Programación Lineal
Proyecto Computacional
El Método Simplex

1 Introducción

El problema a resolver es

$$
\begin{array}{ll}
\text{Maximizar} & \sum_{j=1}^{n} c_i x_i \\
\text{Sujeto a} & \\
& \sum_{j=1}^{n} a_{i,j} x_j \leq b_i, \;\; i = 1,\ 2,\ \ldots,\ m \\
& \\
& x_j \geq 0,\ j = 1,\ 2,\ \ldots,\ n,
\end{array} \tag{1}
$$

Supongamos que obtenemos un diccionario a partir de (1), usando variables de holgura u otra técnica tal que

$$
\begin{array}{rcl}
\zeta & = & \zeta^* + \sum_{j\in\mathcal{N}} \hat{c}_i x_i \\
\hline
x_i & = & \hat{b}_i - \sum_{j\in\mathcal{N}} \hat{a}_{i,j} x_j,\ i \in \mathcal{B} \\
\hline
0 & \leq & x_i,\ \text{para toda}\ i.
\end{array} \tag{2}
$$

donde :

$$\mathcal{B} \cup \mathcal{N} = \{1,\ 2,\ \ldots,\ m,\ m+1,\ \ldots,\ m+n\},\ \ \mathcal{B} \cap \mathcal{N} = \emptyset,\ |\mathcal{B}| = m,$$

y $\hat{c}_j,\ \hat{b}_i,\ \hat{a}_{i,j}$ corresponden a la modificación de los valores $c_i,\ b_i,\ a_{i,j}$ y $\hat{b}_i \geq 0$ para $i \in \mathcal{B}$.

El proceso para obtener un diccionario con estas características se llama **Fase I del Simplex**.

Si no es posible determinar un diccionario con las características de (2) el problema (1) tiene conjunto factible vacío.

El conjunto $\mathcal{B}$ se llama el conjunto de variables básicas para el diccionario (2) y $\mathcal{N}$ el conjunto de variables no básicas.

Como $\hat{b}_i \geq 0$ para todo $i \in \mathcal{B}$ el vector dado por

$$x_i = \hat{b}_i,\ \forall\ i \in \mathcal{B},\ \ x_j = 0,\ \forall\ j \in \mathcal{N}$$

y omitiendo los valores que corresponden a las variables de holgura, se tiene un punto factible para (1), que es un vértice del conjunto factible en (1).

2 El Método Simplex/ Fase II

S0. Se tiene el diccionario

$$\begin{array}{lcl} \zeta & = & \zeta^* + \sum_{j\in\mathcal{N}} \hat{c}_i x_i \\ \hline x_i & = & \hat{b}_i - \sum_{j\in\mathcal{N}} \hat{a}_{i,j} x_j, \; i \in \mathcal{B} \\ \hline 0 & \leq & x_i, \text{ para toda } \; i. \end{array}$$

donde :

$$\mathcal{B} \cup \mathcal{N} = \{1,\ 2,\ \ldots,\ m,\ m+1,\ \ldots,\ m+n\},\ \ \mathcal{B} \cap \mathcal{N} = \emptyset,\ |\mathcal{B}| = m,$$

y $\hat{c}_j,\ \hat{b}_i,\ \hat{a}_{i,j}$ corresponden a la modificación de los valores $c_i,\ b_i,\ a_{i,j}$ y $\hat{b}_i \geq 0$ para $i \in \mathcal{B}$.

S1. Sea $k \in \mathcal{N}$ es el primer índice tal que $\hat{c}_k > 0$. (Regla de Bland).

El índice k será la **variable de entrada**.

Si no existe tal índice k el punto $x_i = 0$ para $i \in \mathcal{N}$ y $x_i = \hat{b}_i$ para $i \in \mathcal{B}$ es solución óptima del problema. La función objetivo no puede crecer más. **SALIDA ÓPTIMA**.

S2. Sea $l \in \mathcal{B}$ el primer índice (Regla de Bland) tal que

$$\left(\frac{\hat{b}_l}{\hat{a}_{l,k}}\right) = \text{Min} \left\{ \left(\frac{\hat{b}_i}{\hat{a}_{i,k}}\right),\ |\ \hat{a}_{i,k} > 0,\ i \in \mathcal{B} \right\}$$

El índice l será la **variable de salida**.

Si no existe el índice l entonces el problema es no acotado superiormente. **SALIDA PROBLEMA NO ACOTADO**.

S3. Despejar la variable x_k de la ecuación

$$x_l = \hat{b}_l - \sum_{j \in \mathcal{N}} \hat{a}_{l,j} x_j$$

$$x_l = \hat{b}_l - \hat{a}_{lk} x_k - \sum_{j \in \mathcal{N} - \{k\}} \hat{a}_{l,j} x_j$$

es decir

$$x_k = \left(\frac{\hat{b}_l}{\hat{a}_{lk}} \right) - \left(\frac{1}{\hat{a}_{lk}} \right) x_l - \sum_{j \in \mathcal{N} - \{k\}} \left(\frac{\hat{a}_{l,j}}{\hat{a}_{lk}} \right) x_j \tag{3}$$

y sustituir en cada ecuación del diccionario en **S0** para obtener el nuevo diccionario.

S4. Actualizar la función objetivo

$$\zeta = \zeta^* + \sum_{j \in \mathcal{N}} \hat{c}_j x_j$$

$$\zeta = \zeta^* + \hat{c}_k x_k + \sum_{j \in \mathcal{N} - \{k\}} \hat{c}_j x_j,$$

se sustituye la ecuación (3) en esta última expresión y se obtiene:

$$\zeta = \left[\zeta^* + \hat{c}_k \left(\frac{\hat{b}_l}{\hat{a}_{lk}} \right) \right] - \left(\frac{\hat{c}_k}{\hat{a}_{lk}} \right) x_l + \sum_{j \in \mathcal{N} - \{k\}} \left[\hat{c}_j - \hat{c}_k \left(\frac{\hat{a}_{lj}}{\hat{a}_{lk}} \right) \right] x_j.$$

Hacer

$$\zeta^* \leftarrow \left[\zeta^* + \hat{c}_k \left(\frac{\hat{b}_l}{\hat{a}_{lk}} \right) \right]$$

$$\hat{c}_j \leftarrow \left[\hat{c}_j - \hat{c}_k \left(\frac{\hat{a}_{lj}}{\hat{a}_{lk}} \right) \right], \;\; j \in \mathcal{N} - \{k\},$$

y

$$\hat{c}_l \leftarrow - \left(\frac{\hat{c}_k}{\hat{a}_{lk}} \right).$$

La nueva función objetivo es

$$\zeta = \zeta^* + \hat{c}_l x_l + \sum_{j \in \mathcal{N} - \{k\}} \hat{c}_j x_j$$

S5. Actualizar las restricciones

Para $i \in \mathcal{B},\ i \neq l$ se tiene que

$$x_i = \hat{b}_i - \sum_{j \in \mathcal{N}} \hat{a}_{ij} x_j$$

es decir

$$x_i = \hat{b}_i - \hat{a}_{ik} x_k - \sum_{j \in \mathcal{N} - \{k\}} \hat{a}_{ij} x_j,$$

de nuevo sustituimos x_k por la expresión en (3) y obtenemos

$$x_i = \left[\hat{b}_i - \left(\frac{\hat{a}_{ik}}{\hat{a}_{lk}}\right) \hat{b}_l\right] - \left(\frac{\hat{a}_{ik}}{\hat{a}_{lk}}\right) x_k - \sum_{j \in \mathcal{N} - \{k\}} \left[\hat{a}_{ij} - \left(\frac{\hat{a}_{lj}}{\hat{a}_{lk}}\right) \hat{a}_{ik}\right] x_j$$

Hacer para $i \in \mathcal{B},\ i \neq l$

$$\hat{b}_i \leftarrow \left[\hat{b}_i - \left(\frac{\hat{a}_{ik}}{\hat{a}_{lk}}\right) \hat{b}_l\right]$$

$$\hat{a}_{ij} \leftarrow \left[\hat{a}_{ij} - \left(\frac{\hat{a}_{lj}}{\hat{a}_{lk}}\right) \hat{a}_{ik}\right],\ \ \forall\ \ j \in \mathcal{N} - \{k\}$$

$$\hat{a}_{il} \leftarrow \left(\frac{\hat{a}_{ik}}{\hat{a}_{lk}}\right).$$

Hacer para k (ver ecuación (3))

$$\hat{b}_k \leftarrow \left(\frac{\hat{b}_l}{\hat{a}_{lk}}\right)$$

$$\hat{a}_{kj} \leftarrow \left(\frac{\hat{a}_{l,j}}{\hat{a}_{lk}}\right),\ \ \forall\ j \in \mathcal{N} - \{k\}$$

$$\hat{a}_{kl} \leftarrow \left(\frac{1}{\hat{a}_{lk}}\right).$$

S6. Actualizar los conjuntos de variables básicas y no básicas

$$\mathcal{B} \leftarrow (\mathcal{B} - \{l\}) \cup \{k\},\ \mathcal{N} \leftarrow (\mathcal{N} - \{k\}) \cup \{l\}$$

Ir a S0.

3 Proyecto

3.1 Método Simplex

Se debe programar en MATLAB la Fase II del Simplex. la sección anterior contiene el álgebra y los detalles de cada uno de los pasos. Para programarlo en la computadora no es necesario seguir al pie de la letra cada uno de las partes algebraicas.

Es importante el manejo de los conjuntos $\mathcal{B}$ y $\mathcal{N}$ en cada iteración.

function $[x,\ fx,\ ban,\ iter] = SIMPLEXFASEII(c,\ A, b,\ \mathcal{B})$
% Versión del Simplex en la fase II.
% $\mathcal{B}$ es el conjunto inicial de variables básicas.
% Salida
% x valor óptimo del problema.
% fx valos de la función objetivo en x.
% ban indica los siguientes casos: $band = -1$ conjunto factible vacío.
% $ban == 0$ función objetivo no acotada superiormente.
% $ban = 01$ se encontró solución óptima.
% $iter$ es el número de iteraciones (cambios de diccionarios) que hace el método.
% %
% A partir de $calB$ usted puede construir el diccionario inicial.

Un código eficiente no contiene más de 35 líneas de programación.

3.2 Problemas Aleatorios

Este programa en MATLAB genera problemas aleatorios como en (1).

% Genera las dimensiones del problema

$m = round(10 * exp(log(100) * rand()));$
$n = round(10 * exp(log(100) * rand()));$

% Acomodamos las dimensiones del problema
if $(m > n)$
    $m1 = n;$
    $n = m;$

```
    m = m1;
end

% Genera matriz y vectores

sigma = 10;
A = round(sigma * (randn(m, n)));
b = round(sigma * abs(randn(m, n))); % b > 0
c = round(sigma * (randn(n,1)));

% Se depuran las entradas de A
tau = 0.85;
for k = 1 : m
    for j = 1 : n
        if abs(A(k,j)) < tau
            A(k,j) = 0.0
        end
    end
end
```

Cada iteración del Simplex (cambio de diccionario) se considera un pivote.

Una parte del proyecto es correr mil veces el método Simplex y en cada caso guardar: (a) los valores de n y m (b) el número de pivotes en el problema
(c) Una variable lógica que indique si se encontró solución óptima o si es no acotado superiormente.

Graficar en el eje X la dimensión $n+m$ y en el eje Y el número de pivotes en cada caso que se resuelva. Si se encontró una solución óptima poner una marca + en color azul, si el problema fue no acotado superiormente poner una marca x en color verde. En la graficación use para ambos ejes la escala *log*.

Para más detalles ver: *Linear Programming, Foundations and Extensions, Four Edition*, International Series in Operational Research Management Science, Springer New York, 2014, pags 44-50.

3.3 Ejemplo de Klee-Minty

El Simplex realiza $(2^n - 1)$ iteraciones para resolver el siguiente problema debido a Klee y Minty.

$$
\begin{array}{ll}
\text{Maximizar} & \sum_{i=1}^{n} 10^{n-i} x_i \\
\text{Sujeto a} & \\
 & 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \leq 100^{i-1}, \;\; i = 1, \ldots, n \\
 & \\
 & x_i \geq 0, \; i = 1, \ldots, n
\end{array}
$$

Ver, Klee-Minty, *How good is the simplex algorithm?* Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin), pp. 159â175. Academic Press, New York, 1972.

Resolver los casos $n = 3, 4, 5, 6.$ y reportar el número de iteraciones en cada caso junto con el tiempo de máquina.