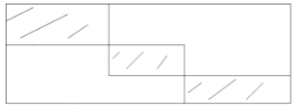
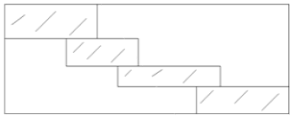
TÉCNICAS DE DESCOMPOSICIÓN EN PROGRAMACIÓN MATEMÁTICA

Dr. Víctor M. Albornoz
Departamento de Industrias.
Campus Santiago Vitacura, Chile.
Universidad Técnica Federico Santa María

Instituto de Computación, Facultad de Ingeniería, UdelaR.
Montevideo, lunes 20 al viernes 24 de Mayo de 2024

CONTENIDOS





5.- Método de Dantzig & Wolfe.

Para detallar este método consideramos primeramente el siguiente modelo de P. Lineal:

$$\begin{aligned} \text{Min} \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.a.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathbf{X} \end{aligned} \qquad (5.1)$$

donde **A** es una matriz cualquiera que define las restricciones complicadas del problema y **X** un poliedro *acotado* que reúne las restantes restricciones lineales del problema.

Denotando por $\mathbf{x^1},\ldots,\mathbf{x^t}$ los vértices de $\mathbf{X}$, para cada $\mathbf{x} \in \mathbf{X}$ se cumple:

$$\mathbf{x} = \lambda_1 \mathbf{x^1} + \lambda_2 \mathbf{x^2} + \cdots + \lambda_t \mathbf{x^t}$$

con escalares $\lambda_1 \geq 0, \lambda_2 \geq 0, \ldots, \lambda_t \geq 0$ tales que:

$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1$$

De este modo el problema (5.1) es equivalente a resolver el siguiente *Problema Maestro*:

$$\begin{aligned} \text{Min } & (\mathbf{c^T x^1})\lambda_1 + (\mathbf{c^T x^2})\lambda_2 + \cdots + (\mathbf{c^T x^t})\lambda_t && (5.2) \\ \text{s.a. } & (\mathbf{A x^1})\lambda_1 + (\mathbf{A x^2})\lambda_2 + \cdots + (\mathbf{A x^t})\lambda_t = \mathbf{b} \\ & \lambda_1 + \lambda_2 + \cdots + \lambda_t = 1 \\ & \lambda_1 \geq 0, \lambda_2 \geq 0, \ldots, \lambda_t \geq 0. \end{aligned}$$

Sea B una matriz de base asociada a una solución básica factible del *Problema Maestro*.

Sean ω y α las respectivas variables duales asociadas a las restricciones en (5.2), entonces:

$$[\omega^T \ \alpha] = c_B{}^T B^{-1}$$ c_B: vector de costos básicos

Las condiciones de optimalidad que deben verificarse para cada variable no-básica λ_j son:

$$(\mathbf{c}^T\mathbf{x}^j) - [\omega^T \ \alpha]\begin{bmatrix}\mathbf{A}\mathbf{x}^j \\ 1\end{bmatrix} = (\mathbf{c}^T\mathbf{x}^j) - \omega^T(\mathbf{A}\mathbf{x}^j) - \alpha = (\mathbf{c}^T - \omega^T A)\mathbf{x}^j - \alpha \geq 0$$

Para verificar lo anterior no es necesario conocer todos los vértices del poliedro X, pues basta con resolver el siguiente *Subproblema*:

$$\text{Min } (\mathbf{c}^T - \boldsymbol{\omega}^T A)\mathbf{x} - \alpha \qquad (5.3)$$
$$\text{s.a. } \mathbf{x} \in X$$

En efecto, si $\mathbf{x}^k$ es el vértice donde se alcanza la solución óptima del *Subproblema* y se cumple:

$$(\mathbf{c}^T - \boldsymbol{\omega}^T A)\mathbf{x}^k - \alpha \geq 0,$$

entonces todos los otros vértices le verifican y hemos alcanzado la solución óptima al problema.

En caso contrario, tenemos una nueva columna para el Maestro Reducido asociada a la variable no-básica λ_k que entra a la base del nuevo problema.

En el contexto del Simplex Revisado, se calcula el vector $\mathbf{y_k}$ para determinar la variable que deja la base. Este determina la columna de la variable entrante λ_k como combinación de las actuales columnas en B, resolviendo:

$$B\mathbf{y_K} = \begin{bmatrix} \mathbf{A}\mathbf{x}^k \\ 1 \end{bmatrix}$$

y se sigue con una nueva iteración del método.

Algoritmo.

0. Obtener una solución factible inicial del *P. Maestro*, que define un *Maestro Reducido* con las columnas dadas.
1. Calcular coeficientes de la función objetivo del *Subproblema,* que define el costo reducido de la solución.
2. Resolver el *Subproblema*. En caso de tener un valor óptimo positivo STOP. En caso contrario, obtener nueva variable básica para el *Maestro Reducido* (nueva columna a ser incorporada en el Maestro Reducido).
3. Resolver el *P. Maestro Reducido.* En el contexto del M.Simplex esto pasa por determinar qué variable básica deja la base y obtener el vector de precios duales. Ir a 1.

Ejemplo. Resolveremos el siguiente problema de Programación Lineal mediante el método de descomposición de Dantzig & Wolfe:

$$\begin{aligned} \text{Max} \quad & 4x_1 + 2x_2 + 6x_3 \\ \text{s.a.} \quad & 3x_1 + x_2 + 4x_3 \leq 17 \\ & 1 \leq x_1 \leq 3 \\ & 1 \leq x_2 \leq 3 \\ & 1 \leq x_3 \leq 3. \end{aligned}$$

El problema puede expresarse como:

$$\begin{aligned} \text{Min} \quad & -4x_1 - 2x_2 - 6x_3 \\ \text{s.a.} \quad & 3x_1 + x_2 + 4x_3 \leq 17 \\ & x \in X \end{aligned}$$

donde $X=\{ (x_1,x_2,x_3)\in\Re^3 / \ 1\leq x_1\leq 3, \ 1\leq x_2\leq 3, \ 1\leq x_3\leq 3\}$ es un poliedro acotado.

Así, denotando por $\mathbf{x^1}, \mathbf{x^2},\ldots, \mathbf{x^t}$ los vértices de X, se tiene la siguiente representación para cada $x\in X$:

$$x = \lambda_1\mathbf{x^1} + \lambda_2\mathbf{x^2} + \cdots + \lambda_t\mathbf{x^t}$$

con escalares no-negativos tales que: $\lambda_1+\lambda_2+\cdots+\lambda_t=1$.

El problema puede expresarse como:

$$\text{Min} \quad \mathbf{c}^T(\lambda_1\mathbf{x}^1 + \lambda_2\mathbf{x}^2 + \cdots + \lambda_t\mathbf{x}^t)$$

$$\text{s.a.} \quad \mathbf{A}(\lambda_1\mathbf{x}^1 + \lambda_2\mathbf{x}^2 + \cdots + \lambda_t\mathbf{x}^t) \leq 17$$

$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1$$

$$\lambda_1 \geq 0,\ \lambda_2 \geq 0,\ \ldots,\ \lambda_t \geq 0$$

Con $\mathbf{c}^T = [-4\ -2\ -6]$ y $\mathbf{A} = [3\ \ 1\ \ 4]$ equivale a resolver:

PROBLEMA MAESTRO (PM):

$$\text{Min } (\mathbf{c}^T\mathbf{x}^1)\lambda_1 + (\mathbf{c}^T\mathbf{x}^2)\lambda_2 + \cdots + (\mathbf{c}^T\mathbf{x}^t)\lambda_t$$

$$\text{s.a. } (\mathbf{A}\mathbf{x}^1)\lambda_1 + (\mathbf{A}\mathbf{x}^2)\lambda_2 + \cdots + (\mathbf{A}\mathbf{x}^t)\lambda_t + s_1 = 17 \quad (\omega)$$

$$\lambda_1 + \lambda_2 + \cdots + \lambda_t = 1 \quad (\alpha)$$

$$\lambda_1 \geq 0,\ \lambda_2 \geq 0,\ \ldots,\ \lambda_t \geq 0,\ s_1 \geq 0.$$

Siendo B la matriz de base asociada a una solución básica factible de (PM), entonces:

$[\omega \;\; \alpha] = c_B^T B^{-1}$ c_B: vector de costos básicos

Las condiciones de optimalidad que deben verificarse para cada variable no-básica λ_j son:

$$(\mathbf{c}^T\mathbf{x}^j) - [\omega \;\; \alpha]\begin{bmatrix}\mathbf{A}\mathbf{x}^j \\ [\;1\;]\end{bmatrix} = (\mathbf{c}^T\mathbf{x}^j) - \omega(\mathbf{A}\mathbf{x}^j) - \alpha = (\mathbf{c}^T - \omega\mathbf{A})\mathbf{x}^j - \alpha \geq 0$$

Lo que se verifica resolviendo el SUBPROBLEMA:

(SP) Min $(\mathbf{c}^T - \omega\mathbf{A})\mathbf{x} - \alpha$

s.a. $x \in X$

En el caso del ejemplo:

(SP) $\text{Min } (-4-3\omega)x_1 + (-2-\omega)x_2 + (-6-4\omega)x_3 - \alpha$

$\text{s.a.} \quad 1 \le x_1 \le 3, \quad 1 \le x_2 \le 3, \quad 1 \le x_3 \le 3.$

Notar que el poliedro de las restricciones posee 8 vértices, pero estos se irán generando. Para tener una base inicial factible de PM se necesitan 2 variables básicas pues hay dos restricciones. Nos daremos la variable de holgura (s_1) y λ_1 asociada a un vértice inicial, por ejemplo $\mathbf{x^1}=(1,1,3)^T$.

Primera iteración.

Variables básicas s_1 y λ_1 ($\mathbf{x^1}=(1,1,3)^T$)

$A\mathbf{x^1} = [\,3 \;\; 1 \;\; 4]\mathbf{x^1} = 16 \quad \mathbf{c^T x^1} = [-4 \;\; -2 \;\; -6]\mathbf{x^1} = -24$

$$B=\begin{bmatrix} 1 & 16 \\ 0 & 1 \end{bmatrix} \qquad Bx_B=\begin{bmatrix} 17 \\ 1 \end{bmatrix} \qquad s_1=1 \;\; \lambda_1 = 1 \qquad B^T\begin{bmatrix}\omega \\ \alpha\end{bmatrix} = \begin{bmatrix} 0 \\ -24 \end{bmatrix}$$

$\omega=0 \quad \alpha=-24$

Como $\mathbf{c^T} - \omega\mathbf{A} = [-4,-2,-6]$ el Subproblema corresponde a:

(SP) Min $-4x_1 - 2x_2 - 6x_3 + 24$

s.a. $1 \le x_1 \le 3, \;\; 1 \le x_2 \le 3, \;\; 1 \le x_3 \le 3.$

Su solución es $\mathbf{x^2}=(3,3,3)^T$

Como v(SP)= –12 se incorpora a la base λ_2.

Con $\mathbf{x^2}=(3,3,3)^T$ calculamos

$A\mathbf{x^2} = [\ 3\ \ 1\ \ 4]\mathbf{x^2} = 24$

Se resuelve $B\mathbf{y_2} = \begin{bmatrix} 24 \\ 1 \end{bmatrix}$ con $B = \begin{bmatrix} 1 & 16 \\ 0 & 1 \end{bmatrix}$

De donde $y_2 = \begin{bmatrix} 8 \\ 1 \end{bmatrix}$

Ello determina la variable que deja la base al calcular

$\lambda_2 = \min\{ 1/8 , 1/1 \} = 1/8$ de donde deja la base s_1.

Segunda iteración.

Variables básicas λ_1 ($\mathbf{x^1}=(1,1,3)^T$) y λ_2 ($\mathbf{x^2}=(3,3,3)^T$)

$A\mathbf{x^2} = 24$ $\quad$ $\mathbf{c^T x^2} = [-4\ -2\ -6]\mathbf{x^2} = -36$

$$B=\begin{bmatrix} 16 & 24 \\ 1 & 1 \end{bmatrix} \quad Bx_B=\begin{bmatrix} 17 \\ 1 \end{bmatrix} \quad \lambda_1=7/8 \quad \lambda_2=1/8 \quad B^T\begin{bmatrix} \omega \\ \alpha \end{bmatrix} = \begin{bmatrix} -24 \\ -36 \end{bmatrix}$$

$\omega = -3/2 \quad \alpha=0$

Como $\mathbf{c^T} - \omega\mathbf{A} = [-4\ \ -2\ -6] + (3/2)[3\ \ 1\ \ 4] = [1/2\ -1/2\ \ 0]$

(SP) Min $x_1/2 - x_2/2$

s.a. $1 \le x_1 \le 3,\ \ 1 \le x_2 \le 3,\ \ 1 \le x_3 \le 3.$

Una solución óptima $x^3=(1,3,3)^T$

Como $v(SP)= -1$ se incorpora a la base λ_3.

Con $\mathbf{x^3}=(1,3,3)^T$ calculamos

$A\mathbf{x^3} = [\ 3\ \ 1\ \ 4]\mathbf{x^3} = 18$

Se resuelve $By_3 = \begin{bmatrix} 18 \\ 1 \end{bmatrix}$ con $B = \begin{bmatrix} 16 & 24 \\ 1 & 1 \end{bmatrix}$

De donde $y_3 = \begin{bmatrix} 3/4 \\ 1/4 \end{bmatrix}$

Ello determina la variable que deja la base al calcular

$\lambda_3 = \min\{(7/8)/(3/4)\ ,\ (1/8)/(1/4)\} = 1/2$ deja la base λ_2.

Tercera iteración.

Variables básicas λ_1 ($\mathbf{x^1}=(1,1,3)^T$) y λ_3 ($\mathbf{x^3}=(1,3,3)^T$)

$A\mathbf{x^3} = 18$ $\quad$ $\mathbf{c^T x^3} = [-4 \ -2 \ -6]\mathbf{x^3} = -28$

$$B=\begin{bmatrix} 16 & 18 \\ 1 & 1 \end{bmatrix} \quad Bx_B=\begin{bmatrix} 17 \\ 1 \end{bmatrix} \quad \lambda_1=1/2 \ \ \lambda_3=1/2 \quad B^T \begin{bmatrix} \omega \\ \alpha \end{bmatrix} = \begin{bmatrix} -24 \\ -28 \end{bmatrix}$$

$\omega = -2 \quad \alpha = 8$

Como $\mathbf{c^T} - \omega\mathbf{A} = [-4 \ -2 \ -6] + 2[3 \ 1 \ 4] = [2 \ 0 \ 2]$

(SP) Min $2x_1 + 2x_3 - 8$

s.a. $1 \le x_1 \le 3, \ 1 \le x_2 \le 3, \ 1 \le x_3 \le 3.$

Una solución óptima $\mathbf{x^4}=(1,3,1)^T$

Como v(SP)=-4 se incorpora a la base λ_4.

Con $\mathbf{x^4}=(1,3,1)^T$ calculamos

$A\mathbf{x^4} = [\ 3\ \ 1\ \ 4]\mathbf{x^4} = 10$

Se resuelve $By_4 = \begin{bmatrix}10\\1\end{bmatrix}$ con $B = \begin{bmatrix}16 & 18\\1 & 1\end{bmatrix}$

de donde $y_4 = \begin{bmatrix}4\\-3\end{bmatrix}$

Ello determina la variable que deja la base al calcular

$\lambda_4 = \min\{(1/2)/4\ ,\ .\ \ \} = 1/8$ deja la base λ_1.

Cuarta iteración.

Variables básicas λ_3 ($\mathbf{x^3}=(1,3,3)^T$) y λ_4 ($\mathbf{x^4}=(1,3,1)^T$)

$A\mathbf{x^4} = 10$ $\quad$ $\mathbf{c}^T\mathbf{x^4} = [-4\ -2\ -6]\mathbf{x^4} = -16$

$$B=\begin{bmatrix} 18 & 10 \\ 1 & 1 \end{bmatrix} \quad Bx_B=\begin{bmatrix} 17 \\ 1 \end{bmatrix} \quad \lambda_3=7/8 \quad \lambda_4=1/8 \quad B^T\begin{bmatrix} \omega \\ \alpha \end{bmatrix} = \begin{bmatrix} -28 \\ -16 \end{bmatrix}$$

$\omega = -3/2 \quad \alpha = -1$

Como $\mathbf{c}^T - \omega\mathbf{A} = [-4\ -2\ -6] + (3/2)[3\ 1\ 4] = [1/2\ -1/2\ 0]$

(SP) Min $x_1/2 + -x_2/2 + 1$

s.a. $1 \le x_1 \le 3,\ 1 \le x_2 \le 3,\ 1 \le x_3 \le 3.$

Una solución óptima $\mathbf{x^4}=(1,3,1)^T$

Como v(SP)=0 alcanzamos la solución óptima de PM.

Alcanzada la solución óptima de PM en las variables λ_j, se puede entonces obtener la solución en las variables originales de acuerdo a:

$$\mathbf{x}^* = \lambda_1\mathbf{x}^1 + \lambda_2\mathbf{x}^2 + \lambda_3\mathbf{x}^3 + \lambda_4\mathbf{x}^4 + \ldots + \lambda_t\mathbf{x}^t$$
$$= 0\mathbf{x}^1 + 0\mathbf{x}^2 + (7/8)\mathbf{x}^3 + (1/8)\mathbf{x}^4 + \ldots + 0\mathbf{x}^t$$
$$= (7/8)(1,3,3) + (1/8)(1,3,1) = (1, 3, 11/4)$$

Por lo tanto la solución óptima es:

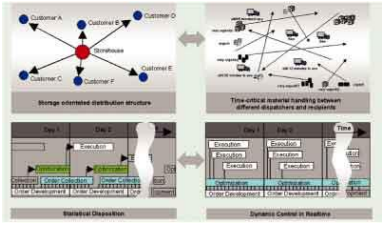
$x^*_1 = 1 \quad x^*_2 = 3 \quad x^*_3 = 11/4$

Con un valor óptimo: $v(P) = 53/2$

Problema de Transporte con múltiples productos.

El problema consiste en decidir cuántas unidades trasladar de múltiples productos desde ciertos puntos de origen a ciertos puntos de destino.

Dados los costos unitarios de transporte, restricciones de capacidad, la oferta y demanda para cada uno de los productos, el problema consiste en minimizar los costos de transporte de modo de satisfacer la demanda por los distintos productos.



Variables de decisión.

$T_{i,j,p}$: unidades transportadas desde el origen i al destino j del producto p

Min $\sum_{i \in O} \sum_{j \in D} \sum_{p \in P} c_{i,j,p} T_{i,j,p}$

s.a.

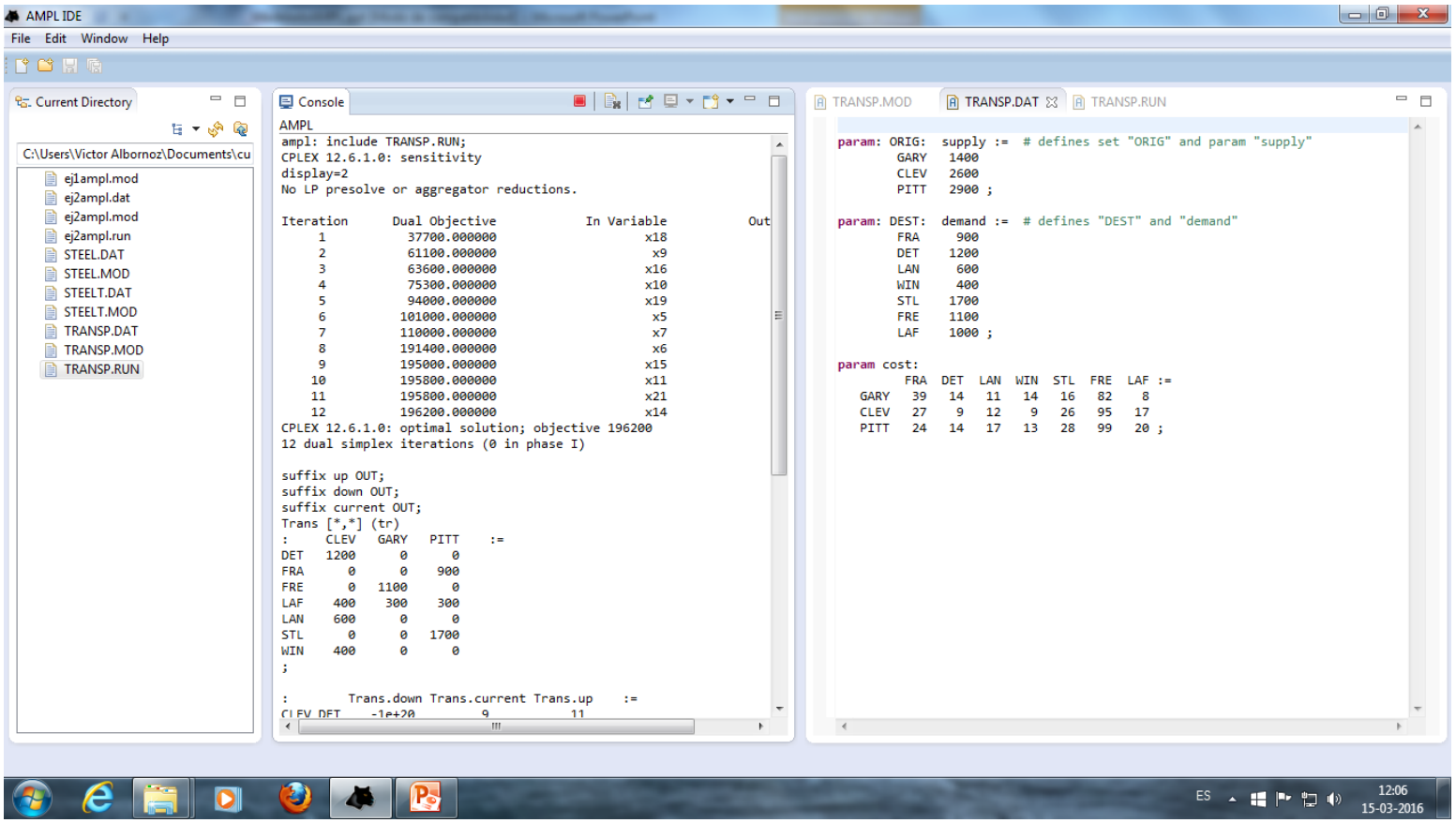
$\sum_{j \in D} T_{i,j\,p} \leq s_{i,p}$ para $i \in O$; $p \in P$

$\sum_{i \in O} T_{i,j,p} = d_{j,p}$ para $j \in D$; $p \in P$

$\sum_{p \in P} T_{i,j,p} \leq limit_{i,j}$ para $i \in O$; $j \in D$

$T_{i,j,p} \geq 0$ $i \in O$; $j \in D$; $p \in P$

En AMPL el modelo descrito y una instancia particular del mismo está en los archivos MULTI.MOD y MULTI.DAT.



Implementación del algoritmo de Dantzig y Wolfe en AMPL, aplicado al mismo problema de transporte para múltiples productos puede consultarse en multi1.mod, multi1.run y multi1.dat

Dantzig & Wolfe propusieron originalmente esta idea aplicada a problemas de programación lineal con una estructura de restricciones bloque angular que se detalla a continuación.

$$\begin{bmatrix} A_1 & A_2 & A_3 & \cdots & A_q \\ D_1 & & & & \\ & D_2 & & & \\ & & D_3 & & \\ & & & \ddots & \\ & & & & D_q \end{bmatrix}$$

Consideremos el siguiente problema de Programación Lineal:

$$
\begin{array}{llllllllll}
Min & c_1^T x_1 & + & c_2^T x_2 & + & \cdots & + & c_q^T x_q & & \\
s.a & A_1 x_1 & + & A_2 x_2 & + & \cdots & + & A_q x_q & = & b \\
 & D_1 x_1 & & & & & & & \leq & d_1 \\
 & & & D_2 x_2 & & & & & \leq & d_2 \\
 & & & & & \ddots & & & & \\
 & & & & & & & D_q x_q & \leq & d_q \\
 & x_1, & & x_2, & & \ldots & & x_q & \geq & 0
\end{array}
\tag{5.4}
$$

donde $x_i \in \mathbb{R}^{n_i}$, $b \in \mathbb{R}^{m_0}$, $d_i \in \mathbb{R}^{m_i}$, A_i de $m_0 \times n_i$ y D_i de $m_i \times n_i$.

Dado lo anterior, el problema (5.4) es equivalente a (5.1) tomando

$$X = \{x \in \mathbb{R}^n \ / \ x = (x_1, x_2, \ldots, x_q),\ x_i \in X_i\}$$

donde

$$X_i = \{x_i \in \mathbb{R}^{n_i} \ / \ D_i x_i \leq d_i,\ x_i \geq 0\}$$

con A una matriz de $m_0 \times n$, $A = [A_1\ A_2\ \cdots\ A_q]$ y $n = n_1 + \cdots + n_q$.

Por simplicidad suponemos que cada conjunto X_i es acotado.

Denotamos por $\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \ldots, \boldsymbol{x}_i^{t_i}$ los vértices del poliedro X_i con $i = 1, \ldots, q$.

Entonces, para cada $x_i \in X_i$

$$x_i = \sum_{j=1}^{t_i} \lambda_i^j \boldsymbol{x}_i^j \qquad \sum_{j=1}^{t_i} \lambda_i^j = 1 \qquad \lambda_i^j \geq 0 \ \ \forall j$$

De este modo

$$c_i^T x_i = c_i^T \left(\sum_{j=1}^{t_i} \lambda_i^j \boldsymbol{x}_i^j\right) = \sum_{j=1}^{t_i} (c_i^T \boldsymbol{x}_i^j)\lambda_i^j$$

$$A_i x_i = A_i \left(\sum_{j=1}^{t_i} \lambda_i^j \boldsymbol{x}_i^j\right) = \sum_{j=1}^{t_i} (A_i \boldsymbol{x}_i^j)\lambda_i^j$$

Así el problema (5.4) equivale al siguiente P.M.:

$$\begin{array}{lrcccl}
Min & \sum_{j=1}^{t_1} (c_1^T \boldsymbol{x}_1^j)\lambda_1^j & + \cdots + & \sum_{j=1}^{t_q} (c_q^T \boldsymbol{x}_q^j)\lambda_q^j & & \\
s.a & & & & & \\
& \sum_{j=1}^{t_1} (A_1 \boldsymbol{x}_1^j)\lambda_1^j & + \cdots + & \sum_{j=1}^{t_q} (A_q \boldsymbol{x}_q^j)\lambda_q^j & = & b \\
& \sum_{j=1}^{t_1} \lambda_1^j & & & = & 1 \\
& & \ddots & & & \\
& & & \sum_{j=1}^{t_q} \lambda_q^j & = & 1 \\
& \lambda_1^j \geq 0, & \cdots & \lambda_q^j \geq 0 & &
\end{array} \qquad (5.5)$$

Supongamos que tenemos una solución básica factible del problema (5.5) en las nuevas variables λ_i^j, y que la matriz de base asociada a esta solución es B.

Como antes, denotamos por $w \in \mathbb{R}^{m_0}$ y $\alpha = (\alpha_1, \ldots, \alpha_q)^T \in \mathbb{R}^q$ los multiplicadores asociados a las restricciones de (5.5) entonces:

$$\begin{pmatrix} w \\ \alpha \end{pmatrix} = c_B B^{-1}$$

Las condiciones de optimalidad que deben satisfacerse para cada λ_i^j no-básica son:

$$0 \leq c_i^T \boldsymbol{x}_i^j - (w^T \; \alpha^T) \begin{pmatrix} A_i \boldsymbol{x}_i^j \\ e_i \end{pmatrix} = c_i^T \boldsymbol{x}_i^j - w^T A_i \boldsymbol{x}_i^j - \alpha_i$$

Para determinar si existe o no un costo reducido negativo, se resuelve el siguiente subproblema:

$$\begin{array}{ll} Min & (c_i^T - w^T A_i) x_i \\ s.a & x_i \in X_i \end{array} \qquad (5.6)$$

Denotamos por $\boldsymbol{x}_i^k$ la solución óptima del subproblema, con valor óptimo $(c_i^T - w^T A_i)\boldsymbol{x}_i^k$.

Si se cumple que

$$(c_i^T - w^T A_i)\boldsymbol{x}_i^k - \alpha_i \geq 0$$

entonces, todos los otros vértices de X_i verifican la condición de optimalidad. Si esto último además se cumple para cada $i = 1, \ldots, q$, el método se detiene pues hemos hallado la solución óptima.

En caso contrario, existe un vértice $\boldsymbol{x}_i^k$ que da origen a un costo reducido negativo para la respectiva variable λ_i^k que entra a la nueva base y se sigue con el Simplex.

Implementación del algoritmo en AMPL, usando varios subproblemas para el ejemplo del problema de transporte con múltiples productos ver archivos multi2.mod, multi2.run y multi2.dat

Firefox | AMPL -- loop2 examples index page

http://www.ampl.com/NEW/LOOP2/index.html

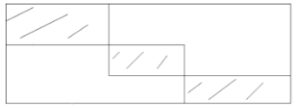
INDEX TO EXAMPLES

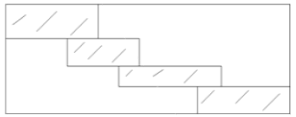
LOOPING AND TESTING 2:
implementing algorithms through AMPL scripts

Script	Uses	Implements
cut1.run	cut1.mod cut.dat	Gilmore-Gomory column generation procedure for the cutting-stock (roll trim) problem
cut2.run	cut2.mod cut.dat	Same as `cut1.run`, but using an alternative arrangement wherein problems are defined immediately before before their members are declared
cut3.run	cut1.mod cut.dat	Same as `cut1.run`, but with better formatting of output
multi1.run	multi1.mod multi1.dat	Dantzig-Wolfe decomposition for a multi-commodity transportation problem, using a single subproblem
multi1a.run	multi1.mod multi1.dat	Same as `multi1.run`, but using the same `repeat` loop for both phase I (infeasible) and phase II (feasible).
multi2.run	multi2.mod multi2.dat	Same as `multi1.run`, but using a separate subproblem for each product; subproblems are represented in AMPL by an indexed collection of named problems
multi3.run	multi3.mod multi3.dat	Same as `multi2.run`, except that the separate subproblems are realized by changing the data to a single AMPL named problem
stoch1.run	stoch1.mod stoch.dat	Benders decomposition for a stochastic programming variant of a multi-period production problem (see Exercise 4-5)
stoch2.run	stoch2.mod stoch.dat	Same as `stoch1.run`, but using a separate subproblem for each scenario; subproblems are represented in AMPL by an indexed collection of named problems
stoch3.run	stoch3.mod stoch.dat	Same as `stoch2.run`, except that the separate subproblems are realized by changing the data to a single AMPL named problem
trnloc1.run *revised!*	trnloc1.mod trnloc.dat	Benders decomposition for a location-transportation problem (original model in trnloc.mod)
trnloc2a.run	trnloc2a.mod trnloc2.dat	Lagrangian relaxation for a location-transportation problem: LP relaxation bound is poor, and subproblem has the integrality property so no improvement can be made
trnloc2b.run	trnloc2b.mod trnloc2.dat	Same as `trnloc2a.run`, but model has upper limits on the `Ship` variables: LP relaxation bound is still poor, but subproblem does *not* have the integrality property and considerable improvement is made
trnloc2c.run	trnloc2c.mod trnloc2.dat	Same as `trnloc2b.run`, but model has 0-1 constraints disaggregated: LP relaxation bound is good, but subproblem has the integrality property and no

ES 18:00 10-10-2013

CONTENIDOS





6.- Conclusiones, extensiones y palabras finales.

Como hemos comentado, existen problemas de gran tamaño donde la presencia de restricciones con una determinada estructura facilita su resolución al emplear métodos de descomposición.

Sin embargo, hemos revisado únicamente técnicas clásicas de descomposición en modelos de programación lineal.

$$\begin{bmatrix} A_1 & A_2 & A_3 & \cdots & A_q \\ D_1 & & & & \\ & D_2 & & & \\ & & D_3 & & \\ & & & \ddots & \\ & & & & D_q \end{bmatrix} \qquad \begin{bmatrix} D_1 & & & & F_1 \\ & D_2 & & & F_2 \\ & & D_3 & & F_3 \\ & & & \ddots & \\ & & & D_q & F_q \end{bmatrix}$$

$$A = \begin{pmatrix} A_{11} & & & & & \\ A_{21} & A_{22} & A_{23} & & & \\ & & A_{33} & & & \\ & & A_{43} & A_{44} & A_{45} & \\ & & & & A_{55} & \\ & & & & A_{65} & A_{66} \end{pmatrix}$$

A stair matrix of the type I

or

$$A = \begin{pmatrix} A_{11} & A_{12} & & & & \\ & A_{22} & & & & \\ & A_{32} & A_{33} & A_{34} & & \\ & & & A_{44} & & \\ & & & A_{54} & A_{55} & A_{56} \\ & & & & & A_{66} \end{pmatrix}.$$

A stair matrix of the type II

Otros métodos de descomposición incluyen a:

L-Shaped decomposition method of van Slyke & Wets (1969).

Generalized Benders Decomposition of Geoffrion (1972).

Simplicial decomposition of von Hohembalken (1977).

Cross Decomposition of van Roy (1983).

Horizontal Decomposition of Meijboom (1986).

Lagrangean decomposition of Guignard and Kim (1987).

Local Decomposition of van de Panne (1987).

Nested Decomposition, Robinson (1989).

Stochastic Decomposition, Higle and Sen (1991).

Column Generation in IP, Vanderveck and Wolsey (1996).

L-Shaped Method for IPSP, Caroe and Tind (1998).

Branch-and-price, Barnhart et al. (1998).

Benders and BFC for 0-1 mixed SP, Escudero et al. (2007).

Two-Stage Column Generation, Salani and Vacca (2008).

Stochastic Scenario Decomposition (MSSP), Higle et al. (2010)

Cluster Benders Decomposition, Aramburu et al. (2012).

Primal-dual column generation method, Gondzio *et al*. (2013).

Bi-objective column generation algorithm, Moradi *et al*. (2015).

Benders method for bilevel programs, Bagloee et al. (2016), Fontaine and Minner (2017) and Che et al. (2019).

Existen numerosos problemas que contribuyen a la toma de decisiones y que pueden ser abordados con modelos de optimización, pero cuya resolución plantea desafíos algorítmicos.

Los Métodos de Descomposición proveen técnicas numéricas de optimización para resolver adecuadamente problemas de gran tamaño.

GRACIAS POR SU INTERÉS Y ASISTENCIA

TÉCNICAS DE DESCOMPOSICIÓN EN PROGRAMACIÓN MATEMÁTICA

Dr. Víctor M. Albornoz S.
Departamento de Industrias.
Campus Santiago Vitacura, Chile.
Universidad Técnica Federico Santa María

Instituto de Computación, Facultad de Ingeniería, UdelaR.
Montevideo, lunes 20 al viernes 24 de mayo de 2024