5
2y

UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE CIENCIAS

Programación Cuadrática
Comparación de Dos Métodos

T e s i s

Que para obtener el título de

Matemático

Presenta

Emmanuel Berriel Valdos

México, D. F. 1991

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

C O N T E N I D O

INTRODUCCION

Los métodos más eficientes para resolver el problema de programación no lineal, aproximan la solución del problema mediante una secuencia de problemas cuadráticos. Es por eso que es muy importante disponer de algoritmos de programación cuadrática que sean lo mas eficientes posible.

De entre los algoritmos que existen sobre programación cuadrática, se han escogido dos para compararse numericamente. Uno de ellos fue desarrollado por E.C.Lemke [11] y el otro fue una aportación del trabajo conjunto realizado por D.Goldfarb y A.Idnani [10]. Una característica común de ambos algoritmos, es que son duales, sin embargo el algoritmo de Lemke considera problemas de programación cuadrática positiva semidefinida, mientras que el algoritmo dado por Goldfarb e Idnani es para problemas de programación cuadrática estrictamente convexos.

Como se sabe, el algoritmo de Lemke, es usado para resolver el problema lineal complementario, el cual consiste en encontrar los vectores w & $z \in \mathbb{R}^n$ tales que satisfagan el sistema

$$w = Mz + q \qquad (a)$$
$$w \geq 0 \ , \ z \geq 0 \qquad (b)$$
$$w^t z = 0 \qquad (c)$$

para un vector $q \in \mathbb{R}^n$ y una matriz $M \in \mathbb{R}^{nxn}$ dados.

Al aplicar el algoritmo de Lemke para encontrar los vectores w & z, se transforma en cada iteración el sistema (a), lo que es muy costoso. R. W. H Sargent [16], implementa el algoritmo de Lemke de forma tal que al pasar de una iteración a otra, no se hace la transformación de todo el sistema sino solo de una parte de él, como veremos en el capítulo II, lo que lo hace menos costoso.

El presente trabajo, hace una comparación numérica de la implementación que hace Sargent del algoritmo de Lemke con el

algoritmo dado por Goldfarb e Idnani mediante la utilización de problemas cuadráticos convexos generados alcatoriamente por el método de Rosen y Suzuki [15].

La razón por la cual se han elegido estos dos algoritmos, se debe a que los resultados numéricos han mostrado que los algoritmos duales son superiores a los algoritmos primales, cuando no se dispone inmediatamente de un punto factible.

Antes de dar la descripción de cada uno de los dos algoritmos, en el capítulo I se dan algunas generalidades sobre programación lineal, dualidad y programación cuadrática y en los siguientes 2 capítulos se hace la descripción de los 2 algoritmos a comparar. Así, en el capítulo II se describe el algoritmo de Lemke y la implementación que de éste hace Sargent, y en el capítulo III, se trata el algoritmo de Goldfarb. Por último, en el capítulo IV se presentan los resultados numéricos y en el V se dan las conclusiones.

CAPITULO

I

PROGRAMACION LINEAL Y EL METODO SIMPLEX

Se dá a continuación, una descripción del problema lineal, del método simplex y del problema dual, ya que están ligados fuertemente al método de solución para el problema cuadrático dado por Lemke que se describirá en el capítulo II. Es más, las condiciones de optimalidad del problema lineal, son un caso especial de las condiciones de optimalidad del método para el problema cuadrático, dado por Lemke.

Considérese el siguiente problema:

$$\text{minimizar} \quad f(x)=c^t x \tag{1.1.1}$$

$$\text{sujeto} \quad Ax=b \tag{1.1.2}$$

$$x\geq 0 \tag{1.1.3}$$

Donde c, $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{mxn}$ y $b \in \mathbb{R}^m$.

A cualquier problema de la forma (1.1), se le llama un problema de programación lineal en forma estandar, a la función f se le llama función objetivo, (1.1.2) son las restricciones del problema y (1.1.3) son las cotas de las variables del problema.

Frecuentemente las restricciones a que está sujeta la función objetivo, son desigualdades, así que el problema de programación lineal quedaría como:

$$\text{minimizar } f(x)=c^t x \tag{1.2.1}$$

$$\text{sujeto} \quad Ax\leq b \tag{1.2.2}$$

$$x\geq 0 \tag{1.2.3}$$

O bién como:

$$\text{minimizar } f(x)=c^t x \qquad (1.3.1)$$

$$\text{sujeto } Ax \geq b \qquad (1.3.2)$$

$$x \geq 0 \qquad (1.3.2)$$

ó bién como una combinación de (1.2) y (1.3).

Para poner un problema de programación lineal, sujeto a desigualdades en forma estandar, se usan las llamadas variables de holgura, de forma tal que si las usamos en el problema (1.2), este quedaría como sigue:

$$\begin{aligned} &\text{minimizar } f(x)=c^t x \\ &\text{sujeto a } Ax+v=b \\ &\text{con } x,\ v \geq 0 \end{aligned} \qquad (P)$$

donde, $v \in \mathbb{R}^m$ es el vector de las llamadas variables de holgura. De aquí que cualquier problema de programación lineal con desigualdades, se puede poner en forma estandar.

Supóngase que el problema de programación lineal a resolver es:

$$\text{minimizar } f(x)=c^t x \qquad (1.4.1)$$

$$\text{sujeto a } Ax \leq b \qquad (1.4.2)$$

$$x \geq 0. \qquad (1.4.3)$$

Geometricamente, cada una de las desigualdades de (1.4), representa un semiespacio en $\mathbb{R}^n$, y es la intersección de estos semiespacios lo que nos produce el llamado conjunto factible, el cual, es un politopo en $\mathbb{R}^n$. Si este politopo es acotado, diremos

que el conjunto factible es un poliédro; es más, si un vector x satisface las desigualdades de (1.4), entonces pertenece al conjunto factible y diremos que x es un punto o una solucion factible del problema (1.4), al agregarle al problema (1.4) las variables de holgura, se obtiene el problema P (pag. 4) y vamos a suponer que el número de variables de P en total es n.

Sea x una solución del problema P. Es claro, que x debe ser una solución factible. Si m de las n variables (m≤n),del problema P son tales que las columnas de la nueva matríz A asociadas con esas variables son linealmente independientes, y las restantes (n-m) variables son cero, se dice entonces que x es una solución básica y las m variables son llamadas variables básicas. Se entiende que no pueden existir mas de m variables básicas ya que solo hay m ecuaciones de restricción.

Geometricamente, cualquier solución básica de P (pag. 4) corresponde a una de las esquinas del politopo generado por las restricciones de P y el mínimo de la función f, si es que existe, está en una de las esquinas del politopo; esto es, el mínimo de f es una solución básica.(ver figura)

El simplex, que es el método mas usado para resolver problemas de programación lineal, consiste de dos fases; la primera de ellas, se emplea para encontrar una solución básica del problema, si es que no se dispone inmediatamente de una de ellas. Una vez que ya se tiene una primera solución básica, la cual corresponde a uno de los vértices del politopo, la segunda fase nos conduce vértice a vértice, es decir , de solución básica en

solución básica, de forma tal que f decrece en cada nuevo vértice encontrado hasta que en un vértice x^* f ya no puede descender mas y este vértice x^* es el punto optimo buscado. Podemos describir el método simplex como sigue:

Supóngase que ya se tiene una solución básica x del problema P, dada o no por la primera fase del simplex. Como x debe tener m variables básicas, entonces x se puede partir como $x=(x_B,x_D)^t$, donde x_B está formado por las m variables básicas y las restantes (n-m) variables contenidas en x_D, son cero. Igualmente el vector c se puede poner como $c=(c_B,c_D)^t$ y con las primeras m columnas de la matriz A (las asociadas a las variables básicas) se puede formar una matriz cuadrada B no singular, de forma tal que, la matriz A se puede partir como $A=[B|D]$.

Esto nos permite poner el problema P (pag. 4) a resolver, en forma matricial como sigue:

$$\text{minimizar } f(x)=\begin{pmatrix}c_B\\c_D\end{pmatrix}^t\begin{pmatrix}x_B\\x_D\end{pmatrix} \qquad (1.5.1)$$

$$\text{sujeto a } [B|D]\begin{pmatrix}x_B\\x_D\end{pmatrix}=b \qquad (1.5.2)$$

$$x\geq 0 \qquad (1.5.3)$$

Puesto que la matriz B no es singular, multiplicando (1.5.2) por B^{-1}, resulta que:

$$[\ I\ |B^{-1}D]\begin{pmatrix}x_B\\x_D\end{pmatrix}=B^{-1}b. \qquad (1.6.1)$$

Despejando x_B de (1.6.1) y sustituyendo x_B en (1.5.1), se tiene :

$$f(x)=c_B B^{-1}b+(c_D-c_B B^{-1}D)x_D \qquad (1.6.2)$$

como las variables no básicas contenidas en x_D son cero, el valor de las variables básicas en este momento es x_B, y $f=c_B B^{-1}b$.

Si llamamos $r=(c_D-c_B B^{-1}D)$ entonces (1.5) se puede poner como:

$$\text{minimizar } f(x)=c_B B^{-1}b+rx_D \qquad (1.7.1)$$

$$\text{sujeto a} \quad [\, I \mid B^{-1}D \,] \begin{pmatrix} x_B \\ x_D \end{pmatrix} = B^{-1}b \qquad (1.7.2)$$

$$x \geq 0 \qquad (1.7.3)$$

Si alguna de las componentes de r es negativa, digamos la componente k, podemos aumentar el valor de la variable no básica x_k desde cero, lo cual permite que f disminuya su valor a medida que x_k aumente el suyo. Como nos interesa que f decrezca lo mas rapidamente posible, supóngase que r_k es la más negativa de todas las componentes de r, entonces, un aumento en x_k producirá el mayor decrecimiento posible en f.

Pongamos (1.7.2) de la siguiente forma:

$$\left(\begin{array}{cccc|ccccc} 1 & 0 & \cdots & 0 & a'_{1m+1} & \cdots & a'_{1k} & \cdots & a'_{1n} \\ 0 & 1 & & 0 & a'_{2m+1} & \cdots & a'_{2k} & \cdots & a'_{2n} \\ - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - \\ 0 & 0 & \cdots & 1 & a'_{mm+1} & \cdots & a'_{mk} & \cdots & a'_{mn} \end{array}\right) \left(\begin{array}{c} x_B \\ \hline x_D \end{array}\right) = \begin{pmatrix} b'_1 \\ b'_2 \\ -- \\ -- \\ b'_m \end{pmatrix} \qquad (1.8)$$

Puesto que las variables no básicas diferentes de x_k permanecen como tales, su valor seguirá siendo cero y entonces, (1.8) se puede poner como:

$$\begin{aligned} x_1+a'_{1k}x_k &= b'_1 \\ x_2+a'_{2k}x_k &= b'_2 \\ &\cdots\cdots \\ &\cdots\cdots \\ &\cdots\cdots \\ x_m+a'_{mk}x_k &= b'_m \end{aligned} \qquad (1.9)$$

, donde, $(x_1 \ldots x_m)=x_B^t$ y $x_k \in x_D$

despejando cada una de las de las x_i básicas de (1.9) tenemos

$$\begin{aligned} x_1 &= b'_1-a'_{1k}x_k \\ x_2 &= b'_2-a'_{2k}x_k \\ &\cdots\cdots\cdots \\ &\cdots\cdots\cdots \\ &\cdots\cdots\cdots \\ x_m &= b'_m-a'_{mk}x_k \end{aligned} \qquad (1.10)$$

Al aumentar el valor de x_k sucederá que las variables básicas x_i de (1.10), aumentarán, disminuirán o quedarán igual de acuerdo a que las $a'_{i,k}$ correspondientes sean positivas, negativas o cero. Si todas las $a'_{i,k}$ son no positivas, al aumentar x_k su valor, las x_i aumentarán o quedan igual, de acuerdo a que la $a'_{i,k}$ correspondiente sea negativa o cero, lo que se desea, ya que las variables deben ser no negativas. Pero como el coeficiente r_k correspondiente a x_k es negativo, sucede entonces que f(x) de (1.7.1) disminuye su valor a medida que x_k aumente, así que f(x) decrecerá sin límite y no tenemos solución finita.

Supóngase que algunas $a'_{i,k}$ son positivas, entonces al aumentar en (1.10) el valor de x_k desde cero, las correspondientes variables básicas x_i disminuyen su valor y llegará un momento en que al menos una de esas x_i , digamos x_j , se hace cero . Esto es; x_j bloquea el crecimiento de x_k , ya que si se sigue aumentando el valor de x_k , x_j se hace negativa y nos saldríamos del conjunto factible. Es por esto que a x_j se le llama variable de bloqueo mientras que a x_k , se le llama variable de conducción.

Se observa que al haberse hecho cero la variable básica x_j , si despejamos x_k de la ecuación correspondiente, se tendrá que $x_k=b'_j/a'_{jk}$. El cociente b'_j/a'_{jk}, es el mínimo de todos los cocientes b'_i/a'_{ik} , $\forall$ i=1,...,m , en los que $a'_{ik}>0$.

Al haber tomado x_k un valor positivo y x_j hacerse cero, no puede suceder que x_k siga siendo no básica, ya que es positiva, así que x_k se hace básica. Pero entonces, habrá m+1 variables en la base, lo que no es posible, por lo que una de las variables básicas debe dejar de serlo y esta debe ser necesariamente la variable de bloqueo x_j que se hizo cero.

Esto nos permite dar criterios para determinar qué variable va a volverse básica y cual va a dejar de serlo; en el primer caso, se va a volver básica aquella variable x_k, no básica, cuyo coeficiente, r_k, sea el mas negativo; para el segundo caso, vamos a escoger la variable básica x_j cuyo coeficiente a'_{jk} sea positivo y con la característica de que el cociente b'_j/a'_{jk} sea el mínimo. Esto es, el índice j de la variable que va a dejar de ser básica se escoje como:

$$j = \min_i \left\{ b'_i/a'_{ik} \ , \ \forall \ a'_{ik}>0 \right\} \qquad (1.11)$$

y el nuevo valor de x_k es igual a dicho cociente mínimo. Sustituyendo el valor de x_k en cada x_i básica de (1.10), se tiene que, el nuevo valor de cada x_i básica es:

$$\begin{aligned} x_i &= b'_i-(a'_{ik}b'_j)/a'_{jk} \\ &= (a'_{jk}b'_i-a'_{ik}b'_j)/a'_{jk} \ . \end{aligned}$$

Puesto que $a'_{jk}>0$ y $a'_{jk}b'_i \geq a'_{ik}b'_j$, entonces x_i permanece no negativa para toda x_i básica diferente de x_j. De esta forma al actualizar las variables básicas se conserva su no negatividad y f decrece, lo que se desea. Sin embargo, esto mismo se puede hacer realizando una operación de pivoteo usando como pivote el elemento a'_{jk} en el sistema (1.8).

Para eso, se tiene que (1.8) se puede poner como:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & a'_{1m+1} & \cdots & a'_{1k} & \cdots & a'_{1n} \\ 0 & 1 & \cdots & 0 & a'_{2m+1} & \cdots & a'_{2k} & \cdots & a'_{2n} \\ & & & & & & \vdots & & \\ & & & & & & a'_{jk} & & \\ & & & & & & \vdots & & \\ 0 & 0 & & 1 & a'_{mm+1} & \cdots & a'_{mk} & \cdots & a'_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ \vdots \\ \\ \vdots \\ b'_m \end{pmatrix} \qquad (1.13)$$

tomando como pivote al elememto a'_{jk} , (1.13) se transforma en:

$$\begin{pmatrix} 1 & 0 & \cdots & -a'_{1k}/a'_{jk} & \cdots & 0 & a''_{1m+1} & \cdots & 0 & \cdots & a''_{1n} \\ 0 & 1 & \cdots & -a'_{2k}/a'_{jk} & \cdots & 0 & a''_{2m+1} & \cdots & 0 & \cdots & a''_{2n} \\ & & & \vdots & & & & & \vdots & & \\ 0 & 0 & & 1/a'_{jk} & \cdots & 0 & a''_{jm+1} & \cdots & 1 & \cdots & a'_{jn}/a'_{jk} \\ & & & \vdots & & & & & \vdots & & \\ 0 & 0 & \cdots & -a'_{mk}/a'_{jk} & \cdots & 1 & a''_{mm+1} & \cdots & 0 & \cdots & a''_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b''_1 \\ b''_2 \\ \vdots \\ b'_j/a'_{jk} \\ \vdots \\ b''_m \end{pmatrix}$$

columna i col k (1.14)

Intercambiando la columa i con la columna k se tiene nuevamente un sistema del tipo (1.8). Todavía más, para actualizar el valor de f en (1.6.1), se tiene que actualizar B^{-1} pero esto se puede lograr multiplicando B^{-1} por la matriz:

$$\begin{pmatrix} 1 & 0 & \cdots & -a'_{1k}/a'_{jk} & \cdots & 0 \\ 0 & 1 & \cdots & -a'_{2k}/a'_{jk} & \cdots & 0 \\ & & & \vdots & & \\ 0 & 0 & \cdots & 1/a'_{jk} & \cdots & 0 \\ & & & \vdots & & \\ 0 & 0 & \cdots & -a'_{mk}/a'_{jk} & \cdots & 1 \end{pmatrix} \qquad (1.15)$$

Obteniéndose un problema del tipo (1.7).

Esto nos permite volver a repetir el proceso si no hemos encontrado la x óptima.

Supóngase ahora, que todos los elementos del vector r en (1.7), son positivos o cero. Si queremos aumentar desde cero el valor de alguna x_k no básica, se tendrá que f(x), en lugar de disminuir su valor, lo aumenta o queda igual, esto es; ya no hay forma de que f(x) decrezca y entonces hemos llegado al óptimo buscado, lo que nos permite decir que la señal de término de la solución, se da cuando los elementos del vector r son mayores o iguales a cero.

De todo lo anterior y suponiendo que ya tenemos una solución básica, el algoritmo simplex queda como sigue

A L G O R I T M O

1.- Si $r_k \geq 0$ para todo k>m: el algoritmo se termina. La x actual es la solución, en caso contrario

2.- Una vez que se encuentre la mas negativa de las r_k, encontrar el índice j, de las variables básicas para el cual:

$$b_j/a_{jk} = \min_i \left\{ b_i/a_{ik} \; ; \; \forall \; a_{ik} > 0 \right\} .$$

Si $a_{ik} \leq 0$ para toda i, entonces no existe solución finita: parar. En caso contrario.

3.- Actualizar la matriz B usando como pivote $a_{j,k}$ y hacer el intercambio entre las variables de bloqueo x_j y de conducción x_k, regresar al paso 1.

D U A L I D A D

Dado cualquier problema de programación lineal, existe un problema asociado a él de forma tal que si el problema es de minimización el otro será de maximización y, los valores óptimos de las funciones objetivo correspondientes, si son finitos, son iguales. Al problema de programación lineal original le llamaremos primal, mientras que a su asociado le llamaremos dual. De esta manera, si el problema primal es de la forma:

$$\begin{aligned} &\text{minimizar } f(x)=c^t x \\ &\text{sujeto a } Ax \geq b \\ &\qquad x \geq 0 \end{aligned} \tag{1.16}$$

El problema dual se define como:

$$\begin{aligned} &\text{maximizar } g(y)=y^t b \\ &\text{sujeto a } A^t y \leq c \\ &\qquad y \geq 0 \end{aligned} \tag{1.17}$$

donde como ya vimos $A \in \mathbb{R}^{mxn}$, b, $y \in \mathbb{R}^m$ & x, $c \in \mathbb{R}^n$.

El vector x contiene las n variables primales, mientras que y es el vector de las m variables duales. De manera general vamos a dar algunos resultados que nos dan la oportunidad de observar la relación entre ambos problemas, primal y dual.

Lemal .Si x & y son factibles para los problemas (1.16) y (1.17), respectivamente, entonces $c^t x \geq y^t b$ (Luenberger p.72).

El lema anterior nos muestra que un vector factible para uno u otro problema, produce una cota sobre el valor del otro problema. De esto, es claro que si x^* & y^* son factibles para (1.16) y (1.17), respectivamente, y si $c^t x^* = b^t y^*$, entonces x^* & y^* son óptimas para sus respectivos problemas y por tanto del lema

anterior se deduce el corolario siguiente:

Corolario.- Si x^* & y^* son factibles para los problemas (1.16) & (1.17) respectivamente,, y si $c^t x^* = b^t y^*$, entonces, x^* & y^* óptimizan a los problemas (1.16) & (1.17) respectivamente.

El teorema siguiente nos asegura que el inverso del corolario anterior, es tambien verdadero.

Teorema de la dualidad.- Si uno de los problemas(1.16) o (1.17) tiene solución óptima finita, el otro tambien la tiene y los valores de las funciones objetivo correspondientes son iguales. Si el valor de la función objetivo de uno de los problemas es no acotado, entonces el otro no tiene solución factible.(Luenberger pags. 72-73)

H O L G U R A C O M P L E M E N T A R I A

Supóngase que se tienen los problemas primal y dual (1.16) y (1.17). Restando el vector v de las variables de holgura a las restricciones del problema primal y agregando el vector u de las variables de holgura a las restricciones del problema dual, (para ponerlos en la forma estandar) se tiene que las restricciones de ambos nos quedan como sigue:

$$Ax-v = b \qquad (1.22.1)$$
$$A^t y+u = c \qquad (1.22.2)$$
$$\text{con } u,v,x,y \geq 0. \qquad (1.22.3)$$

Si x & y son los vectores óptimos correspondientes a (1.16) y (1.17) respectivamente, por el teorema de la dualidad debe suceder que

$$c^t x = y^t b. \qquad (1.23)$$

Premultiplicando (1.22.1) y (1.22.2) por $-y^t$ & por x^t respectivamente se obtiene:

$$-y^t Ax+y^t v = -y^t b \qquad (1.24.1)$$
$$x^t A^t y+x^t u = x^t c. \qquad (1.24.2)$$

Sumando (1.24.1) y (1.24.2) se tiene que

$$-y^t Ax+x^t A^t y+y^t v+x^t u = -y^t b+x^t c. \qquad (1.25)$$

Es claro que el primer y segundo terminos de (1.25) se anula y por (1.23) tenemos que: $-y^t b+x^t c = 0$,

$$\Rightarrow \quad x^t u+y^t v = 0$$
$$\text{para} \quad x \text{ \& } y \text{ óptimas.}$$

Puesto que se requiere que $u,v,x,y \geq 0$ en el óptimo, debe

suceder que

si $x_i > 0$ entonces $u_i = 0$

ó bién si $u_i > 0$ " $x_i = 0$

ó bién si $y_j > 0$ " $v_j = 0$

ó bién si $v_j > 0$ " $y_j = 0$

esto es; si una variable primal es positiva, entonces la correspondiente variable de holgura en el problema dual es cero y viceversa. Así mismo, si una variable dual es positiva, su correspondiente variable de holgura en problema primal es cero y viceversa. A esto es lo que se le llama holgura complementaria.

P R O B L E M A F U N D A M E N T A L

Una vez que se ha visto el concepto de holgura complementaria, se tiene que el problema de programación lineal tiene como condiciones de optimalidad, el encontrar vectores $u,v,x,y \geq 0$ tales que:

$$Ax-v = b \qquad (1.26.1)$$
$$A^t y+u = c \qquad (1.26.2)$$
$$\text{con } u,v,x,y \geq 0.$$

Otra manera de plantear el problema de programación lineal es la siguiente

De (1.26) tenemos que:

$$u = c-A^t y \qquad (1.27.1)$$
$$v = -b+Ax. \qquad (1.27.2)$$

Trabajando con matrices, (1.27) queda como:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} c \\ -b \end{pmatrix} + \begin{pmatrix} 0 & -A^t \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Si se definen

$$w = \begin{pmatrix} u \\ v \end{pmatrix}, \quad q = \begin{pmatrix} c \\ -b \end{pmatrix}, \quad M = \begin{pmatrix} 0 & -A^t \\ A & 0 \end{pmatrix}, \quad z = \begin{pmatrix} x \\ y \end{pmatrix}$$

el problema de programación lineal se traduce en encontrar vectores w & $z \geq 0$ tales que:

$$w = q + Mz$$
$$\text{con} \quad z^t w = 0,$$

el cual es llamado el problema fundamental.

P R O G R A M A C I O N C U A D R A T I C A

Una vez que el problema de programación lineal ha sido descrito, consideremos ahora el problema de programación cuadrática convexa, el cual se formula como sigue:

$$\text{minimizar } f(x)=c^t x+(1/2)x^t Dx \qquad (1.32.1)$$
$$\text{sujeto a } Ax\geq b \qquad (1.32.2)$$
$$x\geq 0 \qquad (1.32.3)$$

donde $D \in \mathbb{R}^{nxn}$ es simétrica y positiva definida, $A \in \mathbb{R}^{mxn}$ x, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$. (1.32.2) son las restricciones y (1.32.3) son la cotas del problema.

Como es sabido, si x es el óptimo de f sujeto a las restricciones, (1.32.2) y (1.32.3), deben existir vectores $u \in \mathbb{R}^n$, & y, $v \in \mathbb{R}^m$, tales que:

$$(a)\nabla_x f(x)-y^tA-u^t=c+Dx-y^tA-u^t=0$$
$$(b)\qquad Ax-v-b=0$$
$$(c)\quad y^t(Ax-b)=y^tv=0$$
$$(d)\qquad u^tx=0$$

las cuales son las condiciones de Kuhn-Tucker para el problema (1.32), Avriel Mordecai, p.152 [1].

Veamos como se pueden obtener estas condiciones, haciendo uso del problema (1.32), de su dual y de las variables de holgura de ambos problemas.

El dual del problema (1.32) es:

$$\text{maximizar } g(y)=b^t y-(1/2)x^t Dx \qquad (1.33.1)$$
$$\text{sujeto a } A^t y \le c+Dx \qquad (1.33.2)$$
$$\text{con } y \ge 0 \qquad (1.33.3)$$

donde $y \in R^m$, (1.33.2) son las restricciones y (1.33.3) son las cotas del problema dual. Van de Pane cap.9 [19]

.

Introduciendo los vectores v & u de las variables de holgura en las restricciones (1.32.2) y (1.33.2) respectivamente, para hacerlas igualdades, se obtiene:

$$Ax-v=b \qquad (1.34.1)$$
$$A^t y+u=c+Dx \qquad (1.34.2)$$

donde las variables primales son x & v, así como las duales son y & u. Se entiende que el vector de variables duales correspondiente al vector x es el vector u y el correspondiente al vector v es el vector y. Rearreglando (1.34) tenemos

$$\begin{aligned} Ax-v&=b \\ A^t y-Dx+u&=c \end{aligned} \qquad (1.35)$$

donde, las variables básicas son v y u, y las no básicas son x & y. Haciendo uso del concepto de holgura complementaria, se tiene entonces que

$$x^t u+y^t v=0 \qquad (1.36)$$

así que, para cualquier solución de (1.35) entonces debe cumplirse (1.36).

La siguiente expresión lagrangiana puede ser formulada para el problema de programación cuadrática,

$$L=c^tx+(1/2)x^tDx-y^t(Ax-v-b) \qquad (1.37)$$

donde y es el vector de los multiplicadores de Lagrange.

La derivada de L con respecto a x es

$$\nabla L=c + Dx - A^ty \qquad (1.38)$$

pero de (1.34.2) es claro que u=∇L, así que (1.38) queda como:

$$u = c + Dx - A^ty. \qquad (1.39)$$

Esta expresión, aparte del del término en Dx es casi la misma que la expresada por las restricciones del problema lineal dual, con las variables de holgura duales ya aumentadas (1.22.2). Esto lo usaremos para combinar (1.34.1) con (1.39) y así poder expresar f en términos de las variables primales x & v, así como de las variables duales u & y, de la siguiente forma, sustituyendo Dx de (1.39) en (1.32.1), f quedará como:

$$\begin{aligned} f(x) &= c^tx+(1/2)x^t(u-c+A^ty) \\ &= c^tx+(1/2)x^tu-(1/2)c^tx+(1/2)x^tA^ty \\ &= (1/2)c^tx+(1/2)x^tu+(1/2)y^tAx \\ &= (1/2)c^tx+(1/2)x^tu+(1/2)y^tv+(1/2)y^tb \quad (Ax=v+b). \end{aligned}$$

Si F=2f, entonces,

$$c^tx+y^tb+x^tu+y^tv. \qquad (1.40)$$

es una expresión bilineal (con términos bilineales x^tu & y^tv).

Haciendo uso de (1.36) tenemos que para cualquier solución de (1.35) los términos bilineales se hacen cero , lo que nos permite poner a F como:

$$F = c^t x + y^t b$$

De esto se tiene que, para encontrar la solución del problema de programación cuadrática hay que resolver el siguiente sistema:

$$\begin{aligned} Ax - v &= b \\ -A^t y + Dx + c &= u \\ c^t x + y^t b &= F, \end{aligned}$$

que rearreglado queda como:

$$\begin{aligned} F \quad - c^t x - b^t y &= 0 \\ -u \quad + Dx - A^t y &= -c \\ v - Ax \qquad &= -b, \end{aligned} \tag{1.41}$$

en donde las variables básicas del sistema son: F, u, v & las no básicas son x & y.

Nuevamente, haciendo uso del concepto de holgura complementaria se observa, de (1.41) que si una variable dual es básica, su asociada primal es no básica y viceversa. Un sistema que cumple con estas condiciones es llamado un sistema estandar y su correspondiente solución es llamada una solución estandar. Esto implica que, para cualquier solución estandar los t érminos bilineales se desvanecen.

De aquí que las condiciones necesarias y suficientes para que una solución del problema cuadrático (1.32) sea la óptima, son las de encontrar vectores u, v, x, y≥0 tales que:

$$Ax - v = b \qquad (1.42.1)$$

$$Dx - A^t y + c = u \qquad (1.42.2)$$

$$x^t u + y^t v = 0 \qquad (1.42.3)$$

que son las condiciones de Kuhn-Tucker ya vistas en (a), (b), (c) y (d) (pag. 17).

Rearreglando (1.42.1) y (1.42.2) tenemos

$$u = c + Dx - A^t y$$
$$v = -b + Ax.$$

Para obtener el problema fundamental, se definen:

$$w = \begin{pmatrix} u \\ v \end{pmatrix}, \quad q = \begin{pmatrix} c \\ -b \end{pmatrix}, \quad z = \begin{pmatrix} x \\ y \end{pmatrix} \quad \& \quad M = \begin{pmatrix} D & -A^t \\ A & 0 \end{pmatrix}.$$

El problema de programación cuadrática se reduce entonces a encontrar vectores w & z≥0 tales que:

$$w = q + Mz$$

$$\text{con} \quad z^t w = 0$$

Hay que notar que si en la matríz M, la submatríz D=0, entonces el problema lineal es un caso especial del problema cuadrático.

CAPITULO
II

ALGORITMO DE LEMKE

Se ha concluído, que la solución óptima para el problema de programación cuadrática convexa se obtiene, cuando sea posible encontrar vectores u, v, x, y ≥0 tales que las condiciones de Kuhn-Tucker se satisfagan; esto es,

$$\begin{aligned} Dx - A^t y + c &= u \\ Ax - b &= v \\ x^t u + y^t v &= 0 \end{aligned} \qquad (2.1)$$

Definiendo nuevamente

$$w=\begin{pmatrix} u \\ v \end{pmatrix}, \quad q=\begin{pmatrix} c \\ -b \end{pmatrix}, \quad z=\begin{pmatrix} x \\ y \end{pmatrix} \quad \& \quad M=\begin{pmatrix} D & -A^t \\ A & 0 \end{pmatrix}, \qquad (2.2)$$

se vió que (2.1) se transforma de forma tal que el problema se traduce en encontrar vectores w & z ≥ 0 tales que:

$$w = q + Mz \qquad (2.3.1)$$

$$\text{con} \quad z^t w = 0 \qquad (2.3.2)$$

El problema (2.3) es llamado el problema fundamental y

si D=0, se tiene el caso lineal. En este sentido el problema de programación lineal es un caso específico del problema fundamental. Es más, al empezar el vector w es el de las variables básicas y el vector z es el de las no básicas.

El primer algoritmo propuesto para la solución de (2.3) fué el método del pivote principal, de Cottle y Dantzig[11], seguido después por el algoritmo dado por Lemke[11] (que es más simple y más rápido). La descripción del algoritmo de Lemke, es como sigue:

Puesto que la no negatividad de las variables es necesaria, ya que para toda i=1,...,n , $w_i z_i=0$, entonces, o bién $w_i=0$ ó bién $z_i=0$. Los pares correspondientes (w_i,z_i) se conocen como pares complementarios, entendiéndose que si la variable w_i es básica, la variable z_i es necesariamente no básica o viceversa.

Es claro que si el vector $q\geq 0$, una solución del problema es:

$$w = q, \quad z = 0. \tag{2.4}$$

Todavía más, si una solución existe, entonces debe ser posible partir los vectores w & z en dos conjuntos de pares complementarios, a saber: $\{w_a,z_a\}$ y $\{w_b,z_b\}$, donde $w_a=0$ y $z_b=0$, de forma tal que (2.3) se satisfaga. La interrogante es ahora como encontrar la partición apropiada ?, lo que Lemke resuelve de manera muy sencilla mediante la adición de una variable z_0 extra cuyos coeficientes agregan una nueva columna M_0 al sistema (2.3), de manera tal que los elementos de esta nueva columna se elaboran como sigue:

$$\begin{aligned} m_{i0} &= q_i \quad si \quad q_i \geq 0 \\ m_{i0} &= -q_i \quad si \quad q_i < 0, \end{aligned}$$

ó bién

$$m_{i0} = 1 \quad \forall\ i=1,\ldots,n.$$

Al hacer esto se genera lo que llama el problema extendido, este se puede enunciar como sigue: encontrar vectores w & z tales que

$$w = M_0 z_0 + Mz + q \qquad (2.5.1)$$

$$z^t w = 0 \qquad (2.5.2)$$

$$\text{con } z_0,\ z \ \&\ w \geq 0, \qquad (2.5.3)$$

Es claro que cualquier solución de (2.5), con $z_0=0$, es una solución del problema original (2.3).

Del problema (2.5) se nota que si $z=0$, w será mayor que cero siempre que z_0 tome valores suficientemente grandes. Esto significa, el algoritmo de Lemke empieza siempre encontrando el mínimo valor de z_0 , digamos z_0^0 , tal que $w = q + M_0 z_0^0 \geq 0$ y para el cual alguna variable del vector w, digamos w_r es igual a cero. Resolviendo entonces la r-ésima ecuación para z_0 y sustituyendo su valor en las demás ecuaciones, resulta un nuevo sistema, de la siguiente forma

$$w' = M_0' w_r + M' z' + q', \qquad (2.7)$$

el cual se caracteriza porque $w' \geq 0$ y porque z_0 es un elemento de w'.

La operación realizada para transformar (2.3) en (2.7) es una de pivoteo y es la misma que se usa en programación lineal. De aquí que, a las variables de la izquierda las llamaremos básicas y a las de la derecha no básicas. De esta manera, en la iteración cero se considera a z_0 como no básica y mediante la operación de pivoteo se ha hecho básica, tomando como pivote al elemento $m_{r,0}$, de z_0 , donde el índice r corresponde al número del renglón para el cual se ha encontrado el

$$\max_{1 \leq i \leq n} \left\{ |q_i| \;\middle|\; q_i < 0 \right\} = r \qquad (2.8)$$

Si en la iteración cero $q_i \geq 0$, $\forall$ i=1,...,n , no hay necesidad de tal pivoteo ya que estamos en el caso (2.4) y la solución es w=q; si $q_i < 0$, para alguna i, llevamos a cabo el pivoteo, lo que ocasionará que w_r deje la base entrando a ella z_0. Esto dará nacimiento al primer par no básico (w_r, z_r), el cual, es un par complementario y es por esto que necesariamente uno de los dos elementos del par, en este caso z_r, debe convertirse en básico; pero hay más, al haber tomado al elemento $m_{r,0}$ como pivote de acuerdo a (2.8), se tiene la garantía de que todas las variables básicas (incluyendo a z_0) son mayores o iguales a cero. La idea es ahora la siguiente: si logramos que z_0 deje la base y se pueda conservar la no negatividad de las variables básicas, se tendrá la solución requerida. Para llevar a cabo esto, Lemke usa lo que él llama principio de complementariedad, del cual ya se hizo uso, que consiste en lo siguiente: para cualquier par complementario (w_i, z_i) no debe suceder que ambos sean no básicos a la vez, esto es; uno debe ser básico y el otro no.

Al haber dejado la base w_r, se ha formado el par (w_r, z_r) cuyos elementos son ambos no básicos, así que, z_r debe volverse básico para no violar el principio de complementariedad. La forma en que se efectúa esto es la siguiente: tómese la columna de la matríz M correspondiente a la variable z_r cuyos elementos son de la forma m_{ir}. El pivote a usar debe ser tal que permita conservar la no negatividad de las variables básicas, lo que se consigue mediante el siguiente criterio: los índices del pivote m_{ir} son aquellos para los cuales se encuentra el

$$\min \left\{ q_j / m_{jr} \;\middle|\; m_{jr} > 0,\ \forall\ j=1,\ldots,n \right\}, \qquad (2.9)$$

exáctamente como en el simplex, lo que permite que z_r crezca unicamente lo suficiente, bloqueando tal crecimiento la variable básica w_i, donde el índice i se determina usando el criterio (2.9). Es por esto que a esta variable se le llama variable de bloqueo, mientras que a la variable z_r que entra a la base se le llama variable de conducción. Si la variable de bloqueo es z_0, se realiza el pivoteo y ya se terminó el proceso (ya que deja la base

y se hace cero). En caso contrario, la variable de bloqueo w_i formará con la z_l correspondiente un nuevo par no básico y el método se vuelve a aplicar usando el criterio (2.9). Si ocurre que la variable de conducción es tal que $m_{ir}\leq 0$, $\forall$ i=1,...,n, entonces por mas que aumente el valor de z_r no habrá variable de bloqueo y el proceso termina en un rayo; esto es, no existe solución. La forma de terminación de acuerdo al algoritmo es un rayo o bien la reducción de z_0 al valor cero, proporcionando el algoritmo una solución finita.

El algoritmo de Lemke puede escribirse como sigue:

Suponiendo que q es tal que no todos sus componentes son mayores o iguales a cero.

1) Encontrar el indice j tal que

$$\max_{1\le i\le m} \left\{ |q_i| \;\middle|\; q_i < 0 \right\} = q_j$$

pivotear w_j con z_0, y hacer c=j

lo que nos transforma M y q en M' y q'

2) Encontrar el

$$\min \left\{ q'_i/m'_{i,c} \;\middle|\; m'_{i,c} > 0,\ \forall\ i=1,\ldots,n \right\} = q'_r/m'_{r,c}$$

si $m'_{i,c} \le 0\ \forall\ i=1,\ldots,n$, no hay solucion.
En caso contrario,

3) Pivotear w'_r con z'_c. Si $w'_r = z_0$, se tiene la solución.

En caso contrario

4) Hacer c=r, y regresar al paso 2.

IMPLEMENTACION DE SARGENT DEL ALGORITMO LEMKE

De acuerdo al algoritmo dado por Lemke hay que transformar todo el sistema para pasar de una iteración a otra. Sin embargo, para efectuar la prueba del paso 2 del algoritmo de Lemke, solo dos vectores son los que realmente se necesitan. Estos son: q' y M'_r , donde M'_r es la columna correspondiente a la variable de conducción. R. W. H. Sargent [16] propone que en lugar de transformar todo el sistema en cada iteración, unicamente se actualicen estos dos vectores generandolos de los datos originales.

Cada vez que se efectúa una iteración los miembros de cada par complementario (w_i, z_i) están en lados opuestos del sistema, excepto el par no básico (w_r, z_r), el cual contiene a la nueva variable de conducción y a la anterior variable de bloqueo que acaba de ser pivoteada. Esto nos produce una partición del sistema original, que puede expresarse de la siguiente manera,

$$\begin{pmatrix} w_1 \\ w_r \\ \hline w_2 \end{pmatrix} = \left(\begin{array}{cc|cc} M_{10} & M_{11} & M_{1r} & M_{12} \\ M_{r0} & M_{r1} & M_{rr} & M_{r2} \\ \hline M_{20} & M_{21} & M_{2r} & M_{22} \end{array}\right) \begin{pmatrix} z_0 \\ z_1 \\ \hline z_r \\ z_2 \end{pmatrix} + \begin{pmatrix} q_1 \\ q_r \\ \hline q_2 \end{pmatrix} \qquad (2.10)$$

donde la variable de bloqueo está en el bloque $\begin{pmatrix} w_1 \\ w_r \end{pmatrix}$ y la variable de conducción está en el bloque $\begin{pmatrix} z_0 \\ z_1 \end{pmatrix}$

La partición del sistema (2.10) se puede resumir como

$$\begin{pmatrix} \bar{w}_1 \\ \bar{w}_2 \end{pmatrix} = \begin{pmatrix} \bar{M}_{11} & \bar{M}_{12} \\ \bar{M}_{21} & \bar{M}_{22} \end{pmatrix} \begin{pmatrix} \bar{z}_1 \\ \bar{z}_2 \end{pmatrix} + \begin{pmatrix} \bar{q}_1 \\ \bar{q}_2 \end{pmatrix} \qquad (2.11)$$

el cual una vez transformado queda como

$$\begin{pmatrix} \bar{z}_1 \\ \bar{w}_2 \end{pmatrix} = \begin{pmatrix} \bar{M}'_{11} & \bar{M}'_{12} \\ \bar{M}'_{21} & \bar{M}_{22} \end{pmatrix} \begin{pmatrix} \bar{w}_1 \\ \bar{z}_2 \end{pmatrix} + \begin{pmatrix} \bar{q}'_1 \\ \bar{q}'_2 \end{pmatrix} \qquad (2.12)$$

donde (2.11) y (2.12) estan relacionados mediante las siguientes formulas.

$$\bar{M}_{11}\,\bar{M}'_{11} = I\,, \quad \bar{M}_{11}\,\bar{M}'_{12} = -\bar{M}_{12}\,,$$

$$\bar{M}_{11}\,\bar{q}'_1 = -\bar{q} \qquad \bar{M}'_{21} = \bar{M}_{21}\,\bar{M}'_{11}\,, \qquad (2.13)$$

$$\bar{M}'_{22} = \bar{M}_{22} + \bar{M}_{21}\,\bar{M}'_{12}\,, \quad \bar{q}'_2 = \bar{q}_2 + \bar{M}_{21}\,\bar{q}'_1$$

Ahora bien, puesto que la matríz M_{11} no es singular, se puede factorizar como

$$\bar{M}_{11} = V^t D R_{11}$$

donde, R_{11} es una matríz triángular superior con diágonal principal llena de unos, D es una matríz diágonal, y V es una

matríz cuadrada tal que

$$\bar{M}_{11} = (V^t D^{1/2})(D^{1/2} R_{11})$$

es la factorización ortogonal QR de $\bar{M}_{11}$, con $V^t D^{1/2}=Q$ y $D^{1/2}R_{11}=R$.

Ahora, como

$$V^t D^{1/2} = Q$$

$$\Rightarrow \qquad V^t = QD^{-1/2}$$

por lo tanto

$$VV^t = D^{-1/2}Q^t QD^{-1/2} = D^{-1}$$

Además,

$$V^t DV = QD^{-1/2}DD^{-1/2}Q^t = QIQ^t = I$$

de esto, se tiene que

$$\bar{M}_{11} = V^t D R_{11}$$

se puede poner como:

$$V\bar{M}_{11} = R_{11} \qquad (2.14)$$

Como ya se dijo, de lo que trata la implementación de Sargent, es de no transformar toda la matríz M en cada iteración, sino unicamente actualizar la columna $\bar{M}_r$, correspondiente a la variable de conducción, y el vector q, para lo que usaremos las formulas dadas en (2.13), lo que nos permitirá obtener la columna $\bar{M}_r'$ y el vector $\bar{q}'$ que puedan ser usados para efectuar la prueba del paso 2 del algoritmo de Lemke sin alterar los elementos de la matríz M. Para lograr la actualización de la columna $\bar{M}_r$ y del vector q mediante (2.13), Sargent hace uso de la factorización ortogonal QR de la matriz $\bar{M}_{11}$ en (2.11) actualizando tal ortogonalización mediante el método dado por Gill, Murray & Saunders [5], lo que permite que el trabajo de cálculo se vea disminuido.

Un ejemplo, nos permitirá entender mejor el método

desarrollado por Sargent. Supóngase que el problema a resolver es el siguiente:

	z_0	z_1	z_2	z_3	z_4	q
w_1	-1	-2	-1	1	1	1
w_2	-6	-1	-2	-1	1	-6
w_3	-2	-1	1	0	0	-2
w_4	-4	-1	-1	0	0	-4

en este momento, no existe una submatriz $\bar{M}_{11}$ de M factorizada en factores QR=VDR así que si llevamos un contador m, del número de renglones de $\bar{M}_{11}$, entonces m=0. Se observa que el elemento del vector q con valor mas negativo es -6 el cual se encuentra en el renglón 2 lo que implica que el renglón de bloqueo es el 2, este contiene desde luego a la variable w_2, lo que nos dice que z_0 se transforma en básica, w_2 en no básica y la próxima variable de conducción es z_2. Ponemos entonces, el renglón 2 de la matriz M en el lugar m+1=1, y la columna correspondiente a la variable z_2, en el lugar m+2=2 con lo que la matriz M rearreglada nos queda como sigue:

Sargent

	z_0	z_2	z_1	z_3	z_4	q
w_2	-6	-2	-1	-1	1	-6
w_1	-1	-1	-2	1	1	1
w_3	-2	1	-1	0	0	-2
w_4	-4	-1	-1	0	0	-4

Lemke

	w_2	z_1	z_2	z_3	z_4	q
z_0						
w_1						
w_3						
w_4						

La primera submatríz $\bar{M}_{11}$, de M a factorizar tiene solo un elemento, a saber, -6 que factorizada en VDR,queda como:

$$\bar{M}_{11}=[-6]=[-1/6][36][1]$$

ahora bién, se tiene que actualizar la columna $\bar{M}_r$ correspondiente a la variable z_2, para efectuar el paso 2 del algoritmo de Lemke. Esto se hace como sigue:

Al empezar el proceso se tiene que todas las variables w_i son básicas y todas las z_i son no básicas incluyendo desde luego a z_0. Si llevamos un contador m, del número de renglones de la matriz $\bar{M}_{1,1}$ factorizada, entonces m=0 pues como se dijo, no hay nada factorizado. Ahora bién, como ya se sabe, la primera variable de conducción es z_0, supóngase entonces, que w_r es la primera variable de bloqueo. Rearreglando la matriz M de forma tal que el renglón correspondiente a la variable w_r pase al primer renglón, esto es; al renglón m+1=0+1=1, entonces la primera matriz $\bar{M}_{1,1}$ a factorizar tiene un solo elemento, a saber $m_{r,0}$. Pongamoslo como m_{11}, ya que está ocupando el lugar 11 de la matríz M rearreglada. En este momento z_0 entra a la base y w_r sale de ella.

Haciendo la factorización ortogonal de la matríz $\bar{M}_{11}$ (que consta de un solo elemento) tenemos de (2.14) que :

$$V\bar{M}_{11} = R_{11}$$

(mas adelante se verá como realizar tal factorización) y como w_r fué la variable de bloqueo (y salió de la base), entonces z_r será la próxima variable de conducción. Nuevamente se rearregla la matriz M trasladando la columna M_r correspondiente a la variable z_r, al lugar m+2=0+2=2 (ya que el primer lugar está ocupado por la columna correspondiente a z_0), de esta forma se nos ha generado la submatriz de M, $\left(\bar{M}_{11} \middle| \bar{M}_{1r}\right)$ que en este caso consta solo de 2 elementos m_{11} y m_{12} donde m_{12} es el primer elemento correspondiente a la columna $\bar{M}_r$ puesta en el 2º lugar. La razón por la cual se puso la columna $\bar{M}_r$ en el lugar 2 de la matríz M es la siguiente: puesto que z_r es la próxima variable de conducción, es deseable actualizar la columna correspondiente a tal variable, en este caso $\bar{M}_r$, para efectuar la prueba del paso 2 del algoritmo de Lemke, la actualización de tal columna se puede realizar como

sigue: de (2.14) se tiene que $V\bar{M}_{11} = R_{11}$.

$$\Rightarrow \quad V\left(\bar{M}_{11} \;\vdots\; \bar{M}_{1r}\right) = \left(R_{11} \;\; R_{1r}\right) \qquad (2.15)$$

(para alguna R_{1r}). De (2.13), tenemos que $M_{11} \;\; M'_{1r} = M_{1r} \quad \Rightarrow$

$$V\bar{M}_{11} \;\bar{M}'_{1r} = V\bar{M}_{1r} \qquad (2.16)$$

pero de (2.15),

$$V\bar{M}_{11} = R_{11} \qquad y \qquad V\bar{M}_{1r} = R_{1r} \qquad (2.17)$$

combinando (2.16) y (2.17) nos queda:

$$R_{11} \;\bar{M}'_{1r} = R_{1r} \qquad (2.18)$$

y ya se tiene actualizada la primera parte ($\bar{M}'_{1r}$), de la columna $\bar{M}_r$.

Para actualizar la 2ª parte de la columna $\bar{M}_r$, se vuelve a hacer uso de las fórmulas (2.13) obteniendose que

$$\bar{M}'_{2r} = \bar{M}_{2r} - \bar{M}_{21} \;\bar{M}'_{1r} \qquad (2.19)$$

y ya se tiene la columna $\bar{M}'_r$ completa ($\bar{M}_r$ actualizada).

De la forma en que se obtienen (2.18) y (2.19) es claro que no se hace uso de la matriz V para la actualización de $\bar{M}_r$, lo que nos dice que unicamente hay que trabajar con la matriz R_{11} .

Si el siguiente renglón de bloqueo es tal que su índice r> m+1=1, entonces la variable contenida en tal renglón pertenece a las variables w_i que no han dejado la base. Vamos a suponer que w_c es la variable contenida en tal renglón ; la próxima variable de conducción es z_c y es hasta este momento que se incrementa el valor de m ; así que $m = m + 1 = 1$.

Nuevamente, la matríz M se rearregla de forma tal que el renglón r conteniendo a la variable w_c pase al lugar m+1 = 2 y la nueva submatríz $\bar{M}_{11}$ de la matríz M a factorizar es de la forma

$$\bar{M}_{1,1} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$$

donde, m_{21} y m_{22} , son los primeros elementos correspondientes al renglón que contiene a la variable w_c. Tomando en cuenta que la variable contenida en el renglón r de bloqueo, fué w_c, entonces la próxima variable de conducción debe ser z_c . Lo que nos dice que debemos tomar la columna correspondiente a la variable z_c , y ubicarla en el lugar m+2=3, lo que da por resultado la submatríz $\left(\bar{M}_{11} \;\middle|\; \bar{M}_{1r}\right)$ de la matríz M, esto es;

$$\left(\bar{M}_{11} \;\middle|\; \bar{M}_{1r}\right) = \left(\begin{array}{cc|c} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{array}\right)$$

donde, $\begin{pmatrix} m_{13} \\ m_{23} \end{pmatrix}$, son los primeros elementos de la columna $\bar{M}_r$ de la variable z_r. Aplicando (2.18) y (2.19), se obtendrá la columna $\bar{M}'_r$. Y nuevamente se repite el proceso, siempre que el índice r del renglón de bloqueo sea tal que r>m+1.

Veamos que pasa si el renglón de bloqueo es tal que su índice r=m+1.

Supóngase que en un momento dado la submatríz $\left(\bar{M}_{11} \;\middle|\; \bar{M}_{1r}\right)$ tiene

la forma:

$$\left(\begin{array}{ccc|c} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} \end{array}\right)$$

se debe observar lo siguiente: la columna M_r a actualizar es la cuarta; además como apenas se va a a efectuar la prueba del paso 2 resulta que m=2. Todavía mas, supóngase que las variables correspondientes a esta submatriz estan arregladas de la siguiente manera:

$$\begin{array}{c|ccc} & z_0 & z_7 & z_4 & z_2 \\ w_7 & & & & \\ w_4 & & & & \\ w_2 & & & & \end{array} \qquad (2.20.a)$$

Este arreglo en el algoritmo de Lemke sería el siguiente:

$$\begin{array}{c|ccc|c} & w_7 & w_4 & w_2 & z_2 \\ z_0 & & & & \\ z_7 & & & & \\ z_4 & & & & \end{array} \qquad (2.20.b)$$

Aquí las variables que han dejado la base son w_7, w_4, y w_2 y las básicas son z_0, z_7 y z_4, y en este momento la variable de conducción es z_2. Es claro que en caso de que el renglón de bloqueo sea por ejemplo el 2, la variable que deja la base, no es w_4 como se observa en (2.20.a), sino z_7 como se observa en (2.20.b). Así mismo si el renglón de bloqueo es el 3, la variable de bloqueo no es w_2, sino z_4, y si el renglón de bloqueo es el 1, la variable de bloqueo es, desde luego, z_0. Sea $r = 2 \leq m+1 = 3$ el índice del renglón de bloqueo, la variable que se encuentra en tal renglón es w_4 la cual está asociada a z_7. Entonces, la variable de

bloqueo es z_7, lo que implica que la próxima variable de conducción es w_7 y desde luego z_7 dejará la base. Se tiene entonces que encontrar la factorización de la submatríz $\tilde{M}$ de M que ya no contenga la columna correspondiente a la variable z_7.

Ahora bien, puesto que la próxima variable de conducción es w_7 habrá que actualizar la columna correspondiente a tal variable la cual es el vector e_7 (ya que las básicas que pertenecen al vector w forman parte de la matríz I). Para eso pasamos z_7 al último lugar (m+2=4) y ya que z_7 está asociada a w_4 en el sentido que z_7 entró a la base por w_4, tambien llevamos a w_4 al último lugar (m+1=3), quedando las variables arregladas de la siguiente forma:

$$\begin{array}{c|c} & z_0 \quad z_4 \quad z_2 \\ w_7 & \\ w_2 & \\ w_4 & \end{array} \; z_7 \qquad z_7 \text{ queda fuera de } \tilde{M}\text{, no así } w_4 \qquad (2.21)$$

Lo que se ha hecho es trasladar tanto la columna como el renglón r=2 de la matríz (2.20) a los últimos lugares respectivamente (m+2 y m+1), de esta forma, la submatríz $\tilde{M}$ de M a factorizar es:

$$\tilde{M} = \begin{pmatrix} m_{11} & m_{13} & m_{14} \\ m_{31} & m_{33} & m_{34} \\ m_{21} & m_{23} & m_{24} \end{pmatrix} \qquad (2.22)$$

Supóngase que ya se factorizó y que $\tilde{V}\,\tilde{M} = \tilde{R}_{11}$, puesto que la columna correspondiente a la variable de conducción es $e_7=(1,0,\ldots,0)^t$, ya que w_7 está en el primer lugar, entonces en lugar de (2.16) se tendrá

$$\tilde{V}\ \tilde{M}\ M^{1}_{1r} = \tilde{V}e^{1}_{7} \qquad e_7 = \begin{pmatrix} e^{1}_{7} \\ e^{2}_{7} \end{pmatrix} \qquad (2.24)$$

y por tanto

$$\tilde{R}_{11}\ M^{1}_{1r} = \tilde{V}e^{1}_{7}$$

y como e^{2}_{7} es la parte de e_7 que no contiene a 1 $\Rightarrow$ $e^{2}_{7} = 0$ entonces

$$M^{1}_{2r} = - M_{21}\ M^{1}_{1r} \quad \text{(de acuerdo a las fórmulas 2.13)}$$

(entendiendose que M^{1}_{r} es e_7 actualizado).

Hecho esto se aplica la prueba del paso 2 del algoritmo de Lemke para determinar el nuevo renglón de bloqueo.

Hasta este punto, la matríz que se ha factorizado es (2.22) que es la que corresponde a las variables de (2.21). Sin embargo W_7 acaba de entrar a la base, entonces es necesario quitar de la matríz (2.22) el renglón correspondiente a w_7 , esto es; hay que quitar a w_7 de las variables no básicas y ponerla en las básicas y además poner m=m-1, de aquí que al quitar el primer renglón de (2.22) quede como:

$$\begin{pmatrix} m_{31} & m_{33} & m_{34} \\ m_{21} & m_{23} & m_{24} \end{pmatrix} \qquad (2.26)$$

que corresponde al arreglo de las variables:

$$\begin{matrix} & z_0 & z_4 & z_2 \\ w_2 & & & \\ w_4 & & & \end{matrix}$$

Al haber quitado un renglón, habrá que refactorizar la matríz (2.26) y seguir con el proceso de acuerdo a que la última

variable de bloqueo pertenezca al vector w ó al vector z. La forma en que nos damos cuenta de si la variable de bloqueo pertenece a w ó a z es observando si el índice r del renglón de bloqueo es mayor ó menor ó igual que m+1 respectivamente.

De aquí observamos que, si la variable de bloqueo pertenece a w (r>m+1), aumentamos un renglón y una columna a la matríz M_{11} y m=m+1; de manera contraria si la variable de bloqueo pertenece a z (r≤m+1), quitaremos un renglon y una columna a la matríz M_{11} y m=m-1.

Es por esto que el algoritmo dado por Sargent, para implementar el método de Lemke se va a dividir en 2 partes. Y ya que para la actualización de la columna M_r solo se hace uso de la matríz R, esas 2 partes se refieren a la actualización de tal matríz. En la primera parte se actualiza R en caso de que se aumenten un renglón y una columna de la matríz a factorizar y es por eso que se llama algoritmo agrega. En el 2^{o} caso, se hace la actualización de R cuando un renglón y una columna son quitados de la matríz a factorizar y se le llama algoritmo elimina.

Es claro que si la variable de bloqueo es z_0, es suficiente actualizar el vector q', obteniéndose la solución, esto es;

$$z_a = q'_1 \qquad \text{y} \qquad w_b = q'_2 .$$

Para obtener el vector q' actualizado, no es necesario actualizar el vector q original cada vez, lo que se hace para actualizar q es usar el q' anterior así que el nuevo q' se obtiene como

$$q'_r = q'_r \,/\, m'_{rc} \tag{2.28}$$

y $$q'_i = q'_i - m'_{ic}\, q'_r \qquad \text{para } i \neq r$$

Una vez que se ha visto como actualizar la columna M_r y el vector q' en cada iteración, procederemos a ver como es que se

actualiza la matríz R cada vez que se agrega o se quita un renglón y una columna de la matríz M_{11} .

Para eso, vamos a suponer que tenemos una matríz M de 2x3 ya factorizada, esto es; que M está puesta como VM = R o lo que es lo mismo,

$$V\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{pmatrix} = \begin{pmatrix} 1 & R_{12} & R_{13} \\ 0 & 1 & R_{23} \end{pmatrix}$$

Supóngase ahora, que la variable de bloqueo es w_r, agreguemos entonces a la matríz M el renglón correspondiente a la variable w_r , que como ocupará el renglón número 3 lo pondremos como, $\begin{pmatrix} m_{31} & m_{32} & m_{33} \end{pmatrix}$, formandose la matríz:

$$M' = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \qquad (2.30)$$

lo que queremos es encontrar una matríz R' tal que V'M' = R'. Para lograr esto, se hace uso del método desarrollado por Gill, Murray y Saunders [5], el cual puede describirse como sigue:

Primeramente, se tiene que

$$M = \left(\begin{array}{cc:c} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{array}\right) = V^t D R$$

$$= V^t \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} \begin{pmatrix} 1 & R_{12} & R_{13} \\ 0 & 1 & R_{23} \end{pmatrix} .$$

Al agregar a la matríz M el renglón correspondiente a la variable w_r tenemos que,

$$M' = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{22} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} = \begin{pmatrix} V^t & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & R_{12} & R_{13} \\ 0 & 1 & R_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

$$= \begin{pmatrix} V^t & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ p_1 & p_2 & p_3 \end{pmatrix} \begin{pmatrix} 1 & R_{12} & R_{13} \\ 0 & 1 & R_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

donde, p_1, p_2 y p_3 son tales que,

$$\begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix} \begin{pmatrix} 1 & R_{12} & R_{13} \\ 0 & 1 & R_{23} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} m_{31} & m_{32} & m_{33} \end{pmatrix}$$

y de acuerdo al algoritmo [5], para llevar a cabo la factorización de M', se factoriza la matriz elemental

$$\begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ p_1 & p_2 & p_3 \end{pmatrix} \qquad (2.32)$$

la cual una vez factorizada queda como:

$$\begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ p_1 & p_2 & p_3 \end{pmatrix} = \hat{V}^t \hat{D} \begin{pmatrix} U & p_3 \beta \\ 0 & 1 \end{pmatrix}$$

donde U es una matriz triángular superior con unos en la diágonal principal y cada $u_{i,j}$ es de la forma,

$$u_{i,j} = p_i\beta_j \qquad \text{para } i<j$$

y además

$$\bar{\bar{D}}=\hat{D}\begin{pmatrix} I & 0 \\ 0_3 & (p)^2 \end{pmatrix}.$$

Una vez que se ha hecho la factorización de (2.32), la factorización de (2.30), se obtiene como sigue:

$$M' = V^t\hat{V}^t\bar{\bar{D}}\begin{pmatrix} U & p_j\beta \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & R_{12} & R_{13} \\ 0 & 1 & R_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \bar{V}^t\ \bar{\bar{D}}\ \bar{R}.$$

Ya tenemos pues, como factorizar (2.30), sin embargo nos falta agregar a la matriz M', la columna correspondiente a la variable z_r, que es la próxima variable de conducción y hay que actualizar la factorización de tal columna para poder obtener M'_r mediante (2.18) y efectuar la prueba del paso 2 del algoritmo de Lemke, para ir a la próxima iteración.

Supóngase que agregamos a M' la columna correspondiente a la variable z_r que ocupará la cuarta columna, esta columna la ponemos como $M_4 = \begin{pmatrix} m_{14} \\ m_{24} \\ m_{34} \end{pmatrix}$, lo que nos produce la matriz:

$$\bar{M} = \left(\begin{array}{ccc|c} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{array}\right) \qquad (2.34)$$

Entonces, falta por factorizar la columna M_4, lo que se

hace como sigue: de (2.15) se tiene que $VM_{1r} = R_{1r}$ ⇒

$$\bar{R}_4 = \begin{pmatrix} R_{14} \\ R_{24} \\ R_{34} \end{pmatrix} = \bar{V}M_4 \qquad (2.36)$$

multiplicando (2.36) por $\bar{R}^t\ \bar{D}$

$$\bar{R}^t\ \bar{D}\ \bar{R}_4 = \bar{R}^t\ \bar{D}\ \bar{V}\ M_4$$

pero $M' = \bar{V}^t\ \bar{D}\ \bar{R}$, por tanto, $\bar{R}^t\ \bar{D}\ \bar{R}_4 = M'^t\ M_4$ y entonces,

$$\bar{R}_4 = \bar{D}^{-1}\ \bar{R}^{-t}\ M'^t\ M_4$$

con lo que se concluye la factorización de (2.34).

De todo lo anterior, se observa que es necesario calcular los valores de p_i, de β_i, de cada elemento d_i en D y de los elementos de la matriz $\bar{R}^{-1}$para obtener la factorización de (2.34). El siguiente algoritmo, Sargent[16], que es el primero de las 2 partes de las que ya se ha hablado, es una generalización del método de Gill, Murray y Saunders [5] y nos produce de manera iterativa, cada uno de los valores requeridos para lograr tal factorización.

Se supone que el orden de la matriz M antes de agregarle un renglón y una columna es mx(m+1).

A L G O R I T M O A G R E G A

1.- poner $t_0=1$

$$u_{0j} = \sum_{i=1}^{m+1} M_{ij} \quad M_{im+2} \quad , \text{ para } j=1,\ldots,m+1$$

$$v_{0j} = M_{m+1j} \quad , \text{ para } j=1,\ldots,m+1$$

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

$$p_i=v_{i-1i} \quad , \quad t_i=t_{i-1}p_i/D_i$$
$$\bar{D}_i=D_i t/t_{i-1} \quad , \quad \beta_i=p_i/D_i t_i$$
$$v_{ij} = v_{i-1j} - v_{i-1i} R_{ij} \quad , \quad j=i+1,\ldots,m+1$$
$$\bar{R}_{ij} = R_{ij} + \beta_i v_{ij} \quad , \quad j=i+1;\ldots,m+1$$
$$u_{ij} = u_{i-1j} - u_{i-1i} \bar{R}_{ij} \quad , \quad j=i+1,\ldots,m+1$$
$$\bar{R}_{im+2} = u_{i-1i} / \bar{D}_i$$

3.- poner

$$\bar{D}_{m+1} = v_{mm+1} / t_m \quad , \quad \bar{R}_{m+1m+2} = u_{mm+1} \bar{D}_{m+1}.$$

Veamos ahora como se realiza la factorización cuando el renglón de bloqueo es tal que su índice r ≤ m+1, esto es, cuando la variable de bloqueo es alguna z_1. Para eso,supóngase que en un momento dado la submatríz $\left(M_{11} \mid M_{1r}\right)$ tiene la forma:

$$M = \left(\begin{array}{ccc|c} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{array}\right). \qquad (2.37)$$

Supóngase que el renglón r de bloqueo es el 2 ≤ m+1=3. Como ya vimos, lo que se hace es trasladar tanto la columna como el renglón r=2 de la matríz (2.37) a los últimos lugares respectivamente (m+2 y m+1), de esta forma la submatríz de M a factorizar es:

$$\tilde{M} = \begin{pmatrix} m_{11} & m_{13} & m_{14} \\ m_{31} & m_{33} & m_{34} \\ m_{21} & m_{23} & m_{24} \end{pmatrix} \qquad (2.38)$$

$$= V'^{t} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_3 & 0 \\ 0 & 0 & d_2 \end{pmatrix} \begin{pmatrix} 1 & R_{13} & R_{14} \\ 0 & 1 & R_{34} \\ 0 & R_{23} & R_{24} \end{pmatrix}$$

$$= V'^{t} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_3 & 0 \\ 0 & p_2 & d_2 \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & p_3 \end{pmatrix} \begin{pmatrix} 1 & R_{13} & R_{14} \\ 0 & 1 & R_{34} \\ 0 & 0 & 1 \end{pmatrix}$$

$$\tilde{V}\ \tilde{M}\ M'_{1r} = \tilde{V} e_7^1$$

donde, p_2 y p_3 tienen las caracteristicas ya descritas y V'^{t} es la matríz que se obtiene de V^{t} al haber trasladado el renglón y la columna r=2 a los últimos lugares.

Una vez hecho esto la factorización se lleva a cabo de la misma manera que en el algoritmo agrega. Sin embargo, no es necesario comenzarla desde el principio sino desde el renglón r=2. Supóngase que $\tilde{M}$ ya ha sido factorizada, quedándonos como:

$$\tilde{M}=\tilde{V}^t\tilde{D}\ \tilde{R}$$

como se vió en (2.24), se necesita el producto Ve_1. Sin embargo no se tiene almacenada la matríz V, La forma en que se evita tal almacenamiento es la siguiente: se tiene que

$$\tilde{M}^t e_1=M_1^t$$

y $$\tilde{M}^t=\tilde{R}^t\tilde{D}\ \tilde{V}. \qquad (2.39)$$

Multiplicando (2.39) por e_1, $\tilde{M}^t e_1 =\tilde{R}^t\tilde{D}\ \tilde{V}\ e_1$, lo que implica que

$$\tilde{M}_1^t = \tilde{R}^t(\tilde{D}\ \tilde{V}\ e_1). \qquad (2.40)$$

Si ponemos $\tilde{p} = \tilde{D}\ \tilde{V}\ e_1$, entonces, $\tilde{V}\ e_1 = \tilde{D}^{-1}\tilde{p}$, donde el valor de $\tilde{p}$ se obtiene de (2.40) como $\tilde{R}^t\tilde{p} = \tilde{M}_1^t$. Además, puesto que $\tilde{V}^t\tilde{D}\tilde{V}=I$. entonces $\tilde{V}^t\tilde{p} = e_1$. Tenemos pués como determinar $\tilde{V}\ e_1$. Nos queda por ver como factorizar la matríz que se obtiene al quitar el renglón 1 de (2.38). Tal matríz es la siguiente:

$$\begin{pmatrix} m_{31} & m_{33} & m_{34} \\ m_{21} & m_{23} & m_{24} \end{pmatrix}$$

$$=\begin{pmatrix} M'' & M''_{m+1} & M''_{m+2} \end{pmatrix},$$

$$=\left(I - e_1 e_1^t\right)\tilde{M}$$

$$= \left(I \; - \; e_1 e_1^t \right) \tilde{V}^t \tilde{D} \; \tilde{R}.$$

Pero como $\tilde{V}^t \tilde{p} = e_1$, se tiene,

$$= \left(I \; - \; \tilde{V}^t \tilde{p} \; \tilde{p}^t \tilde{V} \right) \tilde{V}^t \tilde{D} \; \tilde{R},$$

$$= \tilde{V}^t \tilde{D} \; \tilde{R} \; - \; \tilde{V}^t \tilde{p} \; \tilde{p}^t \tilde{V} \; \tilde{V}^t \tilde{D} \; \tilde{R},$$

y ya que $\tilde{V}^t \tilde{V}^- = \tilde{D}^{-1}$,

$$= \tilde{V}^t \tilde{D} \; \tilde{R} \; - \; \tilde{V}^t \; \tilde{p} \; \tilde{p}^t \; \tilde{R},$$

$$= \tilde{V}^t \left(\tilde{D} \; - \; \tilde{p} \; \tilde{p}^t \right) \; \tilde{R},$$

aplicando I.3 de Sargent[16],

$$= \tilde{V}^t \left(\hat{V}^t \; D \; \{ U \; \tilde{p}_s \; \beta \} \right) \; \tilde{R},$$

(donde $\tilde{p}_s \neq 0$, $p_i = 0$ si $i > s$)

$$= V^t D \left(R \;\; R_{m+1} \;\; R_{m+2} \right) \; ,$$

las operaciones para realizar todo lo anterior se pueden calcular usando I.4 e I.5 en Sargent [16].

La parte 2 del algoritmo de Sargent que se usa para factorizar la matriz M, a la que se le quitan el renglón y la columna r, empieza trasladando tal renglón y columa a los últimos lugares y el cálculo comienza a partir del renglón r la descripción del algoritmo es la siguiente:

A L G O R I T M O E L I M I N A

1.-Poner $t_{r-1}=D_r$,

$v_{r-1j} = D_r\bar{R}'_{rj}$ $\qquad j=r,\ldots,m+1$

($\bar{R}'$ se obtiene de R al trasladar la columna r al último lugar).

2.-Para i=r,...,m , calcular

$p_i=v_{i-1j}$, $\qquad t_i=t_{i-1} + p_i^2/\bar{D}_i$

$\tilde{D}_i = \bar{D}_i t_i/t_{i-1}$, $\qquad \beta_i = p_i/\tilde{D}_i t_{i-1}$

($\bar{D}$ se obtiene de la matriz diagonal D al pasar el elemento D_r al último lugar)

$v_{ij} = v_{i-1j} - p_i\bar{R}_{ij}$ $\qquad j=i+1,\ldots,m+1$

($\bar{R}$ se obtiene de $\bar{R}'$al trasladar el renglón r al último lugar)

$\tilde{R}_{ij} = \bar{R}_{ij} + \beta_i v_{ij}$ $\qquad j=i+1,\ldots,m+1$

3.- Poner $p_{m+1} = v_{mm+1}$, $\qquad \tilde{D}_{m+1} = p_{m+1}^2/t_m$

Calculado lo anterior, se efectua la actualización del vector M_r y para eso se necesita el vector $\tilde{V}e_k$, donde k es el índice del renglón donde se encuentra la variable $_iW$ de conducción, el cual se puede obtener, como ya vimos de la siguiente forma:

$$\tilde{V}e_k = \bar{D}^{-1}\,\tilde{p} \text{ , donde, } \tilde{R}^t\,\tilde{p} = \tilde{M}_k^t \text{ .} \qquad (2.42)$$

4.-Calcular $\tilde{p}$, M'_r y q' usando (2.42),(2.40) y (2.28)

5.-Si $\tilde{p}_{m+1} \neq 0$ poner s= m+1 e ir al paso 7;

en caso contrario s=m

6.-Si $\tilde{p}_s = 0$, poner $D_s = \tilde{D}_s$, $R_{sj} = \tilde{R}_{sj}$, j=s+1,...,m+1,

poner s=s-1, y repetir 6.

en caso contrario, poner:

$$v_{sj} = R_{sj} = 0, \qquad j=s+1,\ldots,m,$$

$$v_{sm+1} = 0, \quad R_{sm+1} = 1, \quad D_s = \tilde{D}_{m+1}$$

7.- poner $t_{s-1} = \tilde{p}_s^2 / \tilde{D}_s$

8.- para i=s-1,...,1

$$v_{ij+1} = \tilde{p}_{i+1}, \qquad t_{i-1} = t_i + \tilde{p}_i^2 / \tilde{D}_i$$

$$D_i = \tilde{D}_i t_i / t_{i-1}, \qquad \beta_i = -\tilde{p}_i / \tilde{D}_i t_i$$

$$v_{ij} = v_{i+1j} + \tilde{p}_{i+1}\tilde{R}_{i+1j}, \quad j=i+2,\ldots,m+1$$

$$R_{ij} = \tilde{R}_{ij} + \beta_i v_{ij}, \quad j=i+1,\ldots,m+1.$$

Si $\tilde{p}_{m+1}=0$, entonces R es singular, sin embargo en la próxima iteración se restaura la no singularidad de R de la siguiente forma:

Si en la siguiente iteración se eliminan el renglón y la columna r, el renglón es transferido como antes al renglón m+1, y hecho esto, se intercambia con el renglón s-1. La forma triangular superior R es restaurada aplicando los pasos 1, 2, y 3 de elimina del renglón r al renglón s-2, al terminar el paso 3, $R_{s-1j} = v_{s-2j} / p_{s-1}$ para j=s,...,m+1, y después se siguen aplicando los siguientes pasos de elimina tal como están dados.

Si por el contrario, en la siguiente iteración, se agrega un renglón y una columna, se pone igual que antes, el renglón r en el lugar m+1; hecho esto, se intercambian los renglones s y m+1 y se aplica el algoritmo agrega unicamente a los primeros s-1 renglones además,

$$\tilde{D}_{m+1} = D_s, \; D_s = v_{s-1s}^2 / t_{s-1}, \; R_{sj} = v_{s-1j} / v_{s-1s}, \quad j=s+1,..,m+1$$

y desde luego hay que calcular $\bar{R}_{im+2}$, para i=1,...,m+1.

CAPITULO

III

ALGORITMO DE GOLDFARB E IDNANI

El problema a resolver, es nuevamente un problema de programación cuadrática, sujeto a desigualdades lineales. En este caso, sin embargo,el problema es estrictamente convexo y además las componentes del vector x solución, no tienen que ser no negativas, así que el problema se puede escribir como:

$$\text{minimizar } f(x)=c^t x + 1/2x^t Dx \qquad (3.1.a)$$

$$\text{sujeto a } S(x)=A^t x-b\geq 0 \qquad (3.1.b)$$

$$(x \text{ no necesariamente}\geq 0)$$

Donde $D \in \mathbb{R}^{nxn}$, $A \in \mathbb{R}^{nxm}$, $b \in \mathbb{R}^m$, y x, $c \in \mathbb{R}^n$.

El método generado por Goldfarb e Idnani para la solución de (3.1) consiste, básicamente, en ir minimizando una secuencia de subproblemas del problema (3.1). El primer subproblema de tal secuencia es la minimización de la función objetivo sin restricciones (3.1.a) y los siguientes términos de la secuencia son tales que, la secuencia es estrictamente creciente con respecto a los valores óptimos de la función objetivo; esto es, si f_i y f_{i+1} son los valores óptimos de la función objetivo f para los subproblemas i e i+1 respectivamente, entonces, $f_i<f_{i+1}$.

Puesto que el mínimo sin restricciones de f pertenece al espacio dual del problema (3.1), el primer subproblema de la secuencia (como ya se dijo), consiste en la minimización de la función objetivo f sin restricciones. Una vez que se tiene la solución x de este subproblema, se checa si ésta satisface todas las restricciones (3.1.b). Si x cumple con ellas, el algoritmo termina, ya que esta solución es óptima para el dual y factible para el problema primal. En caso de que alguna o algunas de las

restricciones no sean satisfechas (vamos a decir que están violadas por x), se toma una de ellas y el nuevo subproblema de la secuencia a optimizar es el que tiene por función objetivo a f y además, contiene a la restricción violada. Nuevamente, ya que se encuentra la solución x de este nuevo subproblema se checa la factibilidad de x para (3.1.b). Si x satisface las restricciones, x es la solución. En caso contrario, se vuelve a aplicar el algoritmo, agregando una nueva restricción violada, hasta la solución de (3.1).

Algunos conceptos que usaremos para la descripción del algoritmo son los siguientes:

$S_j(x)=n_j^t x-b_j$, es la restricción j-ésima de (3.1.b), donde, n_j es la columna j de la matríz A.

Vamos a decir que S_j es activa en x si $S_j(x)=0$. La letra K denota al conjunto (1,2,...,m), de índices de las restricciones de (3.1). Al conjunto de índices de las restricciones activas en x se le denota con E. Si J es un subconjunto cualquiera de K, entonces P(J) denota al subproblema de (3.1) sujeto al conjunto de restricciones con índice en J. De esta forma, si $J=\emptyset$, $P(\emptyset)$ denota al subproblema de (3.1) donde hay que encontrar el mínimo de f sin restricciones.

Un conjunto de restricciones linealmente independientes, son aquellas cuyas normales son linealmente independientes. El conjunto de índices E siempre estará formado por índices de restricciones linealmente independientes.

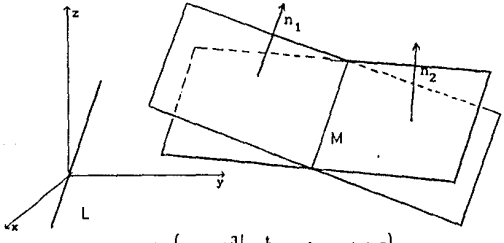
Si la solución x de un subproblema P(J), satisface un conjunto activo de restricciones linealmente independientes con índices en E, y $E \subset J$, a la pareja (x,E) se le llama un S-par (un par solución) y cada vez que tengamos un S-par y una restricción p violada por la x del S-par, se nos formará una V-triada (x,E,p). La matríz cuyas columnas son los vectores normales a las restricciones con índice en E, se denota con N y la cardinalidad de E se denota con q. Se denota con E^+ al conjunto $E \cup (p)$ donde,

p es un elemento de K-E; E^- denota un subconjunto de E cuya cardinalidad es q-1. Así mismo, N^+ y N^- denotan las matrices cuyas columnas son los vectores n_i de las restricciones con índices en E^+ y E^-; n^+ se usa para denotar al vector normal n agregado a N para formar N^+ y, n^- indica la columna quitada a N para formar N^-.

Sea M la variedad lineal formada por la intersección de las restricciones S_j, $\forall\ j \in E$; esto es

$$M = \left\{ x \,\middle|\, n_j^t x - b_j = 0, \forall\ j \in E \right\}.$$

Es claro que cada uno de los $n_j \in N$ es ortogonal a M y por tanto $N \perp M$. Al espacio vectorial paralelo a M lo denotaremos con la letra L y entonces, $N^t \perp L$ (ver figura).



$$M = \left\{ x \in \mathbb{R}^3 \,\middle|\, n_i^t x = b_i\ ,\ i=1,2 \right\}$$

Como es bien sabido, una condición necesaria y suficiente para que un punto $x \in M$ sea el mínimo de f sujeto a las restricciones con índice en E, es que el gradiente, $g(x)=\nabla f(x)=c+Dx$, sea ortogonal a M; esto es, que $g(x)=Nu(x)$, donde como ya se dijo, N es la matriz de dimensión nxq con columnas n_i, para $i \in E$ y, u(x) es el vector de los multiplicadores de Lagrange con componentes ≥ 0.

Sea $\tilde{x}$ un punto cualquiera en M. Se desea llegar a partir de $\tilde{x}$ al punto x óptimo de f sobre M. Para esto, supongamos que $g=g(x)$ y $\tilde{g}=g(\tilde{x})$; se observa entonces que, $g-\tilde{g}=D(x-\tilde{x})$ y ya que $g(x)=Nu(x)$, entonces,

$$x-\tilde{x}=D^{-1}(g-\tilde{g})=D^{-1}Nu(x)-D^{-1}\tilde{g}. \qquad (3.2)$$

Hay que observar, que $x-\tilde{x}$ está en el espacio L y, puesto que $L \perp N^t$, $N^t(x-\tilde{x})=N^tD^{-1}Nu(x)-N^tD^{-1}\tilde{g}=0$. Si de esta expresión despejamos u(x) tenemos que:

$$u(x)=(N^tD^{-1}N)^{-1}N^tD^{-1}\tilde{g}.$$

Si denotamos por $N^*=(N^tD^{-1}N)^{-1}N^tD^{-1}$, sustituyendo u(x) en (3.2) y realizando las operaciones correspondientes, se obtiene

$$x-\tilde{x}=-D^{-1}(I-NN^*)\tilde{g}. \qquad (3.3)$$

A la matriz N^* se le conoce como la inversa generalizada de N de Moore-Penrose con respecto a D; es mas, si $H=D^{-1}(I-NN^*)$, entonces $x=\tilde{x}-H\tilde{g}$.

Una propiedad de la matriz H es su simetría así que $(N^tH)^t=HN=D^{-1}(N-NN^*N)=D^{-1}(N-N)=0$, lo que nos indica que $H \perp N$. Si $v=Hw$, entonces $N^tv=0$, lo que nos dice que cualquier vector de la forma Hw es ortogonal a las columnas de N y entonces Hw debe estar en L; esto es, H proyecta cualquier vector w en el espacio vectorial L. Otras dos propiedades de H son: que es positiva semidefinida y que HDH=H (D. Goldfarb [9]).

Al describir el algoritmo, se supone que las columnas de N son linealmente independientes.

DESCRIPCION DEL ALGORITMO

En el caso del algoritmo de Lemke, no es necesario contar con un punto inicial. Sin embargo, el algoritmo de Goldfarb si lo necesita, ya que el punto inicial debe ser dual factible y el mínimo sin restricciones del problema (3.1) es un punto que cumple con esta condición. Este será nuestro punto inicial y, por tanto, el primer subproblema de la secuencia es P(∅). Como ya se sabe, la solución de este subproblema es: $x_0=-c^tD^{-1}$ y entonces el primer S-par es (x_0,E) con $E=\emptyset$. Una vez obtenido el punto inicial, evaluamos $S_j(x_0)$, $\forall \quad j \in K$. Si $S_j(x_0)\geq 0 \ \forall \ j \in K$, la solución de (3.1) está en x_0, pués es óptimo para el dual y factible para el primal. En caso de que $S_p(x_0)<0$, para alguna $p \in K$, entonces la restricción p es una restricción violada en el punto x_0 y, por lo tanto, x_0 no es factible para el problema primal.

Debemos ahora, encontrar a partir de x_0, un punto x en el cual cuando menos, $S_p(x)=0$. Tomando en cuenta esto, el nuevo subproblema a resolver es, P(E) con $E=(p)$. Para resolver este subproblema, tomemos en cuenta lo siguiente: puesto que f está sujeta unicamente a la restricción p, entonces el lagrangiano es,

$$\begin{aligned}(x,u)&=f(x)-uS_p(x)\\&=f(x)-u(n_p^tx-b_p).\end{aligned}$$

En el mínimo de f sujeto a $S_p(x)=0$ se debe cumplir que ,

$$\nabla_x (x,u)=g(x)-un_p= c+Dx-un_p=0 \qquad (3.4)$$

$$\text{y} \qquad n_p^t x-b_p=0. \qquad (3.5)$$

Despejando x de (3.4) se tiene que $x=-D^{-1}c+uD^{-1}n_p$ y ya que $x_0=-D^{-1}c$, entonces

$$x=x_0+uD^{-1}n_p. \qquad (3.6)$$

Multiplicando (3.6) por n_p^t y sustituyendo en (3.5) se puede despejar u para obtener

$$u=\left(-n_p^t x_0+b_p\right)\div\left(n_p^t\cdot D^{-1}n_p\right)$$

$$=\left(-S_p(x_0)\right)\div\left(n_p^t\ D^{-1}n_p\right).$$

Ya que $-S_p(x_0)>0$ (pues la restricción p es violada en x_0) y $n_p^t D^{-1}n_p>0$ (D es positiva definida) entonces, u>0.

Es claro que el punto x, encontrado de esta forma, no solo cumple con $S_p(x)=0$, sino que además x es el mínimo de f sujeto a que $S_p(x)=0$ y el multiplicador u (que es variable dual) es positivo. Si E=(p)Uω, el S-par correspondiente al subproblema P(E) es (x,E), con x obtenida mediante (3.6); además N está formada unicamente por el vector n_p y la primera variedad lineal que se tiene es $M=\left\{x \mid n_p^t x-b_p=0\right\}$. Nuevamente, evaluamos $S_j(x)$, $\forall$ j $\in$ K-E; si $S_j(x)\geq 0$ $\forall$ j $\in$ K-E , el algoritmo termina, pues la x así encontrada es óptima y factible. En caso de que $S_p<0$, para alguna p $\in$ K-E, la restricción p es violada por la x actual, lo que implica que x no es factible para el problema primal. De acuerdo al algoritmo, hay que localizar un nuevo punto x_1, en el que, $S_p(x_1)=0$ y además, x_1 óptimize f sobre las restricciones con índice en E U (p).

En general, debemos encontrar una dirección de movimiento z y una longitud de paso t, de forma tal que ambas nos permitan llegar al punto x_1 buscado. Para encontrar la dirección, se procede de la siguiente forma. Sea n_p el vector ortogonal a la restricción S_p. Como la matriz H proyecta cualquier vector sobre el espacio vectorial L, el cual es paralelo a la variedad lineal M

y además ortogonal a N, si tomamos $z=Hn_p$, entonces z será un vector paralelo a M y ortogonal a N lo que implica que, z es una dirección factible.

Esta selección de la dirección z nos permite que el punto $x_1=x+tz$ sea un punto que esté en M que, como ya se sabe, contiene a las x factibles de las restricciones activas; de aquí que el punto x_1 así obtenido es tal que, $S_i(x_1)=0$, $\forall\ i \in E$ i.e; está selección del punto x_1, permite que las restricciones activas lo sigan siendo. Queda entonces, por calcular la magnitud del paso t, que permita que además $S_p(x_1)=0$. Tenemos que

$$S_p(x_1)=S_p(x+tz)=n_p^t(x+tz)-b_p$$

$$=n_p^tx-b_p+tn_p^tz=S_p(x)+tn_p^tz,$$

e igualando a cero y despejando t se obtiene

$$t=t_2=-S_p(x)/n_p^tz. \qquad (3.7)$$

Ahora bién, considerando que $z=Hn_p$, con H positiva semidefinida (ya que $Hw=0 \Leftrightarrow w=Nv$) y puesto que n_p no es combinación lineal de N, entonces $n_p^tz=n_p^tHn_p \geq 0$, esto junto con el hecho de que $-S_p(x)> 0$, nos garantiza que $t=t_2>0$.

Sin embargo, la x_1 así obtenida puede romper con la factibilidad de los multiplicadores de Lagrange asociados a las S_i activas y ya que el algoritmo es dual, hay que conservar la factibilidad de las variables duales. Para observar como se puede

romper la factibilidad de las variables duales, supóngase lo siguiente:

Si x_1 es el óptimo de f sujeto a las restricciones en $EU(p)=E^*$ debe suceder que $g(x_1)=N^+u^+(x_1)$, o lo que es lo mismo $(N^+)^*g(x_1)=u^+(x_1)$. Desarrollando esta expresión se produce lo siguiente: $u^+(x_1)=(N^+)^*g(x+tz)=(N^+)^*[c+D(x+tz)]=(N^+)^*[c+Dx+tDz]$ y ya que $z=Hn_p$ lo anterior es igual a $(N^+)^*[c+Dx+tDHn_p]$ y como $H=D^{-1}(I-NN^*)$, entonces, $(N^+)^*[c+Dx+tDHn_p]=(N^+)^*[g(x)+t(n_p-NN^*n_p)]$, si hacemos $r=N^*n_p$, entonces, lo anterior se puede poner como

$$u^+(x_1) = (N^+)^*[g(x)+t(n_p-Nr)]$$

$$=(N^+)^*\left[g(x)+t[-N\,|\,n_p]\begin{pmatrix}r\\1\end{pmatrix}\right]$$

$$=(N^+)^*\left[g(x)+tN^+\begin{pmatrix}-r\\1\end{pmatrix}\right]$$

entonces $$u^+(x_1)=(N^+)^*g(x)+t\begin{pmatrix}-r\\1\end{pmatrix}. \qquad (3.9)$$

Es más, se puede probar usando Goldfarb (Lootsma pag(241))[7] que:

$$u^+(x)=(N^+)^*g(x)\geq 0, \qquad \begin{array}{l}\text{(mayor si } S_j \text{ es activa}\\ \text{cero si } S_j \text{ no lo es)}\end{array}$$

de aquí que (3.9) queda como

$$u^+(x_1)=u^+(x)+t\begin{pmatrix}-r\\1\end{pmatrix}. \qquad (3.10)$$

De esta forma, si t es demasiado grande, y $r_j>0$ para alguna j, puede suceder que $u_j^+(x_1)< 0$, rompiendo la factibilidad

de las variables duales.

Vamos a suponer que E ≠ ∅. Desarrollando (3.10), se tiene que

$$\left(u_1^+(x_1),u_2^+(x_1),\ldots,u_q^+(x_1),u_{q+1}^+(x_1)\right)=$$

$$=\left(u_1^+(x),u_2^+(x),\ldots,u_q^+(x),0\right)+t(-r_1,-r_2,\ldots,-r_q,1),$$

(donde q es la cardinalidad de E), lo que implica que $u_j^+(x_1)=u_j(x)-r_jt$. Tomando en cuenta que $u_j^+(x_1)\geq 0$ para conservar la factibilidad, la t que se escoja debe ser tal que:

$$u_j^+(x_1)=u_j^+(x)-tr_j\geq 0 \quad \forall \quad j=1,\ldots,q \qquad (3.11)$$

Para garantizar que la t satisfaga (3.11), tomamos $t=u_j^+(x)/r_j$, $\forall\ j\in E$ y $r_j>0$, de esta forma tendríamos que $t>0$ ya que $u_j^+(x)>0$. Sin embargo, de todos estos posibles valores de t debemos tomar aquel en que el vector $u^+(x_1)\geq 0$. De aquí que para que (3.10) quede no negativo, la selección de t es:

$$t=t_1=\min_{r_j>0}\left\{\frac{u_j}{r_j}\right\}=\frac{u_k^+}{r_k} \qquad 1\leq j\leq q. \qquad (3.12)$$

De (3.12) y de la anterior forma de seleccionar t en (3.7), se tiene que el valor óptimo de t debe ser:

$$t=\min\{t_1,t_2\}. \qquad (3.13)$$

Si la t mínima es t_1, entonces la k-ésima componente del vector $u^+(x_1)$ es cero, ya que, $u_k^+(x_1)=u_k^+(x)-(u_k(x)/r_k)r_k=0$; o sea el multiplicador $u_k^+(x_1)$, asociado a S_k, vale cero, y la restricción S_k, debe desactivarse y por lo tanto la quitamos del conjunto de restricciones activas, lo que implica que ahora

tendremos, $E^-=E-(k)$ y $q=q-1$. Además, al quitar la restricción k se tiene una nueva $H=H^-$ y una nueva $N=N^-$. Al haber seleccionado $t=t_1$, se produjo un punto nuevo, a saber, $x_1=x+tz$ el cual no cumple con que $S_p(x_1)=0$ (ya que $t_1<t_2$). Si hacemos $x=x_1$ con la H nueva volvemos a encontrar una nueva dirección y una nueva longitud de paso. De acuerdo a (3.13), esto nos producirá un nuevo ciclo y ya que el ciclo anterior no produjo una x_1 óptima se le llamará ciclo parcial.

Si $t=t_2\leq t_1$, el punto $x_1=x+t_2z$ es tal que $S_p(x_1)=0$ y además cada una de las componentes del vector $u^+(x_1)$ es >0; es más, $u^+(x_1)=(N^+)^*g(x_1)$, de aquí que x_1 es una solución óptima del subproblema P(E) con $E=EU\{p\}$ y, por lo tanto el par (x_1,E) es un S-par. Puesto que en este ciclo la x_1 producida es óptima para el subproblema P(E), le llamaremos ciclo total. Como ya que se ha activado a la restricción p, tenemos que $H=H^+$, $N=N^+$ y $q=q+1$.

Antes de proseguir, debemos observar que el número de ciclos parciales es finito, ya que en cada ciclo parcial se quita una restricción del conjunto activo, por lo que a lo más se tendrán q-1 ciclos parciales. Además, como en cada ciclo se requiere que las variables duales sean no negativas, entonces tenemos un método de tipo dual.

Para obtener t_1 es necesario que al menos una componente del vector r sea positiva. Si sucede que el vector $r\leq 0$, entonces, necesariamente $t=t_2$.

Es posible que el problema (3.1) no tenga solución. Para saber como el algoritmo puede detectar este caso, supóngase que el vector n_p correspondiente a la nueva restricción violada S_p, sea una combinación lineal de las columnas de la matríz N; esto es,

$$n_p=Nr. \qquad (3.14)$$

Supóngase también que el vector $r\leq 0$ y que x_1 es la solución para

el subproblema :

$$P(E \cup \{p\}), \tag{3.15}$$

puesto que x_1 es la solución de (3.15), entonces $S_p(x_1)=0$ y ya que en la solución x anterior $S_p(x)<0$; entonces $S_p(x_1)-S_p(x)>0$ lo que implica que, $n_p^t x_1 - b_p - n_p^t x + b_p > 0$ y por lo tanto $n_p^t(x_1-x)>0$. Pero como $x_1-x=tz$, ya que $t>0$, entonces

$$n_p^t z>0, \tag{3.16}$$

combinando (3.14) con (3.16) se tiene

$$n_p^t z = r^t N^t z>0, \tag{3.17}$$

pero se tenía que, $N^t z=N^t H n_p$ y H ı N por lo que, $N^t z=0$ lo que combinado con el hecho de que $r\leq 0$, hacen de (3.17) una contradicción. Así que si $n_p=Nr$ y $r\leq 0$, no existe solución para el problema (3.15) y por lo tanto, tampoco el problema (3.1) tiene solución.

Puede suceder también que $z=Hn_p=0$ (esto es, que n_p sea una combinación lineal de N) y $q\geq 0$, lo que nos dice, que a pesar de que no hay restricciones activas, no existe una dirección factible, por lo tanto el problema (3.1) no tiene solución.

Si en cambio $n_p=Nr$ (esto es $z=0$) pero al menos una componente r_k de r es positiva, se puede hacer lo siguiente: puesto que $n_p=Nr$, entonces, $n_p=n_1r_1+n_2r_2+\ldots+n_qr_q$ y ya que $r_k>0$, podemos poner n_k como:

$$n_k=\frac{1}{r_k}\left[-\sum_{i\in E^-} r_i n_i + n_p\right]. \tag{3.18}$$

Mas todavía, se sabe que (x,E), es un S=par ⇒ g(x)=Nu(x), entonces,

$$g(x)=u_1n_1+u_2n_2+\ldots+u_qn_q = \sum_{i\in E^-} u_in_i+u_kn_k \qquad \text{y usando(3.18)}$$

$$= \sum_{i\in E^-} u_in_i + \frac{u_k}{r_k}\left[-\sum_{i\in E^-} r_in_i+n_p\right]$$

$$\text{por lo tanto} \quad g(x)= \sum_{i\in E^-}\left(u_i-\frac{r_i}{r_k}u_k\right)n_i+\frac{u_k}{r_k}n_p$$

de donde es claro, hemos quitado la n_k y puesto la n_p.

Si se define $\hat{E}=E^-U(p)$, entonces las columnas de $\hat{N}$ son linealmente independientes; esto es, se ha quitado del conjunto activo a la restricción k y q=q-1. En este caso x no se mueve, sin embargo se actualizan H y N y se vuelve a calcular una dirección de paso con la nueva H y la n_p que ya tenemos. De todo lo anterior se puede decir que el algoritmo dado por Goldfarb e Idnani resuelve el problema (3.1) o bién indica que no existe solución factible, en un número finito de pasos.

Se mostrará ahora, que la función objetivo toma valores estrictamente crecientes en cada ciclo total.

Sea $x_1=x+tz$, donde la restricción n_p es ya activa. Aplicando el teorema de Taylor a f en x_1,

$$f(x_1)-f(x)=tz^tg(x)+\frac{1}{2}t^2z^tDz$$

pero

$$z^t g(x) = n_p^t H g(x) = n_p^t H N^t u^t(x) = n_p^t H n_p u_{q+1} \quad \text{(ya que } H \perp N\text{)}$$

$$= n_p^t z u_{q+1}(x)$$

y ya que $u_{q+1}(x) \geq 0$ y $z^t n_p = n_p^t H n_p = n_p^t H D H n_p = z^t D z > 0$, entonces, debe suceder que:

$$f(x_1) - f(x) = t z^t g(x) + \frac{1}{2} t^2 z^t D z > 0,$$

lo que nos garantiza que $f(x_1) > f(x)$ para z y t no cero.

A L G O R I T M O

0) Encontrar el mínimo sin restricciones:
poner $x=-D^{-1}c$, $f=\frac{1}{2}c^{t}x+f_0$, $H=D^{-1}$, $E=\emptyset$, $q=0$

1) Seleccionar una restricción violada, si la hay; calcular $S_j(x)$. Si $S_j(x)\geq 0 \ \forall \ j \notin E$, la x actual es la solución del problema, pués es óptima y factible.
En caso contrario, seleccionar $p \in K-E$ y poner $u=\begin{bmatrix} u \\ 0 \end{bmatrix}$; si $q=0$, poner $u=0$.

2) Determinarla solución óptima de un nuevo subproblema.

2a) Calcular $z=Hn_p$ (z=dirección de paso, p=índice de la restricción violada seleccionada.
Calcular $r=N^{*}n_p$ (la derivada direccional de u(x), a lo largo de z).
si $r\leq 0$ o $q=0$, poner $t_1=\infty$; en caso contrario, poner

$$t_1 = \min_{r_j>0} \left\{ \frac{u_j}{r_j} \right\} = \frac{u_k}{r_k} \qquad 1\leq j\leq q$$

2b) Si $z=0$ (quitar la restricción correspondiente), si $t_1=\infty$, el subproblema p(EU{p}) no tiene solución y por tanto el problema principal tampoco.
En caso contrario, quitar la restricción k (ver detalles de implementación más adelante); poner $E=E-\{k\}$, $q=q-1$, $u=u+t_1\begin{bmatrix} -r \\ 1 \end{bmatrix}$, quitar la componente k de u, actualizar H y N^{*}, y regrear al paso (2a).

2c) Si $z\neq 0$, calcular $t_2=-S_p(x)/z^{t}n_p$ y poner $t=\min(t_1,t_2)$, $x=x+tz$, $f=f+tz^{t}n_p(t/2+u_{q+1})$ y

$u=u+t\begin{bmatrix}-r\\1\end{bmatrix}$.

Si $t_2 \geq t_1$ se tiene un ciclo total, poner $E=E\cup\{p\}$,

(i.e., se agrega n_p ala matríz N) actualizar H y N^* (ver la implementación), e ir al paso (1).

En caso contrario, esto es, en caso de que $t_1<t_2$, se tiene un ciclo parcial. Poner, $E=E-\{k\}$, $q=q-1$, quitar la componente k de u, actualizar H y N^*, e ir al paso (2a).

I M P L E M E N T A C I O N

Al principio del algoritmo se tiene que encontrar el mínimo sin restricciones de la función f. Para eso, hay que calcular D^{-1}(paso 0 del algoritmo). Una manera de mejorar la estabilidad numérica es descomponiendo la matríz D en los factores de Cholesky (Strang [18]), i.e. $D=LL^t$. Como se requiere D^{-1}, es necesario encontrar, la inversa de L, lo que es muy sencillo ya que L es triangular inferior.

Al ir agregando o quitando los vectores n_p a la matríz N, de las restricciones activas, tenemos que actualizar tambien las matrices H y N^* (paso 2 del algoritmo). Para eso, se tiene que:

$$\begin{aligned} N^* &= (N^tD^{-1}N)^{-1}N^tD^{-1} \\ &= (N^tL^{-t}L^{-1}N)^{-1}N^tL^{-t}L^{-1}. \end{aligned} \tag{3.20}$$

Si llamamos $B=L^{-1}N$ (la implementación está basada en la descomposición de G en los factores de Cholesky y en la factorización QR de la matríz B) y factorizamos B en QR, entonces,

$$B=Q\begin{pmatrix}R\\0\end{pmatrix}=\begin{bmatrix}Q_1,Q_2\end{bmatrix}\begin{pmatrix}R\\0\end{pmatrix} \tag{3.21}$$

donde , Q es una matriz ortogonal de nxn partida en Q_1 y Q_2 , el número de columnas de Q_1 es q, o sea igual al número de restricciones activas, y R es una matríz triangular superior de qxq

Regresando a (3.20) y aplicando (3.21), tenemos:

$$N^*=(B^tB)^{-1}B^tL^{-1}$$

$$=\left[\left(Q\begin{bmatrix}R\\0\end{bmatrix}\right)^t\left(Q\begin{bmatrix}R\\0\end{bmatrix}\right)\right]^{-1}\left(Q\begin{bmatrix}R\\0\end{bmatrix}\right)^tL^{-1}$$

$$=\left([R^t\ 0]Q^tQ\begin{bmatrix}R\\0\end{bmatrix}\right)^{-1}\left([R^t\ 0]\begin{bmatrix}Q_1^t\\Q_2^t\end{bmatrix}\right)L^{-1}$$

$$=\left([R^t\ 0]\begin{bmatrix}R\\0\end{bmatrix}\right)^{-1}\left(R^t\ Q_1^t\right)L^{-1}$$

$$=(R^tR)^{-1}\left(R^tQ_1^t\right)L^{-1}$$

$$=R^{-1}R^{-t}R^tQ_1^tL^{-1}$$

de donde se obtiene

$$N^*=R^{-1}\left(Q_1^tL^{-1}\right). \qquad (3.22)$$

De la misma forma,

$$H=D^{-1}[I-NN^*]$$

$$=L^{-t}L^{-1}-L^{-t}L^{-1}NR^{-1}Q_1^tL^{-1}$$

$$=L^{-t}\left(I-BR^{-1}Q_1^t\right)L^{-1}$$

$$=L^{-t}\left(\left(Q_1|Q_2\right)\begin{bmatrix}Q_1^t\\Q_2^t\end{bmatrix}-Q\begin{bmatrix}R\\0\end{bmatrix}R^{-1}Q_1^t\right)L^{-1}$$

$$=L^{-t}\left(Q_1Q_1^t+Q_2Q_2^t-Q_1RR^{-1}Q_1^t\right)L^{-1}$$

$$=L^{-t}\left(Q_2Q_2^t\right)L^{-1}$$

y finalmente

$$H=\left(L^{-t}Q_2\right)\left(L^{-t}Q_2\right)^t. \qquad (3.23)$$

Si ponemos $$J=L^{-t}Q=\left(L^{-t}Q_1 \mid L^{-t}Q_2\right)=\left(J_1 \mid J_2\right) \qquad (3.24)$$

entonces $$H=J_2J_2^t \quad y \quad N^*=R^{-1}J_1^t. \qquad (3.25)$$

A pesar de que la implementación está basada en la factorización QR de la matriz B en (3.21), la matriz Q no es almacenada, en su lugar, se guarda y se actualiza la matriz $J=L^{-t}Q$ junto con las matrices L y R

Como vimos en el algoritmo, es necesario el cálculo de los vectores $z=Hn^+$ y $r=N^*n^+$ (paso 2a). Para hacer eso, sea $d=J^tn^+$ y aplicando (3.24), se tiene que

$$d=J^tn^+=\begin{bmatrix}J_1^t\\ J_2^t\end{bmatrix}n^+=\begin{bmatrix}d_1\\ d_2\end{bmatrix} \qquad (3.26)$$

que junto con (3.25) nos da

$$z=J_2d_2 \ y \ r=R^{-1}d_1. \qquad (3.27)$$

Se observa de (3.22) y (3.23), que para la actualización de H y N^* solo necesitamos actualizar J y R. La forma en que Goldfarb e Idnani hacen tal actualización es mediante el uso de matrices de Givens (Daniel [2], Gill [5]). Por su forma, el proceso de factorización que nos ocupa puede ser ilustrado para el caso de un vector de 2x2 $w=(w_1,w_2)^t$. En este caso, la matriz de Givens es $Q=\begin{bmatrix}c & s\\ s & -c\end{bmatrix}$ y se selecciona de forma tal, que transforme al vector w en el vector $(\omega,0)^t$, donde $\omega = \pm|w|$. Para esto, se realizan los siguientes cálculos,

$$\mu = \max(|w_1|,|w_2|), \quad \omega = sig(w_1)\mu\left[\left(w_1/\mu\right)^2+\left(w_2/\mu\right)^2\right]^{1/2}$$

$$c = w_1/\mu \quad y \quad s = w_2/\mu.$$

De acuerdo con Daniel [2], esto se realiza para evitar problemas de redondeo al elevar al cuadrado números muy grandes o muy pequeños y w=QR, donde $R=(\omega,0)^t$. Es más, si se tiene otro vector $y=(y_1,y_2)^t$ cualquiera, para calcular $\hat{y}=Qy$, de acuerdo con Gill [5] y Daniel [2] el cálculo se lleva a cabo como sigue:

$$v = \frac{w_2}{(\omega + w_1)} = s/(1+c)$$

$$\hat{y}_1 = cy_1 + cy_2$$

&

$$\hat{y}_2 = v(y_1 + \hat{y}_1) - y_2.$$

ADICIONANDO UNA RESTRICCION
(OBTENCION DE J^+)

Cuando una restricción p con normal n^+, es agregada al conjunto activo E, la factorización (3.21) es reemplazada por

$$B^+=L^{-1}N^+=L^{-1}[N|n^+]=[L^{-1}N|L^{-1}n^+]$$

$$=[B|L^{-1}n^+]=Q^+\begin{pmatrix}R\\0\end{pmatrix}. \qquad (3.28)$$

De (3.21), (3.24) y (3.26) se sigue que

$$Q^tB^+=Q^t\left(B\middle|L^{-1}n^+\right)$$

$$=\left(Q^tB\middle|Q^tL^{-1}n^+\right)$$

$$=\left(Q^tQ\begin{bmatrix}R\\0\end{bmatrix}\middle|\begin{matrix}d_1\\d_2\end{matrix}\right)=\begin{pmatrix}R & d_1\\0 & d_2\end{pmatrix}.$$

De esta forma, la factorización de B^+ (3.28) se obtiene como:

$$Q^+=\begin{pmatrix}I_q & 0\\0 & \bar{Q}^t\end{pmatrix} \qquad \text{y} \qquad R^+=\begin{pmatrix}R & d_1\\0 & \delta\end{pmatrix} \qquad (3.29)$$

donde, $\delta=\|d_2\|$ y $\bar{Q} = Q_{12}\cdot Q_{23}\cdot\cdots\cdot Q_{n-q-1,n-q}$ es el producto de las matrices de Givens seleccionadas de forma tal que

$$\bar{Q}d_2=\delta e_1.$$

Todavía más,

$$J^t = L^t Q^t = L^t Q \begin{pmatrix} I_q & 0 \\ 0 & \bar{Q}^t \end{pmatrix} = \left(L^t Q_1 \middle| L^t Q_2 \right) \begin{pmatrix} I_q & 0 \\ 0 & \bar{Q}^t \end{pmatrix} =$$

$$= \left(L^t Q_1 \middle| L^t Q_2 \bar{Q}^t \right) = \left(J_1 \middle| J_2 \bar{Q}^t \right) = \left(J_1^t \middle| J_2^t \right)$$

(para obtener $\bar{Q}$ se puede hacer uso de una matriz de Householder, sin embargo, al realizar el producto de esta con J_2 el gasto operacional es muy grande)

ELIMINANDO UNA RESTRICCION
(OBTENCION DE J^-)

Supóngase ahora, que se quiere quitar una columna de la matriz N. Como ya se dijo, lo que hay que actualizar son las matrices R y J. Supongamos que la matríz R es de 6x6 .i.e,

$$R=\begin{pmatrix} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{pmatrix}.$$
$$\uparrow$$

Vamos a quitarle la columna k=3, quedándonos la matriz:

$$\hat{R}=\begin{pmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{pmatrix} \leftarrow$$

que es una matriz de Hessenberg a partir del renglón 4. como $B=L^{-1}N=QR=[Q_1|Q_2]\begin{pmatrix} R \\ 0 \end{pmatrix}$, se tiene

$$Q_1^t B^- = Q_1^t L^{-1} N^- = \begin{pmatrix} R_1 & S \\ 0 & T \end{pmatrix}$$

donde la matriz particionada $\begin{pmatrix} R_1 & S \\ 0 & T \end{pmatrix}$, es la misma que R sin la columna k. Es claro que T es de dimensión [(6-3+1)x(6-3)],i.e, T es una matriz de Hessenberg de dimensión 4x3.

Para triangularizar $\hat{R}$ basta triangularizar T. Lo que haremos será seleccionar una secuencia de matrices de Givens de forma tal que introduzcan ceros bajo la diagonal de T para producirnos:

$GT=R_2$, donde $G=G_{q-1,q}\cdot\ldots G_{k,k+1}$ y desde luego las matrices $G_{q-1,q}, G_{q-2,q-1}, \ldots, G_{k,k-1}$ son las matrices de Givens correspondientes.

De esta forma, $R^{-}=\begin{pmatrix} R_1 & S \\ 0 & R_2 \end{pmatrix}$

Puesto que $J=\left(L^{-t}Q_1 \middle| L^{-t}Q_2\right)$ unicamente hay que transformar Q_1 de Q al hacer la reducción, entonces,

$$J^{-}=J\begin{pmatrix} I_{k-1} & 0 & 0 \\ 0 & G^{t} & 0 \\ 0 & 0 & I_{n-q} \end{pmatrix}$$

lográndose lo que se quería, la actualización de J^{-}. Falta todavía, efectuar la actualización del vector d en (3.26) para encontrar los vectores z y r en (3.27). Pero esto se puede hacer, aplicándole al vector d la misma operación que se usa para encontrar J^{-} y de esta manera encontrar z y r, terminando de esta forma, la actualización de las matrices R y J y el cálculo de z y r, cuando se elimina una columna.

CAPITULO IV

RESULTADOS NUMERICOS

En este capítulo, se exponen los resultados numéricos de los tres algoritmos que se estudiaron: El método de Lemke, la implementación que de éste hace Sargent y el método de Goldfarb.

Todas las corridas se realizaron en la computadora A-12 de *UNISYS* en doble precisión y el conteo de operaciones, contempla número de sumas+ restas+ multiplicaciones + divisiones y cada raíz cuadrada vale diez operaciones.

Los primeros 6 problemas fueron tomados de la literatura [11][9]. Como ya se dijo, el método de Lemke, está diseñado para problemas que restringen sus variables a valores no negativos, $x \geq 0$, lo que no ocurre con el método de Goldfarb-Idnani(G-I). Al correr estos 6 primeros problemas, se observó que en 2 de ellos, el 4 y el 5, la solución a la que se llega con el método de Goldfarb-Idnani, no es no negativa. Para subsanar tal situación, se agregaron al número de restricciones del problema las cotas $x_i \geq 0$ para $i=1,\ldots,n$. De aquí que cada vez que el método de G-I sea usado y las soluciones se requieran no negativas ,habrá que aumentar al conjunto de restricciones, las cotas $x_i \geq 0$ para $i=1,\ldots,n$. En el caso de Lemke (Sargent) esto no es necesario. Los problemas resueltos son los siguientes:

	G	A	c	b
1)	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} -2 \\ -2 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 4 \end{pmatrix}$
2)	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 4 \end{pmatrix}$

	G	A	c	b
3)	$\begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 6 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$
4)	$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 1 \\ -2 & 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -5 \\ -5 \\ -21 \\ 7 \end{pmatrix}$	$\begin{pmatrix} -8 \\ -10 \\ -5 \end{pmatrix}$
5)	$\begin{pmatrix} 2 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 & -2 & -1 & -1 \\ -3 & -1 & -2 & 1 \\ 0 & 1 & 4 & 0 \end{pmatrix}$	$\begin{pmatrix} -1 \\ -3 \\ 1 \\ -1 \end{pmatrix}$	$\begin{pmatrix} -5 \\ -4 \\ -1 \end{pmatrix}$
6)	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -4 & -3 & 0 \\ 2 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -5 \\ 0 \end{pmatrix}$	$\begin{pmatrix} -8 \\ 2 \\ 0 \end{pmatrix}$

Donde:

G=matríz del término cuadrático
A=Matríz de las restricciones
c=vector de la x lineal
b=vector de los términos independientes

cuya solución es:

1).- $(x_1,x_2)=(2,2)$.
2).- $(x_1,x_2)=(2,2)$.
3).- $(x_1,x_2)=(.5,1.5)$
4).- $(x_1,x_2,x_3,x_4)=(2.5,2.5,5.25,0)$
5).- $(x_1,x_2,x_3,x_4)=(0.\overline{27},2.\overline{09},0,0.\overline{54})$
6).- $(x_1,x_2,x_3)=(1/21)(10,22,44)$

Al aplicar el método de G-I, a los problemas 4) y 5), la matríz A de las restricciones y el vector b de los términos

independientes se modifican, quedandonos 4) y 5) como:

5) $\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$ $\begin{pmatrix} -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 1 \\ -2 & 1 & 0 & 1 \\ \multicolumn{4}{c}{I_{4x4}} \end{pmatrix}$ $\begin{pmatrix} -5 \\ -5 \\ -21 \\ 7 \end{pmatrix}$ $\begin{pmatrix} -8 \\ -10 \\ -5 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

6) $\begin{pmatrix} 2 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$ $\begin{pmatrix} -1 & -2 & -1 & -1 \\ -3 & -1 & -2 & 1 \\ 0 & 1 & 4 & 0 \\ \multicolumn{4}{c}{I_{4x4}} \end{pmatrix}$ $\begin{pmatrix} -1 \\ -3 \\ 1 \\ -1 \end{pmatrix}$ $\begin{pmatrix} -5 \\ -4 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

Los resultados para estos problemas, consideran el número de operaciones e iteraciones y se resumen en la tabla 1

	LEMKE		SARGENT		GOLDFARB	
P	OPER	ITER	OPER	ITER	OPER	ITER
1	237	5	218	3	52	0
2	238	5	269	3	134	1
3	296	4	179	2	144	1
4	1184	4	318	2	488	1
5	1480	5	518	3	1046	4
6	614	6	547	4	342	2

TABLA 1

Donde:

p=número de problema
OPER=número de operaciones hasta la solución
ITER=número de iteraciones hasta la solución

Los siguientes problemas que se corrieron, son de programación cuadrática estrictamente convexos y fueron generados

aleatoriamente usando el algoritmo dado por Rosen y Suzuky [15], en el que la solución óptima es conocida de antemano. Este necesita los datos siguientes: un punto x^* que se quiera tener como óptimo, las variables u_i duales cuyo valor determina cuales restricciones serán las activas en la solución y las matrices del término cuadrático de la función objetivo y de las restricciones. Con estos datos, se encuentran los coeficientes del término lineal de la función objetivo y los términos independientes de las restricciones.

En este trabajo los problemas se diseñaron de la siguiente forma.(Goldfarb [9]).

Cada serie de problemas se generó aleatoriamente de acuerdo al número de variables n={9, 27, 81}, al número de restricciones m={n,3n}, al número de restricciones activas k={m/9,m/3}, en la solución y a la condición (según Goldfarb) ={1,0}, 1=buena ó 0=mala, de la matríz Hessiana G del término cuadrático.

La manera en que se generó aleatoriamente cada serie de problemas es la siguiente:

Los elementos no diagonales de la matríz G se generaron dentro del intervalo (-1,1), esto es, $G_{i,j}$ = 2random(r1)-1 siendo r1 la semilla generadora cuyo valor se tomó como r1=.001*rnc. Donde, rnc es una varible que va tomando valores de forma tal que la semilla generadora varíe en cada uno de los problemas de la serie corespondiente. Su valor se da en mas adelante en (4.1).

Los elementos diagonales de G se formaron sumando los valores absolutos de los elementos no diagonales de cada renglón i, obteniendose la suma S_i. Si se quiera que G sea bién condicionada, $G_{i,i}=S_i+$random(r2)+1; en caso contrario, $G_{i,i}=G_{i-1,i-1}+S_i+ S_{i-1}+$ random(r2), con r2=0.004*rnc.

Las variables duales u_i activas en la solución de cada

problema, se generaron de acuerdo al siguiente esquema: si el número de variables n, es 9, $u_i \in (0,30)$, esto es, u_i=random(r3)*30, con r3=0.2*rnc. Si el número de variables n es 27 u 81, $u_i \in (0,30m)$ ó $u_i \in (0,81m)$ respectivamente.

Las componentes x_i^* del vector x^* se generaron en el intervalo (0,5), i.e., x_i^* = random(r4)*5, con r4=0.003*rnc, y los elementos de la matriz A de las restricciones, se generaron de forma tal que cada $A_{ij} \in (0,1)$ esto es, A_{ij}=2*r5-1, con r5=0.008*rnc. Una vez generada A sus columnas fueron normalizadas.

Como en el algoritmo de Goldfarb e Idnani las soluciones pueden ser negativas, como se vió en los ejemplos anteriores, al programar su algoritmo hay que aumentar al conjunto de las m restricciones las cotas $x_i \geq 0$, i=1,...,n. Para el caso del algoritmo de Lemke esto no es necesario pués sus soluciones son siempre no negativas.

A partir de estos datos, se encontraron los coeficientes c_i del término lineal c de la función objetivo y los términos independientes b_i de las restricciones, que de acuerdo a Rosen y Suzuky [15] se calculan como sigue:

$$c = Au - Gx^*$$
$$b = Ax^* - S$$

donde S_i=0 si u_i es activa en la solución, y $S_i \in (0,1)$ si u_i =0.

De esta forma, el problema de programación cuadrática generado y cuya solución x^* es conocida. Es:

$$\text{minimizar } f(x)=c^t x+x^t Gx$$
$$\text{sujeto a } A^t x \geq b$$
$$\& \quad x \geq 0.$$

Se realizaron 8 corridas diferentes de cada serie; así, para n=9, m=9, k=1 y G bien condicionada, se corrieron 8

problemas diferentes.

Para n=9, m=9, k=1 y G mal condicionada se corrieron 8 problemas diferentes
etc..

Veamos un ejemplo:
para n=9, m=9, k=1=m/9 (num. de restricciones activas en la sol) tomamos la semilla de random rnc como: rnc=5*Iprob, donde Iprob=1,...,8. De esta forma, cambia la semilla de random en cada problema generandose 8 problemas diferentes con el mismo número de variables, restricciones y número de restricciones activas en la solución.

En los ejemplos que se corrieron, las cosas quedaron de la siguiente manera:

si	n=9,	m=9,	k=m/9=1	entonces	rnc=5*Iprob
	n=9,	m=9,	k=m/3=3	"	rnc=6*Iprob
	n=9,	m=27,	k=m/3=9	"	rnc=6.1*Iprob
	n=9,	m=27,	k=m/9=3	"	rnc=6.2*Iprob
	n=27,	m=27,	k=m/9=3	"	rnc=4.1*Iprob
	n=27,	m=27,	k=m/3=9	"	rnc=4.3*Iprob
	n=27,	m=81	k=m/3=27	"	rnc=6*Iprob
	n=27,	m=81	k=m/9=9	"	rnc=7*Iprob
	n=81,	m=81	k=m/9=9	"	rnc=8*Iprob
	n=81,	m=81	k=m/3=27	"	rnc=9*Iprob
	n=81,	m=243	k=m/9=27	"	rnc=10*Iprob
	n=81,	m=243	k=m/3=81	"	rnc=11*Iprob.

(4.1)

Una vez que se llevaron a cabo las 8 corridas de cada serie se promediaron tanto el número de iteraciones como el número de operaciones (sumas, restas, multiplicaciones, divisiones y 10*raíz cuadrada) y se hizo la comparación con cada uno de los algoritmos, obteniendose la tabla 2

					LEMKE		SARGENT		GOLDFARB	
s	n	m	k	IBC	OPER	ITER	OPER	ITER	OPER	ITER
1	9	9	1	1	8	11	4	9	2	1
1	9	9	1	0	8	11	4	9	2	1
2	9	9	3	1	10	14	7	12	4	3
2	9	9	3	0	10	13	7	11	4	3
3	9	27	3	1	39	14	11	12	5	3
3	9	27	3	0	36	13	10	11	5	3
4	9	27	9	1	72	25	37	24	14	12
4	9	27	9	0	57	20	27	18	12	10
5	27	27	3	1	200	33	102	31	44	3
5	27	27	3	0	192	31	98	29	49	4
6	27	27	9	1	305	50	214	48	101	11
6	27	27	9	0	246	40	169	38	101	11
7	27	81	9	1	1647	69	577	67	159	15
7	27	81	9	0	1186	50	366	48	151	13
8	27	81	27	1	3320	139	1957	137	546	74
8	27	81	27	0	1910	80	961	78	358	44
9	81	81	9	1	5877	110	2925	108	1205	9
9	81	81	9	0	5048	95	2564	93	1376	12
10	81	81	27	1	8721	163	5972	161	2491	29
10	81	81	27	0	6183	116	4082	114	2639	31
11	81	243	27	1	62276	294	17989	292	3431	32
11	81	243	27	0	29658	140	8616	138	3899	39
12	81	243	81	1	139595	659	no conv.	----	17866	324
12	81	243	81	0	53060	251	26729	249	9602	135

TABLA 2

DONDE:

s=serie de problemas

n=número de variables

m=número de restricciones

k=número de restricciones activas en la solución

IBC=(1 , 0) matríz del término cuadrático bién ó mal condicionada.

ITER=número de iteraciones (promedio)

OPER=número de operaciones en miles (promedio).

(recordar que cada uno de estos es el promedio del número de operaciones e iteraciones de las 8 corridas de cada serie).

CAPITULO V

CONCLUSIONES

Una vez que se vaciaron en las dos tablas anteriores el número de operaciones e iteraciones hasta la solución para la tabla 1 y el promedio para la tabla 2, se cotejó el número de operaciones obtenido con lo que se reporta en las publicaciones. Para esto se observó lo siguiente:

Sargent dá el número de operaciones por iteración en términos de m y n, donde m es el orden de la matríz M_{11}(2.13) actual y n el número de elementos básicos. De esta forma, al agregar un renglón y una columna a M_{11} el número de operaciones reportado es, $2[n(m+2)+2m^2+7m+3]$ (sumas, restas, mult y div). Y cada vez, se quita un renglón y una columna de M_{11}, el número de operaciones reportado es, $2[n(m+2)+4(m^2+29m+21)/3]$; en los dos casos, lo reportado coincide con lo obtenido. Observando estas 2 fórmulas, se ve que si m es pequeña, el costo por iteración es bajo y va aumentando conforme aumenta su valor.

En el caso de Goldfarb e Idnani se reporta el número de operaciones hasta la solución, (multiplicaciones, divisiones, 10*raíz cuadrada) el número de sumas y restas aunque no se reporta, es del mismo orden que el de multiplicaciones y divisiones; asi que tomando como base la mitad de lo obtenido en las tablas 1 y 2, (pues en este trabajo se toman en cuenta todas

las operaciones) se observó que el promedio de operaciones obtenido es en todos los casos, excepto para los problemas de la corrida 9, menor . Esto a pesar de que Goldfarb no toma en cuenta el número de operaciones que se realizan al evaluar las restricciones no activas para encontrar la restricción violada que se convertirá en activa. Si no se toman en cuenta esas operaciones el promedio de operaciones obtenido, es aun menor

Hay que hacer notar que la implementación que hace Sargent al algoritmo de Lemke, no solo no mejoró la estabilidad numerica sino que en algunos casos se vió muy inestable P. E. en el problema 8 de la serie en que n=27, m=81, k=27, e IBC=1, la solución que se obtiene con el algoritmo de Sargent es muy mala en cambio el algoritmo de Lemke si da la solución correcta lo mismo ocurre para tres problemas de la serie en que n=81, m=243, k=27 e IBC=1. Para la serie de problemas en que n=81 m=243, k=81 e IBC=1 en ninguno de ellos, como se ve en la tabla 2, se llega a la solución independientemente del número de iteraciones. Parece ser que al aumentar el número de iteraciones el problema se desestabiliza cada vez más hasta que lo que se está procesando está completamente alejado de solución alguna y por tanto, para esos problemas, el algoritmo no converge. Lo que si se puede decir es que el número de operaciones, en el caso de convergencia, si disminuye en comparación con el algoritmo de Lemke.

Es muy importante (en el caso del algoritmo de Goldfarb e Idnani) señalar que al hacer la selección de la restricción violada que entrará a formar parte del conjunto activo, se hará de

forma tal que se tome la que tenga el valor más negativo, ya que si esto no se hace, el número de iteraciones y por tanto el de operaciones se ve enormemente incrementado. Otra cosa importante de apuntar, es que el cálculo de $\bar{Q}$ para la actualización de la matríz J en (3.29) debe llevarse a cabo con matrices de Givens pués, si se hace con el método de Householder, aumenta muchísimo el número de operaciones, como se muestra en la tabla 3, además de que la estabilidad numérica se ve enormemente deteriorada, por lo que el número de iteraciones se incrementa y en algunos casos no se obtiene la solución.

En la tabla 3, se dan los promedios de operaciones e iteraciones de los mismos problemas de la tabla 2 aplicando solo el algoritmo de Goldfarb y tomando en cuenta lo siguiente:

En la parte A se aplicó el algoritmo a los problemas, calculando la matriz $\bar{Q}$ mediante el método de Householder, y luego multiplicando esta por J_2 para la actualización de J cuando se agrega una restricción al conjunto activo. Considerando que la restricción que se elige para entrar al conjunto activo, es la primera restricción violada que se encuentra. Y además, que los x_i no se toman como restricciones pues el punto solución es positivo.

En la parte B, se hace lo mismo que en la parte A pero considerando que la restricción que se elige para entrar al conjunto activo, es la más violada.

En la parte C, la actualización de J se hace siempre con matrices de Givens y la selección de la restricción a entrar al

conjunto activo es la más violada, al igual que en la tabla 2.

					A		B		C	
s	n	m	k	IBC	OPER	ITER	OPER	ITER	OPER	ITER
1	9	9	1	1	3	2	3	1	2	1
1	9	9	1	0	3	2	3	1	2	1
2	9	9	3	1	7	6	5	3	4	3
2	9	9	3	0	5	4	5	3	4	3
3	9	27	3	1	18	19	7	3	5	3
3	9	27	3	0	14	12	7	3	5	3
4	9	27	9	1	26	32	16	11	13	11
4	9	27	9	0	20	19	14	9	12	9
5	27	27	3	1	429	25	95	3	44	3
5	27	27	3	0	305	14	95	3	44	3
6	27	27	9	1	641	40	298	10	96	10
6	27	27	9	0	565	29	304	11	98	11
7	27	81	9	1	1875	298	360	13	148	13
7	27	81	9	0	1306	122	366	12	142	12
8	27	81	27	1	1885	290	828	68	520	68
8	27	81	27	0	1297	163	661	40	341	40
9	81	81	9	1	49179	144	8224	9	1205	9
9	81	81	9	0	34652	71	8467	10	1230	10
10	81	81	27	1	54178	187	21453	28	2456	28
10	81	81	27	0	46853	107	21331	27	2434	27

TABLA 3

Donde:

s=serie de problemas

n=número de variables

m=número de restricciones

k=número de restricciones activas en la solución

IBC=1 si la matríz del término cuadrático está bién condicionada ó IBC=0 si está mal condicionada

ITER=número de iteraciones (promedio)

OPER=número de operaciones en miles (promedio)

Un ejemplo más y que solo se corrió para el algoritmo de

G-I, pués sus solución es negativa, es el siguiente (Powell [14])

$$\text{minimizar } f(x)=x_1 + x_2 + 1/2(10^{-10}x_1^2 + 10^{-20}x_2^2)$$

sujato a las restricciones

$$a_{1k}x_1 + a_{2k}x_2 \geq -1 , \quad k=1,2,\ldots,20$$

donde para cada k los coeficientes tienen los valores

$$a_{1k}=\cos(0.68 + 0.01k)$$
$$a_{2k}=\text{sen}(0.68 + 0.01k)$$

cuya solución es;

$$x_1^*=-\cos(0.785)/\cos(0.005)$$
$$x_2^*=-\text{sen}(0.785)/\cos(0.005)$$

De acuerdo a la publicación de Powell [14], quién hace una implementación al algoritmo de Goldfarb e Idnani, (la que no se llevó acabo en este trabajo) las restricciones activas en la solución son la 10 y la 11 y los multiplicadores de Lagrange correspondientes son 0.650800943 y 0.763430178. Losresultados obtenidos para este mismo problema con el algoritmo de G-I de este trabajo son: 0.650806518 y 0.763424609. Si el ejemplo de Powell es corrido sin seleccionar la restrición más violada, el número de iteraciones para llegar a la solución es 20; en cambio si se selecciona la restricción mas violada el número de restricciones para llegar a la solución, se reduce a 6. De aquí la importancia de seleccionar la restricción más violada.

Con respecto a la implementación de los algoritmos, el más difícil fué el de Sargent, sobre todo cuando hay que quitar un renglón y una columnna de la matriz M_{11}. En el caso de la implememtación del algoritmo de G-I el problema fué mínimo comparado con el de la implementación del de Sargent.

Es claro, como se observa en las tablas 1) y 2) del capítulo IV), que el más eficiente de los dos métodos, es el algoritmo de Goldfarb e Idnani. Sin embargo, si no se toma la restricción más violada y la actualización de J no se realiza con matrices de Givens, el algoritmo de G-I es bastante menos eficiente y más caro (como se muestra en la tabla 3) que la implementación que hace Sargent del algoritmo de Lemke.

B I B L I O G R A F I A

[1] Avriel Mordecai "Nonlinear Programming Analysis and Methods" , Prentice -Hall (1976).

[2] J.W. Daniel, W.B. Graggs, L. Kaufman and G.W. Stewart. "Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorizations", Mathematics of Computation 30 (1976) 772-795.

[3] R. Fletcher, "A general quadratic progamming algorithm", Journalof institute of Mathematics and its applications (1971) 76-91.

[4] P.E. Gill, G.H. Golub, W. Murray and M.A. Saunders, "Methods for modifying matrix factorizations", Mathematic of Computation 28(1974) 505-535.

[5] P.E. Gill, W. Murray, M.A. Saunders, "Methods for Computing and Modifying the LDV Factors of a Matrix", Mathematics of Computation, 29(132) 1051-1077.

[6] P.E. Gill and W. Murray, "Numerical stable methods for quadratic Progamming", Mathematical Programming 14(1978) 349-372.

[7] D. Goldfarb, "Extension of Newton's method and simplex methods for solving quadratic programs", in: F.A. Lootsma, ed., Numerical methos for non linear optimization (Academic Press, London, 1972) 239-254.

[8] D. Goldfarb, "Matrix factorizations in optimization of non linear funtions subject to linear constrains", Mathematical Programming 10(1975)1-31

[9] D. Goldfarb and A. Idnani, "A numerical stable dual method for solving stricty convex quadratic programs", Mathematical Programming 27(1983) 1-33.

[10] D. Goldfarb and A. Idnani, "Dual and primal-dual methods for solving strictlyconvex quadratic programs" in: J.P. Henart, ed., Numerical Analysis, Proceedings Cocoyoc Mexico 1981, Lecture Notes in Mathematics 909(Spring-Verlag, Berlin, 1982) 226-239.

[11] S. Gomez, "Two complementary pivot methods for solving the quadratic programming problem", Thesis Master of Science Degree (Imperial College of Science and Techonology, University of London 1975).

[12] D.G. Luemberger, "Introduction to linear and non linear programming", Addisson Wesley 1973.

[13] L. Peñafiel Millán "Programación Lineal", Trillas 1982

[14] M.J.D. Powell "ZQPCVX A Fortran subroutine for convex for convex cuadratic programming", Departament of AppliedMathematics

and theoretical Physics, University of Cambridge, DAMP/1983/NA17

[15] J.B. Rosen And S. Suzuki, "Construction of nonlinear progamming test Problems", Communications of the ACM (1965) 113

[16] R.W.H. Sargent, "An efficient implementation of the Lemke algorithm and its extension to deal with upper and lower bounds", Mathematical Programming Study 7(1978) 36-54.

[17] G. W. Stewart, "Introduction to Matrix Computation".New York Academic 1973

[18] G. Strang, "Algebra lineal y sus aplicaciones", Fondo Educativo Interamericano 1985.

[19] C. Van de Pane, "Methods for Linear and Quadratic Programming", New York Elsevier 1975.