. "A survey of methods for solving constrained minimization problems via unconstrained minimization". F. A. LOOSTMA (ed.). "Numerical methods for nonlinear optimization". Academic Press, Londres, 1972, pp. 313-348.
- /42/ LOOSTMA, F.A. "Introduction to non-linear optimization". Part I - 1977 y -- Part II - 1978.
- /43/ MANGASARIAN, O.L. "Non-linear programming". McGraw-Hill, New York, 1969.
- /44/ MANGASARIAN, O.L. "Unconstrained lagrangians in nonlinear programming". Computer Sciences Technical Report 174, Universidad de Wisconsin, Madison, 1973.
- /45/ MURRAY, W. "An algorithm for constrained minimization". R. FLETCHER (ed.). "Optimization". Academic Press, New York, -- 1969, pp. 247-258.
- /46/ MURTAGH, B.A. y SAUNDERS, M.A. "Non-linear programming for large sparse systems". Systems Optimization Laboratory, Tech. Rep. SOL 76-15, Universidad de -- Stanford, California, 1976.
- /47/ MURTAGH, B.A. y SAUNDERS, M.A. "Large scale linearly constrained optimization" Mathematical Programming 14, 1978, pp. 41-72.
- /48/ PERRY, A. "An improved conjugate gradient algorithm". Nota técnica. Dept. -- de Decision Sciences, Graduate School -- of Management, Universidad de Northwestern, Evanston, Illinois, 1976.
- /49/ PIERRE, D.A. y LOWE, M.J. "Mathematical programming via augmented lagrangians: an introduction with computer programs".

Addison-Wesley, Londres, 1975.

- /50/ POLAK, E. "Computational methods in optimization: a unified approach". Academic Press, 1971.
- /51/ POWELL, M.J.D. "A method for nonlinear constraints in minimization problems". R. FLETCHER (ed.). "Optimization". Academic Press, New York, 1969, pp. 283-298.
- /52/ POWELL, M.J.D. "Restart procedures for the conjugate gradient method". Mathematical programming 12, 1977, pp. 241-254.
- /53/ POWELL, M.J.D. "Algorithms for nonlinear constraints that use lagrangian functions". Mathematical Programming 14, -- 1978, pp. 224-248.
- /54/ ROCKAFELLAR, R.T. "A dual approach to solving nonlinear programming problems by unconstrained optimization". Mathematical Programming 5, 1973, pp. 354-373.
- /55/ SCICON. "Users guide to Sciconic (3.2)". Scicon Computers Services Ltd., Milton Keynes, U.K., 1976.
- /56/ SHANNO, D.F. "Conditions of quasi-newton methods for function minimization". Mathematics of Computation 24, 1970, - pp. 657-667.
- /57/ WILDE, D.J. "Optimum seeking methods" - Prentice-Hall, Englewoods Cliffs, New York, 1964.
- /58/ WILDE, D.J. y BEIGHTLER, C.S. "Foundations of Optimization". Prentice-Hall, Englewood Cliffs, New York, 1967.
- /59/ WILKINSON, J.H. "The algebraic eigenvalue problem". Oxford University Press, Londres, 1965.
- /60/ WOLFE, P. "Methods of non-linear programming". R.L. GRAVES y P. WOLFE (eds.) "Recent advances in mathematical programming". McGraw-Hill, New York, 1963. pp. 76-77.
- /61/ WOLFE, P. "Methods for linear constraints" J. ABADIE (ed.). "Non-linear programming" North-Holland, Amsterdam, 1967, pp.120-124
- /62/ WOLFE, P. "Convergence theory in non-linear programming". J. ABADIE (ed.). "Integer and non-linear programming". North Holland, Amsterdam, 1970, pp. 1-36.
- /63/ WRIGHT, M.H. "Sequential Augmented Lagrangian algorithm". Tesis doctoral, -- Universidad de Stanford, California, -- 1977.
- /64/ ZANGWILL, W.I. "Non-linear programming: an unified approach". Prentice-Hall, Englewoods Cliffs, New York, 1969.
- /65/ ZOUTENDIJK, G. "Non-linear programming: computational methods". J. ABADIE (ed.). "Integer and non-linear programming". - North-Holland, Amsterdam, 1970, pp. 38-86.
- /66/ ZOUTENDIJK, G. "Mathematical programming methods". North-Holland, Amsterdam, 1976.

