

# 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

0. Clase Bienvenida.

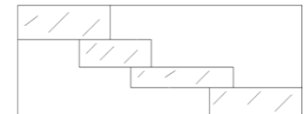
1. Introducción a los Métodos de Descomposición.

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.

#### 4.- Generación de Columnas.

De manera resumida, el *Método de Generación de Columnas* consiste en aplicar el método *Simplex Revisado* a un problema de programación lineal donde no se conocen todas las columnas de la matriz de restricciones y donde las columnas necesarias a obtener serán resultado de la resolución de un nuevo problema que determina el menor costo reducido para la actual solución básica factible o solución óptima de un problema (maestro) reducido.

Si el problema de Programación Lineal a resolver con Simplex Revisado es:

$$\text{Min} \quad c_1x_1 + c_2x_2 + \dots + c_nx_n$$

$$\text{s.a.} \quad a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \quad i=1, \dots, m \quad (1)$$

$$x \geq 0$$

donde  $x \in \mathbb{R}^n$ . Asumimos que la matriz de restricciones es de rango máximo, esto es que tiene un conjunto inicial de  $m$  columnas *l.i.*, que suponemos son las  $m$  primeras y que definen una solución básica factible inicial.

En el problema de corte de piezas, el método parte de una solución básica factible que en general es fácil de conseguir (asumiendo la relajación continua de las variables de decisión).

En efecto, se puede partir de  $m$  patrones iniciales cada uno de los cuales contiene únicamente  $\lfloor W/w_i \rfloor$  trozos del tamaño  $w_i$ , es decir para  $i=1, \dots, m$  se define:

$$a_{ij} = \lfloor W/w_i \rfloor \text{ si } i=j \text{ y } 0 \text{ sino para } j=1, \dots, m$$

En lo que sigue, denotamos entonces por  $B$  la matriz de base formada por dichas columnas de la matriz  $A$  y por  $D$  las restantes columnas.

Sea  $x_B$  solución del sistema  $Bx_B=b$  con  $x_B \geq 0$ , entonces el vector  $x=(x_B, x_D)=(B^{-1}b, 0)^T$  es llamado una *solución básica factible* del problema (1).

A partir de la matriz de base  $B$ , hacemos la siguiente partición de las variables y parámetros del problema

$$A = [B \ D] \quad c^T = (c_B^T \ c_D^T) \quad x^T = (x_B^T \ x_D^T)$$

De modo que (1) equivale a resolver:

$$\begin{aligned} \text{Min} \quad & c_B^T x_B + c_D^T x_D \\ \text{s.a.} \quad & Bx_B + Dx_D = b \\ & x_B \geq 0, x_D \geq 0 \end{aligned} \tag{2}$$

Dado lo anterior, la función objetivo del problema corresponde a:

$$\begin{aligned}c^T x &= c_B^T x_B + c_D^T x_D \\ &= c_B^T (B^{-1}b - B^{-1} D x_D) + c_D^T x_D \\ &= c_B^T B^{-1} b + (c_D^T - c_B^T B^{-1} D) x_D\end{aligned}$$

al vector  $c_D^T - c_B^T B^{-1} D$  se le denomina *vector de costos reducidos* de las variables no-básicas.



Si todas las componentes del vector de *costos reducidos* son no-negativas tenemos que la *actual solución básica factible es óptima*.

Sino, el Método Simplex Revisado construye una nueva solución básica factible, incorporando una nueva variable básica (de costo reducido negativo) y eliminando una variable básica de la actual solución básica.

Volviendo al problema de corte de piezas, si  $B$  es la respectiva matriz de base asociada a esta solución básica factible, es posible saber si es óptima mediante el cálculo de los *costos reducidos* para cada variable no-básica asociada a un patrón  $x_j$ , verificando si se cumple:

$$1 - c_B^T B^{-1} a^j = 1 - \lambda^T a^j \geq 0$$

con  $\lambda^T = c_B^T B^{-1}$  para cada patrón  $j$ .

Lo anterior equivale igualmente a que se cumpla:

$$\min_{j=1, \dots, n} \{1 - \lambda^T a^j\} \geq 0$$

Para verificar esto último no es necesario conocer todos los patrones sino simplemente verificar si le cumple aquel patrón con el menor costo reducido, esto es, aquel que minimiza el siguiente *Suproblema*:

$$\text{Min } 1 - \lambda^T \mathbf{a}$$

$$\text{s.a. } a_1 w_1 + a_2 w_2 + \dots + a_m w_m \leq W$$

$$\mathbf{a} \geq \mathbf{0}, \text{ entero,}$$

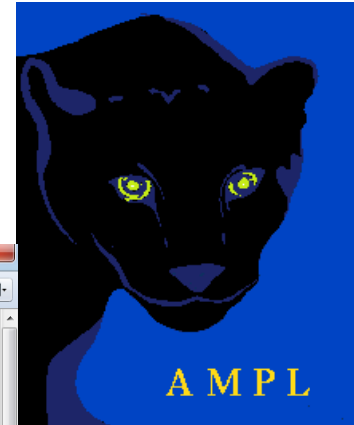
donde  $w_i$ , es el tamaño de los trozos demandados al cortar piezas de tamaño  $W$ .

Así entonces, si  $\mathbf{a}=\mathbf{a}^k$  es el patrón donde se alcanza la solución óptima del Subproblema y este verifica:  $1-\lambda^T\mathbf{a}^k \geq 0$ , entonces todos los otros patrones verifican la no-negatividad y hemos alcanzado la solución óptima. Sino se sigue con el M.Simplex.

Notar igualmente que  $\lambda^T = \mathbf{c}_B^T\mathbf{B}^{-1}$  es igualmente la solución óptima dual del problema que incluye únicamente las columnas existentes y que define lo que será el Problema Maestro Reducido (de corte de piezas).

# Implementación en AMPL

[www.ampl.com](http://www.ampl.com)



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
<a href="#">cut1.run</a>	<a href="#">cut1.mod</a> <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> <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> <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> <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> <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> <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> <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> <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> <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> <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> <i>revised!</i>	<a href="#">trnloc1.mod</a> <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> <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> <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> <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

Windows taskbar: ES 18:00 10-10-2013

Si discute a continuación el detalle de la implementación del método tomando los archivos cut1.mod, cut1.run y cut1.dat:

The screenshot displays the AMPLIDE software interface. On the left, a file explorer shows the current directory: C:\Users\Victor Albornoz\Documents\an. The console window in the center shows the following output:

```
AMPL
: 20 45 50 55 75 :=
1 5 0 0 0 0
2 0 2 0 0 0
3 0 0 2 0 0
4 0 0 0 2 0
5 0 0 0 0 1
6 1 2 0 0 0
7 1 0 0 0 1
8 3 0 1 0 0
;

Cut [*] :=
1 0
2 0
3 8.25
4 5
5 0
6 17.5
7 8
8 7.5
;

Gurobi 5.1.0: optimal solution; objective 47
1 simplex iterations
Cut [*] :=
1 0
2 0
3 8
4 5
5 0
6 18
7 8
8 8
;

ampl:
```

On the right, the code editor shows the following AMPL code:

```
cut1.dat  cut1.mod  cut1.run
# -----
# GILMORE-GOMORY METHOD FOR
# CUTTING STOCK PROBLEM
# -----

option solver gurobi;
option solution_round 6;

model cut1.mod;
data cut1.dat;

problem Cutting_Opt: Cut, Number, Fill;
option relax_integrality 1;
option presolve 0;

problem Pattern_Gen: Use, Reduced_Cost, Width_Limit;
option relax_integrality 0;
option presolve 1;

let nPAT := 0;

for {i in WIDTHS} {
let nPAT := nPAT + 1;
let nbr[i,nPAT] := floor(roll_width/i);
let {i2 in WIDTHS: i2 <> i} nbr[i2,nPAT] := 0;
};

repeat {
solve Cutting_Opt;

let {i in WIDTHS} price[i] := Fill[i].dual;

solve Pattern_Gen;

if Reduced_Cost < -0.00001 then {
let nPAT := nPAT + 1;
let {i in WIDTHS} nbr[i,nPAT] := Use[i];
}
}
```

Ejemplo de modelado y resolución de un problema en Agricultura de Precisión mediante Generación de Columnas.

El problema comprende la partición de un terreno en zonas rectangulares en apoyo a la planificación de manejo de un terreno para explotación agrícola.



# El modelo propuesto se resuelve mediante la aplicación del método de Generación de Columnas en apoyo a la determinación de dichas zonas de manejo agrícola (Albornoz y Ñanco, 2016).

## An Empirical Design of a Column Generation Algorithm Applied to a Management Zone Delineation Problem

Victor M. Albornoz and Linco J. Ñanco

### 1 Introduction

The problem of delineating site-specific management zones in agricultural fields arises in the context of precision agriculture, where the control of variability in soil properties is essential to increase productivity and crop quality. Dividing a field into high internal homogeneity zones with respect to any soil property (pH, organic material, phosphorus, etc.) allows the farmer to face this variability. Furthermore, establishing a rectangular management zone partition provides practical advantages with respect to the use of agricultural machinery.

This paper presents different strategies based on the column generation technique to solve the problem proposed in [2], focused on the delineation of rectangular management zones, considering a fixed internal homogeneity level to solve the problem, measured by the relative variance of partition criterion used to assess the efficiency of the field division (see [6]).

Hereinafter, the problem discussed is considered as: given a set of sample points in a field,  $S = \{1, \dots, N\}$ , and a set of potential quarters  $Z = \{1, \dots, K\}$ , where each potential quarter covers a subset of these sample points, find the subset of  $Z$  with the minimum number of elements, which is a partition of field, with a given maximum relative variance. The following parameters are defined to introduce the solved model: parameter  $c_z$ ,  $z \in S$ ,  $z \in Z$  indicates whether quarter  $z$  covers sample point  $s$  or not;  $n_z$ ,  $z \in Z$  indicates the number of sample points considered in quarter  $z$ ;  $\sigma_z^2$ ,  $z \in Z$  is the variance of quarter  $z$  with respect to the soil property analyzed;  $\sigma^2$  is the total variance of field with respect to soil property analyzed; and  $LS$  is the

---

V.M. Albornoz • L.J. Ñanco (✉)  
Departamento de Industrias, Universidad Técnica Federico Santa María, Av. Santa María 6400,  
Santiago, Chile  
e-mail: victor.albornoz@usm.cl; linco.nanco@alumnos.usm.cl

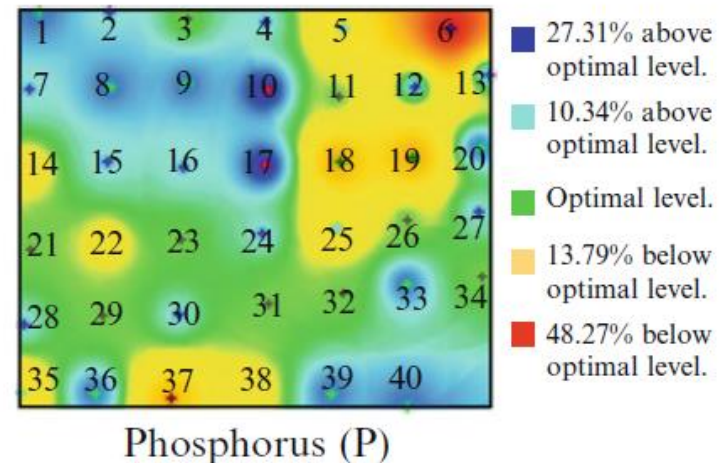
© Springer International Publishing Switzerland 2016  
R.J. Fonsco et al. (eds.), *Computational Management Science, Lecture Notes*  
in Economics and Mathematical Systems 682,  
DOI 10.1007/978-3-319-20430-7\_26

201

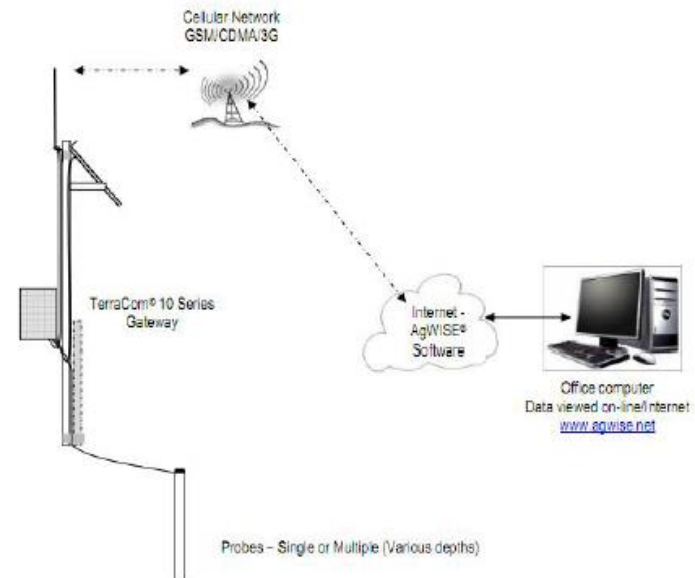


La variabilidad espacial de las propiedades del suelo representa uno de los aspectos más importantes que determina la productividad y calidad de la cosecha obtenida.

Subdividir un terreno en zonas de manejo sitio-específicas permite enfrentar esta variabilidad y diseñar planes de manejo apropiados.



A lo anterior contribuye el uso de nuevas herramientas de TI, pues las muestras de antecedentes del suelo se consiguen con sensores y con la ayuda de un software (SMS Mobile) y un sistema GPS pueden ser recolectadas empleando un sistema GIS.



En Cid-García et al. (2013), se propuso una nueva estrategia que provee zonas de manejo rectangulares, usando el concepto de varianza relativa como medida de homogeneidad (Ortega y Santibañez, 2007).

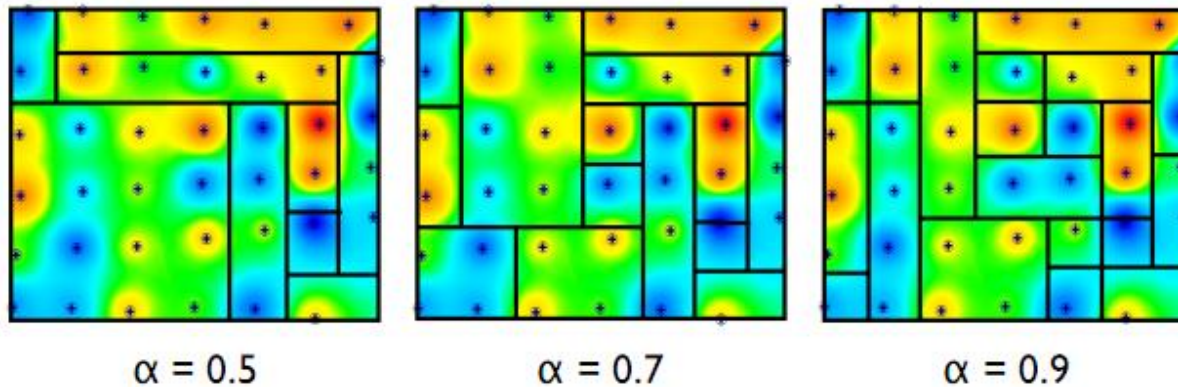
Homogeneous.



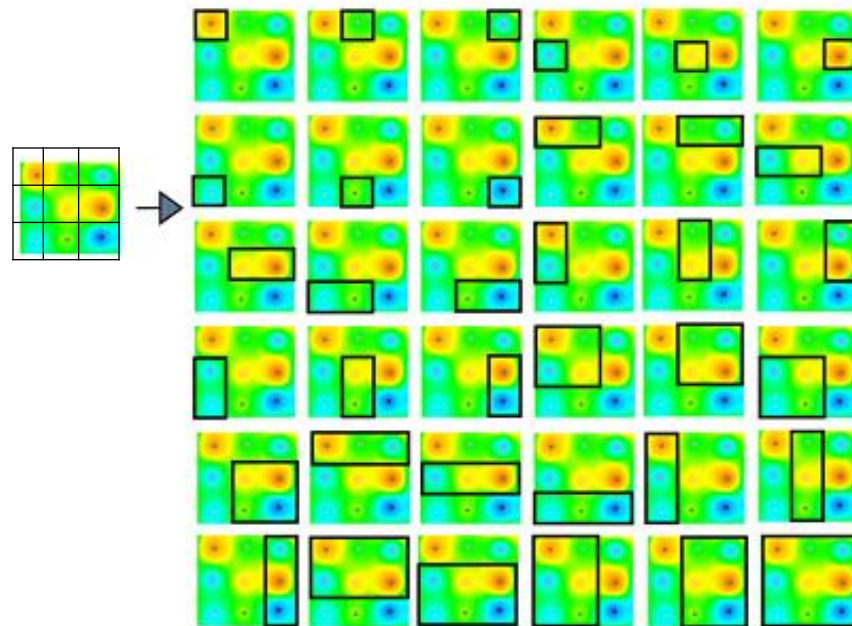
Heterogeneous.



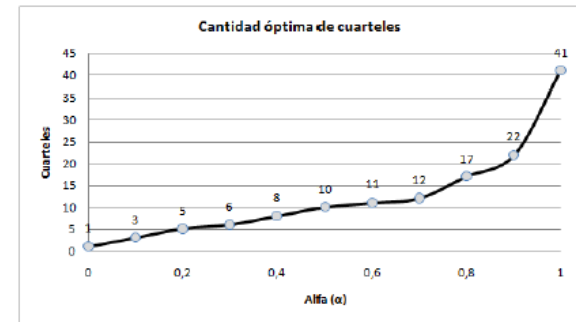
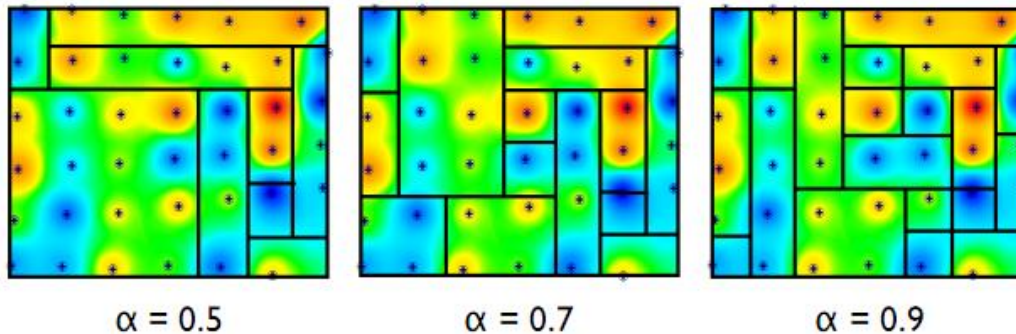
La solución propuesta permite subdividir el terreno en zonas rectangulares lo más homogéneas posible en base a una propiedad específica del suelo.



Las zonas propuestas son elegidas de entre un gran conjunto de potenciales zonas de manejo (rectangulares), a partir de un cierto tamaño mínimo y máximo según la cantidad de datos disponibles (puntos de muestra).



La solución propuesta permite subdividir el terreno en zonas rectangulares lo más homogéneas posible en base a una propiedad específica del suelo y una medida de homogeneidad ( $0 \leq \alpha \leq 1$ ).



## Notación:

$Z$  = conjunto de los potenciales zonas de manejo

$S$  = conjunto de las muestras del terreno de un índice dado

$\sigma_z^2$  = varianza de potencial zona de manejo  $z \in Z$

$n_z$  = cantidad de puntos de muestra en zona de manejo  $z \in Z$

$LS$  = límite superior de la cantidad de zonas a seleccionar

$c_{sz} = 1$  si la potencial zona de manejo  $z \in Z$  incluye el punto de muestra  $s \in S$  y 0 sino

$\sigma_T^2$  = varianza del terreno

$\alpha$  = varianza relativa impuesta a la partición

## Variables de decisión.

$q_z = 1$  si la potencial zona  $z \in Z$  es parte de la partición y  
0 sino,

## Modelo

$$\text{Min } \sum_{z \in Z} q_z \quad (1)$$

$$\text{s.a. } \sum_{z \in Z} c_{sz} q_z = 1, \quad s \in S \quad (2)$$

$$\sum_{z \in Z} q_z \leq LS \quad (3)$$

$$1 - \frac{\sum_{z \in Z} (n_z - 1) \sigma_z^2 q_z}{\sigma_T^2 (N - \sum_{z \in Z} q_z)} \geq \alpha \quad (4)$$

$$q_z \in \{0,1\} \text{ for all } z \in Z$$



# PROBLEMA MAESTRO

(relajación lineal modelo (1)-(4)) :

$$\text{Min } \sum_{z \in Z} q_z \quad (5)$$

s.a.

$$\sum_{z \in Z} c_{sz} q_z = 1, \quad s \in S \quad (6)$$

$$\sum_{z \in Z} q_z \leq LS \quad (7)$$

$$\sum_{z \in Z} [(n_z - 1)\sigma_z^2 + (1 - \alpha)\sigma_T^2] q_z \leq (1 - \alpha)\sigma_T^2 N \quad (8)$$

$$0 \leq q_z \leq 1, \quad z \in Z \quad (9)$$

# Subproblema

## Parámetros.

$\gamma_s$  = variable dual de la restricción (6) para cada  $s \in S$ .

$\omega$  = variable dual de la restricción (7)

$\pi$  = variable dual de la restricción (8)

## Variable de decisión.

$x_s = 1$  si punto muestral  $s$  es parte de la zona de manejo propuesta y 0 en caso contrario.

## SP Model.

$$\text{Min } 1 - \left( \sum_{s \in S} \gamma_s x_s + \omega + \pi \left[ \left( \sum_{s \in S} x_s - 1 \right) \sigma_z^2(x_s) + (1 - \alpha) \sigma_T^2 \right] \right)$$

$$\text{s.a. } x_s \in X$$

En Albornoz y Ñanco (2016) se discute la elección de la mejor estrategia de resolución del Subproblema que evitaba la resolución a optimalidad del mismo y explorando las mejoras a que daba lugar la incorporación de una o más soluciones con costos reducidos negativos.

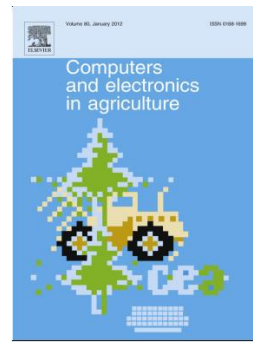
A continuación se muestra los resultados alcanzados en 10 instancias, empleando la mejor estrategia de resolución del Subproblema.

La estrategia adoptada fue implementada en AMPL y resuelta con Cplex 12.4 en un Intel core i5 de 2.5 GHz con 8 GB de RAM.

Instance	N	K	Time[s]	Optimal Value
1	42	588	0.48	10
2	100	3,025	11.52	22
3	150	6,600	106.14	24
4	225	14,400	1,468.62	34
5	300	25,200	8,761.71	47
6	400	44,100	47,654.8	58
7	500	68,250	-	-
8	600	97,650	-	-
9	750	151,125	-	-
10	900	216,225	-	-

Para la mejor estrategia adoptada en la implementación del método de Generación de Columnas se obtuvo los siguientes resultados ( $\alpha=0.5$ ):

Instance	#col	UB	LB	% gap	IS	ISLB	time [s]
1	188	8.89	8.89	4.50E-14	10	9	1.01
2	636	21.08	20.86	1.05	22	21	5.31
3	981	23.77	23.77	9.71E-14	24	24	17.17
4	1,700	33.23	33.23	1.87E-13	34	34	66.23
5	2,111	46.5	45.71	1.69	47	46	146.47
6	3,229	57.52	57.21	0.53	58	58	429.56
7	3,852	76.51	76.51	1.83E-13	77	77	743.093
8	5,160	84.56	84.56	2.93E-13	85	85	1,484.66
9	7,061	106.59	105.21	1.3	107	106	3,395.15
10	8,794	124.33	124.33	6.00E-13	125	125	6,061.11



## Referencias.

Cid-García, N., Albornoz, V., Ortega, R. and Ríos-Solís, Y. (2013). “*Rectangular shape management zone delineation using integer linear programming*”. *Computers & Electronics in Agriculture* 93, 1-9.

Albornoz, V.M., Ñanco, L.J. (2016). An empirical design of a column generation algorithm applied to a management zone delineation problem. In *Computational Management Science. State of the Art 2014*. Fonseca, R.J., Weber, G.W. and Telhada, J. (Eds). *Lecture Notes in Economics and Mathematical Systems* 682, 201-208, Springer.

Albornoz, V.M., Ñanco, L.J., Sáez, J.L. “*Delineating robust rectangular management zones based on column generation algorithm*”. *Computers and Electronics in Agriculture* 161, 194–201, 2019.

Albornoz, V.M., Zamora, G.E. “*Decomposition-based heuristic for the zoning and crop planning problem with adjacency constraints*”. *TOP* 29, 248–265, 2021.

Son numerosos los métodos que extienden CG así como las aplicaciones del mismo, entre otros:

VRPTW, Desrosiers et al. (1984).

**Routing with Time Windows by Column Generation\***

Jacques Desrosiers, François Soumis, and Martin Desrochers  
Centre de Recherche sur les Transports, Université de Montréal,  
Montréal, Québec, Canada H3C 3J7

Gerad, Ecole des Hautes Etudes Commerciales, Montréal, Québec, Canada H3T 1V6

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

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

Consider a set of trips where each trip is specified a priori by a place of origin, a destination, a duration, a cost, and a time interval within which the trip must begin. The trips may include visits to one or more specific points. Our problem is to determine the number of vehicles required, together with their routes and schedules, so that each trip begins within its given time interval, while the fixed costs related to the number of vehicles, and the travel costs between trips, are minimized. The problem is a generalization of the  $m$ -traveling salesman problem. We use column generation on a set partitioning problem solved by simplex and branch-and-bound; columns are generated by a shortest path algorithm with time windows on the nodes. Numerical results for several school bus transportation problems with up to 151 trips are discussed.

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

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

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

A Branch-and-Price algorithm for the Kidney Exchange Problem, Riascos-Álvarez et al. (2020)