

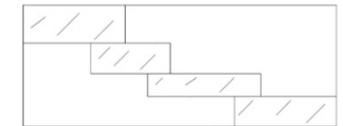
# TÉCNICAS DE DESCOMPOSICIÓN EN PROGRAMACIÓN MATEMÁTICA

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

Instituto de Computación, Facultad de Ingeniería, UdelaR.  
Montevideo, viernes 23 de Mayo de 2025

# CONTENIDO

1. Introducción a los Métodos de Descomposición.
2. Formulación y resolución de modelos en AMPL.
3. Método de Benders.
4. Generación de Columnas.
- 5. Método de Dantzig & Wolfe.**
6. Conclusiones, Extensiones y palabras finales.



## 5.- Método de Dantzig & Wolfe.

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

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

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

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

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

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

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

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

$$\text{Min } (c^T \mathbf{x}^1)\lambda_1 + (c^T \mathbf{x}^2)\lambda_2 + \dots + (c^T \mathbf{x}^t)\lambda_t \quad (5.2)$$

$$\text{s.a. } (A\mathbf{x}^1)\lambda_1 + (A\mathbf{x}^2)\lambda_2 + \dots + (A\mathbf{x}^t)\lambda_t = \mathbf{b}$$

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

$$\lambda_1 \geq 0, \lambda_2 \geq 0, \dots, \lambda_t \geq 0.$$

Si  $B$  es una matriz de base asociada a una solución básica factible del *Problema Maestro*,  $\omega$  y  $\alpha$  las respectivas variables duales asociadas a las restricciones en (5.2), entonces:

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

Así, las condiciones de optimalidad que deben verificarse para cada variable no-básica  $\lambda_j$  son:

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

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

$$\begin{aligned} \text{Min } & (c^T - \omega^T A)\mathbf{x} - \alpha && (5.3) \\ \text{s.a. } & \mathbf{x} \in X \end{aligned}$$

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

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

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  $\lambda_k$  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  $\lambda_k$  como combinación de las actuales columnas en B, resolviendo:

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

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:

$$\text{Max } 4x_1 + 2x_2 + 6x_3$$

$$\text{s.a. } 3x_1 + x_2 + 4x_3 \leq 17$$

$$1 \leq x_1 \leq 3$$

$$1 \leq x_2 \leq 3$$

$$1 \leq x_3 \leq 3.$$

El problema puede expresarse como:

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

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

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

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

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

El problema puede expresarse como:

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

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

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

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

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

PROBLEMA MAESTRO (PM):

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

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

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

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

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

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

Las condiciones de optimalidad que deben verificarse para cada variable no-básica  $\lambda_j$  son:

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

Lo que se verifica resolviendo el SUBPROBLEMA:

$$\begin{aligned} \text{(SP)} \quad & \text{Min } (c^T - \omega A) x - \alpha \\ & \text{s.a. } x \in X \end{aligned}$$

En el caso del ejemplo:

$$\begin{aligned} \text{(SP)} \quad & \text{Min } (-4 - 3\omega)x_1 + (-2 - \omega)x_2 + (-6 - 4\omega)x_3 - \alpha \\ & \text{s.a. } 1 \leq x_1 \leq 3, \quad 1 \leq x_2 \leq 3, \quad 1 \leq x_3 \leq 3. \end{aligned}$$

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  $\lambda_1$  asociada a un vértice inicial, por ejemplo  $\mathbf{x}^1 = (1, 1, 3)^T$ .

Primera iteración.

Variables básicas  $s_1$  y  $\lambda_1$  ( $\mathbf{x}^1=(1,1,3)^T$ )

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

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

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

$$\begin{aligned} \text{(SP)} \quad & \text{Min} \quad -4x_1 - 2x_2 - 6x_3 + 24 \\ & \text{s.a.} \quad 1 \leq x_1 \leq 3, \quad 1 \leq x_2 \leq 3, \quad 1 \leq x_3 \leq 3. \end{aligned}$$

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

Como  $v(\text{SP}) = -12$  se incorpora a la base  $\lambda_2$ .

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  $\mathbf{y}_2 = \begin{bmatrix} 8 \\ 1 \end{bmatrix}$

Ello determina la variable que deja la base al calcular

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

Segunda iteración.

Variables básicas  $\lambda_1 (\mathbf{x}^1=(1,1,3)^T)$  y  $\lambda_2 (\mathbf{x}^2=(3,3,3)^T)$

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

$$B = \begin{bmatrix} 16 & 24 \\ 1 & 1 \end{bmatrix} \quad Bx_B = \begin{bmatrix} 17 \\ 1 \end{bmatrix} \quad \lambda_1 = 7/8 \quad \lambda_2 = 1/8 \quad B^T \begin{bmatrix} \omega \\ \alpha \end{bmatrix} = \begin{bmatrix} -24 \\ -36 \end{bmatrix}$$
$$\omega = -3/2 \quad \alpha = 0$$

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

(SP) Min  $x_1/2 - x_2/2$   
s.a.  $1 \leq x_1 \leq 3, \ 1 \leq x_2 \leq 3, \ 1 \leq x_3 \leq 3.$

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

Como  $v(\text{SP}) = -1$  se incorpora a la base  $\lambda_3$ .

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

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

Se resuelve  $B\mathbf{y}_3 = \begin{bmatrix} 18 \\ 1 \end{bmatrix}$  con  $B = \begin{bmatrix} 16 & 24 \\ 1 & 1 \end{bmatrix}$

De donde  $\mathbf{y}_3 = \begin{bmatrix} 3/4 \\ 1/4 \end{bmatrix}$

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  $\lambda_2$ .

Tercera iteración.

Variables básicas  $\lambda_1 (\mathbf{x}^1=(1,1,3)^T)$  y  $\lambda_3 (\mathbf{x}^3=(1,3,3)^T)$

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

$$B = \begin{bmatrix} 16 & 18 \\ 1 & 1 \end{bmatrix} \quad B\mathbf{x}_B = \begin{bmatrix} 17 \\ 1 \end{bmatrix} \quad \lambda_1 = 1/2 \quad \lambda_3 = 1/2 \quad B^T \begin{bmatrix} \omega \\ \alpha \end{bmatrix} = \begin{bmatrix} -24 \\ -28 \end{bmatrix}$$
$$\omega = -2 \quad \alpha = 8$$

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

$$\begin{aligned} \text{(SP)} \quad & \text{Min} \quad 2x_1 + 2x_3 - 8 \\ & \text{s.a.} \quad 1 \leq x_1 \leq 3, \quad 1 \leq x_2 \leq 3, \quad 1 \leq x_3 \leq 3. \end{aligned}$$

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

Como  $v(\text{SP})=-4$  se incorpora a la base  $\lambda_4$ .

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

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

$$\text{Se resuelve } B\mathbf{y}_4 = \begin{bmatrix} 10 \\ 1 \end{bmatrix} \text{ con } B = \begin{bmatrix} 16 & 18 \\ 1 & 1 \end{bmatrix}$$

$$\text{de donde } \mathbf{y}_4 = \begin{bmatrix} 4 \\ -3 \end{bmatrix}$$

Ello determina la variable que deja la base al calcular

$$\lambda_4 = \min\{(1/2)/4, \dots\} = 1/8 \text{ deja la base } \lambda_1.$$

Cuarta iteración.

Variables básicas  $\lambda_3 (\mathbf{x}^3=(1,3,3)^T)$  y  $\lambda_4 (\mathbf{x}^4=(1,3,1)^T)$

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

$$B = \begin{bmatrix} 18 & 10 \\ 1 & 1 \end{bmatrix} \quad Bx_B = \begin{bmatrix} 17 \\ 1 \end{bmatrix} \quad \lambda_3 = 7/8 \quad \lambda_4 = 1/8 \quad B^T \begin{bmatrix} \omega \\ \alpha \end{bmatrix} = \begin{bmatrix} -28 \\ -16 \end{bmatrix}$$
$$\omega = -3/2 \quad \alpha = -1$$

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

(SP) Min  $x_1/2 + -x_2/2 + 1$   
s.a.  $1 \leq x_1 \leq 3, 1 \leq x_2 \leq 3, 1 \leq x_3 \leq 3.$

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

Como  $v(\text{SP})=0$  alcanzamos la solución óptima de PM.

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

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

Por lo tanto la solución óptima es:

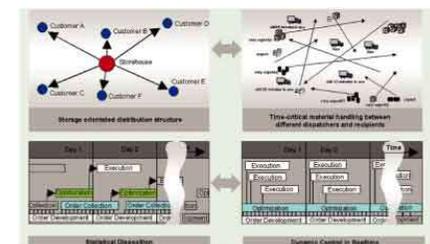
$$x^*_1 = 1 \quad x^*_2 = 3 \quad x^*_3 = 11/4$$

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  $p$

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

*s.a.*

$$\sum_{j \in D} T_{i,j,p} \leq s_{i,p} \quad \text{para } i \in O; p \in P \quad (1)$$

$$\sum_{i \in O} T_{i,j,p} = d_{j,p} \quad \text{para } j \in D; p \in P \quad (2)$$

$$\sum_{p \in P} T_{i,j,p} \leq \text{limit}_{i,j} \quad \text{para } i \in O; j \in D \quad (3)$$

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

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

The screenshot displays the AMPL IDE interface. On the left, the 'Current Directory' pane shows a file tree for 'C:\Users\Victor Alborno\Documents\cu' containing files like 'ej1ampl.mod', 'ej2ampl.dat', 'STEEL.DAT', 'STEEL.MOD', 'TRANSP.DAT', 'TRANSP.MOD', and 'TRANSP.RUN'. The central 'Console' window shows the following output:

```
AMPL
ampl: include TRANSP.RUN;
CPLEX 12.6.1.0: sensitivity
display=2
No LP presolve or aggregator reductions.

Iteration   Dual Objective   In Variable   Out
1           37700.000000    x18
2           61100.000000    x9
3           63600.000000    x16
4           75300.000000    x10
5           94000.000000    x19
6           101000.000000   x5
7           110000.000000   x7
8           191400.000000   x6
9           195000.000000   x15
10          195800.000000   x11
11          195800.000000   x21
12          196200.000000   x14

CPLEX 12.6.1.0: optimal solution; objective 196200
12 dual simplex iterations (0 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;
Trans ["*.*"] (tr)
:   CLEV   GARY   PITT   :=
DET  1200    0    0
FRA   0    0   900
FRE   0  1100    0
LAF  400   300   300
LAN  600    0    0
STL   0    0  1700
WIN  400    0    0
;

:   Trans.down Trans.current Trans.up   :=
CLEV DET  -1e+20    0    11
```

On the right, the 'TRANSP.DAT' window shows the following data:

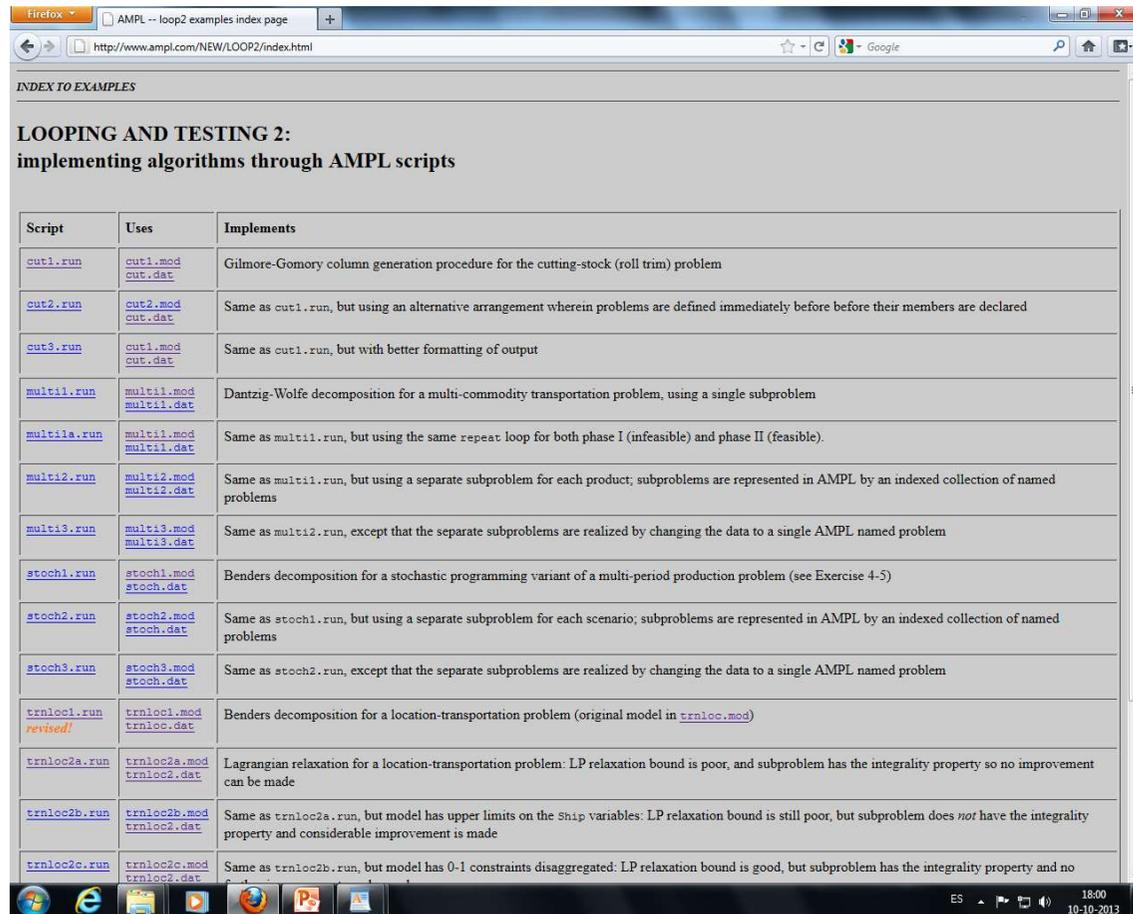
```
param: ORIG: supply := # defines set "ORIG" and param "supply"
GARY  1400
CLEV  2600
PITT  2900 ;

param: DEST: demand := # defines "DEST" and "demand"
FRA   900
DET  1200
LAN   600
WIN   400
STL  1700
FRE  1100
LAF  1000 ;

param cost:
      FRA  DET  LAN  WIN  STL  FRE  LAF :=
GARY  39  14  11  14  16  82  8
CLEV  27  9  12  9  26  95  17
PITT  24  14  17  13  28  99  20 ;
```

The Windows taskbar at the bottom shows the system clock as 12:06 on 15-03-2016.

# 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](#)



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

## 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.

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

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

A stair matrix of the type I

$$\text{or } A = \begin{pmatrix} A_{11} & A_{12} & & & & \\ & A_{22} & & & & \\ & & A_{33} & & & \\ & & & A_{44} & & \\ & & & & A_{55} & \\ & & & & & A_{66} \end{pmatrix}$$

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 Hohenbalken (1977).

Cross Decomposition of van Roy (1983).

Horizontal Decomposition of Meijboom (1986).

Lagrangian decomposition of Guignard and Kim (1987).

Local Decomposition of van de Panne (1987).

Nested Decomposition, Robinson (1989).

Stochastic Decomposition, Hight 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.*