DOCUMENTOS DE TRABAJO

BILTOKI

D.T. 2013.02

Generating cluster submodels from a multistage stochastic mixed integer optimzation model using break stage

Unai Aldasoro, María Araceli Garín, María Merino and Gloria Pérez

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Facultad de Ciencias Económicas.
Avda. Lehendakari Aguirre, 83
48015 BILBAO.

Documento de Trabajo BILTOKI DT2013.02
Editado por el Departamento de Economía Aplicada III (Econometría y Estadística)
de la Universidad del País Vasco.

ISSN: 1134-8984

Generating cluster submodels from a multistage stochastic mixed integer optimization model using break stage

Unai Aldasoro[1], María Araceli Garín[2], María Merino[3] and Gloria Pérez[3]

[1]Dpto. Matemática Aplicada,
Universidad del País Vasco, UPV/EHU. Bilbao (Bizkaia), Spain
e-mail: unai.aldasoro@ehu.es

[2]Dpto. de Economía Aplicada III,
Universidad del País Vasco, UPV/EHU. Bilbao (Bizkaia), Spain
e-mail: mariaaraceli.garin@ehu.es

[3]Dpto. de Matemática Aplicada, Estadística e Investigación Operativa,
Universidad del País Vasco, UPV/EHU. Leioa (Bizkaia), Spain
e-mail: maria.merino@ehu.es, gloria.perez@ehu.es

Abstract

We present a scheme to generate clusters submodels with stage ordering from a (symmetric or a nonsymmetric one) multistage stochastic mixed integer optimization model using break stage. We consider a stochastic model in compact representation and MPS format with a known scenario tree. The cluster submodels are built by storing first the 0-1 the variables, stage by stage, and then the continuous ones, also stage by stage. A C++ experimental code has been implemented for reordering the stochastic model as well as the cluster decomposition after the relaxation of the non-anticipativiy constraints until the so-called breakstage. The computational experience shows better performance of the stage ordering in terms of elapsed time in a randomly generated testbed of multistage stochastic mixed integer problems.

Keywords: Stochastic Optimization, Scenario Cluster Partitioning, Break Stage, C++, MPS.

*This research has been partially supported by the projects MTM2012-31514 from the Spanish Ministry of Economy and Competitiveness, Grupo de Investigación IT-567-13 of the Basque Government, UFI BETS 2011 of the University of Basque Country (UPV/EHU), Spain, and Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED 2011). The computational resources were provided by SGI/IZO-SGIker at UPV/EHU (supported by the Spanish Ministry of Education and Science and the European Social Fund).

1 Introduction

Let us consider the following multistage deterministic mixed 0-1 model

$$\begin{aligned} \min \sum_{t\in\mathscr{T}} & a_t x_t + c_t y_t \\ s.t.\ & A_1 x_1 + B_1 y_1 = b_1 \\ & A'_t x_{t-1} + A_t x_t + B'_t y_{t-1} + B_t y_t = b_t \quad \forall t \in \mathscr{T} - \{1\} \\ & x_t \in \{0,1\}^{nx_t},\ \ y_t \in \mathbb{R}^{+ny_t}, \end{aligned} \tag{1}$$

where $\mathscr{T}$ is the set of stages, such that $T = |\mathscr{T}|$, x_t and y_t are the nx_t and ny_t dimensional vectors of the 0-1 and continuous variables, respectively, a_t and c_t are the vectors of the objective function coefficients, A'_t, A_t, B'_t and B_t are the constraint matrices and b_t is the right-hand-side vector (*rhs*) for stage t.

This model can be extended to consider uncertainty in some of the main parameters, in our case, the objective function, *rhs* and the constraint matrix coefficients. To introduce the uncertainty in the parameters, we use a scenario analysis approach.

Definition 1 *A **scenario** consists of a realization of all the random parameters in all stages, that is, a path through the scenario tree.*

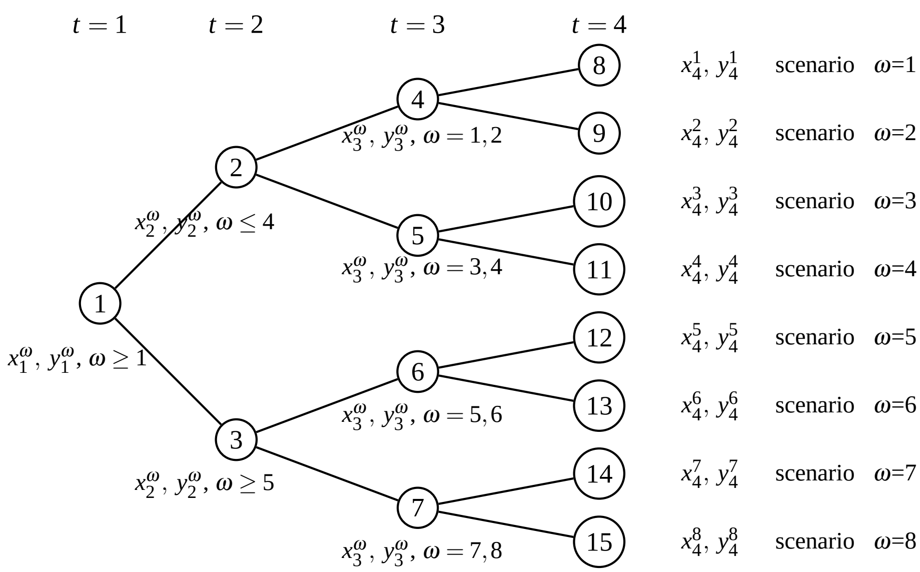

Figure 1: Symmetric scenario tree. Illustrative example.

So, Ω will denote the set of scenarios, $\omega \in \Omega$ will represent a specific scenario, see Figure 1 and w^ω will denote the likelihood or probability assigned by the modeler to scenario ω, such that $\sum_{\omega\in\Omega} w^\omega = 1$. We say that two scenarios belong to the same group in a given stage provided that they have the same realizations of the uncertain parameters up to the stage. Following the *nonanticipativity principle* stated in [Wets, 1974] and restated in [Rockafellar and Wets, 1991], see also [Birge and Louveaux, 2011], among many others, both scenarios should have the same value for the related variables with the time index up to the given stage.

Let also $\mathscr{G}$ denote the set of scenario groups (i.e., nodes in the underlying scenario tree), and $\mathscr{G}_t$ denote the subset of scenario groups that belong to stage $t \in \mathscr{T}$, such that $\mathscr{G} = \cup_{t\in\mathscr{T}}\mathscr{G}_t$. Ω_g denotes the set of scenarios for group g, for $g \in \mathscr{G}$. Note that the scenario group concept corresponds to the node concept in the underlying scenario tree. Note that we will consider the order of scenario groups by stages.

If we consider the *splitting variable* representation of the DEM of the full recourse stochastic version related to the multistage deterministic problem (1) can be expressed as follows,

$$\begin{aligned} z_{DEM} = \min & \sum_{\omega\in\Omega}\sum_{t\in\mathscr{T}} w^{\omega}\left(a_t^{\omega}x_t^{\omega} + c_t^{\omega}y_t^{\omega}\right) \\ s.t.\ & A_1x_1^{\omega} + B_1y_1^{\omega} = b_1 \ \ \forall\omega\in\Omega \\ & A_t'^{\omega}x_{t-1}^{\omega} + A_t^{\omega}x_t^{\omega} + B_t'^{\omega}y_{t-1}^{\omega} + B_t^{\omega}y_t^{\omega} = b_t^{\omega}\ \forall\omega\in\Omega,\ t\geq 2 \\ & x_t^{\omega} - x_t^{\omega'} = 0\ \forall\omega,\omega'\in\Omega_g : \omega\neq\omega',\ g\in\mathscr{G}_t,\ t\leq T-1 \\ & y_t^{\omega} - y_t^{\omega'} = 0\ \forall\omega,\omega'\in\Omega_g : \omega\neq\omega',\ g\in\mathscr{G}_t,\ t\leq T-1 \\ & x_t^{\omega}\in\{0,1\}^{nx_t^{\omega}}\ \ y_t^{\omega}\in\mathbb{R}^{+ny_t^{\omega}},\ \forall\omega\in\Omega,\ t\in\mathscr{T}. \end{aligned} \tag{2}$$

Following the nonanticipativity principle cited above, the corresponding equalities must be satisfied for stage t,

$$\begin{aligned} A_t^{'\omega} = A_t^{'\omega'},\ A_t^{\omega} = A_t^{\omega'},\ B_t^{'\omega} = B_t^{'\omega'},\ B_t^{\omega} = B_t^{\omega'},\ b_t^{\omega} = b_t^{\omega'}, a_t^{\omega} = a_t^{\omega'}, c_t^{\omega} = c_t^{\omega'}, \\ \forall\omega,\omega'\in\Omega_g : \omega\neq\omega',\ g\in\mathscr{G}_t,\ 2\leq t\leq T-1. \end{aligned} \tag{3}$$

Observe that for a given stage t, $A_t^{'\omega}$ and A_t^{ω} are the technology and recourse matrices for the x_t variables and $B_t^{'\omega}$ and B_t^{ω} are the corresponding ones for the y_t variables. Notice that $x_t^{\omega} - x_t^{\omega'} = 0$ and $y_t^{\omega} - y_t^{\omega'} = 0$ are the NAC. Finally, nx_t^{ω} and ny_t^{ω} denote the dimensions of the vectors of the x and y variables, respectively, related to stage t under scenario ω.

And the compact representation of the previous model is as follows,

$$\begin{aligned} z_{DEM} = \min \quad & \sum_{\omega\in\Omega} w^{\omega} \sum_{g\in\mathscr{N}^{\omega}} (a^g x^g + c^g y^g) & \\ \text{s.t.} \quad & A'^g x^{\sigma(g)} + A^g x^g + B'^g y^{\sigma(g)} + B^g y^g = h^g & \forall g\in\mathscr{G} \\ & x^g\in\{0,1\}^{nx^g}, y^g\in\mathbb{R}^{+ny^g} & \forall g\in\mathscr{G}, \end{aligned} \tag{4}$$

where N^{ω} is the set of ancestor groups of scenario ω (including itself) in the scenario tree that is used for representing the random and decision variables, for $\omega\in\Omega$. Additionally, $\sigma(g)$ is the scenario group related to the immediate ancestor group of group g, such that $\sigma(g)\in\mathscr{G}_{t(g)-1}$, for $g\in\mathscr{G}-\mathscr{G}_1$, where $t(g)$ is the stage from set $\mathscr{T}$ to which group g belongs to, such that $g\in\mathscr{G}_{t(g)}$. Additionally, x^g and y^g represent the replicas of the x and y variables for scenario group g, respectively, a^g and c^g are the related objective function vector coefficients for the 0-1 and continuous variables, respectively, A'^g, A^g, B'^g and B^g are the constraint matrices, and h^g is the right-hand-side vector (rhs) for scenario group g, where $g\in\mathscr{G}$.

The scenario tree information given in Figure 1 can also be represented and managed by using the vector $\mathscr{R}$ given in the following definition.

Definition 2 *A general* ***scenario tree*** *compact notation can be uniquely defined by* $\mathscr{R} = (r(g)\ :\ g\in\cup_{t=1}^{T-1}\mathscr{G}_t)$*, where* $r(g)\in\mathbb{N}$ *is the number of branches arising from the stage* $t(g)$ *of group* g*, to the next stage* $t(g)+1$*. That is,*

$$\mathscr{R} = (\overbrace{r_{1|\mathscr{G}_1|}}^{t=1}\ |\ \overbrace{r_{21},\ r_{22},\ \ldots,\ r_{2|\mathscr{G}_2|}}^{t=2}\ |\ \overbrace{r_{31},\ r_{32},\ \ldots,\ r_{3|\mathscr{G}_3|}}^{t=3}\ |\ldots|\ \overbrace{r_{T-1,1},\ r_{T-1,2},\ \ldots,\ r_{T-1,|\mathscr{G}_{T-1}|}}^{t=T-1}),$$

where the number of groups for stage t, $|\mathscr{G}_t|$, corresponds to the sum of branches of the previous stage:

$$|\mathscr{G}_1| = 1,\ |\mathscr{G}_{t+1}| = \sum_{i=1}^{|\mathscr{G}_t|} r_{ti},\ t \leq T-1$$

For the example given in section 1.2 of the book [Birge and Louveaux, 2011], the scenario tree shown in Figure 1 can be defined by: $\mathscr{R} = (2 \mid 2\ 2 \mid 2\ 2\ 2\ 2)$. The set of scenarios is $\Omega = \{1,2,\ldots,8\}$, and the subsets of scenario groups are $\mathscr{G}_1 = \{1\},\ \mathscr{G}_2 = \{2,\ 3\},\ \mathscr{G}_3 = \{4,5,6,7\},\ \mathscr{G}_4 = \{8,9,\ldots,15\}$ and $\mathscr{G} = \cup_{t=1}^{4}\mathscr{G}_t$. Finally, the scenarios in each group g, are: $\Omega_1 = \{1,\ldots,8\}$, $\Omega_2 = \{1,2,3,4\}$, $\Omega_3 = \{5,6,7,8\}$, $\Omega_4 = \{1,2\}$, $\Omega_5 = \{3,4\}$, $\Omega_6 = \{5,6\}$, $\Omega_7 = \{7,8\}$, $\Omega_8 = \{1\}$, $\Omega_9 = \{2\}$, $\ldots$, $\Omega_{15} = \{8\}$.

Definition 3 *A **symmetric tree** is a tree where the number of branches is the same for all conditional distributions in the same stage, that is, the number of branches arising from any scenario group at each stage t to the next one is the same for all groups in $\mathscr{G}_t$,$r_{ti} = r_{tj},\ \forall i \neq j,\ 1 \leq i,j \leq |\mathscr{G}_t|,\ t \leq T-1$.*

In general, for any multi-stage stochastic problem with T stages and $|\Omega|$ scenarios, the information about until what stage the scenario submodels have common information, and when the NAC must be explicit, is saved in the subsets $\mathscr{G}_t$ and Ω_g, $g \in \mathscr{G}_t$, $t \in T$, i.e., in the scenario tree $\mathscr{R}$ or, alternatively, in the *scenario tree matrix*, defined below.

Definition 4 *The **scenario tree matrix**, $ST \in \mathscr{M}_{|\Omega|\times|\mathscr{G}|}$, is a matrix where the corresponding value for the pair (ω,g) gives the related stage t, such that*

$$ST(\omega,g) = \begin{cases} t, & \text{if } \omega \in \Omega_g \text{ and } g \in \mathscr{G}_t \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Notice that the scenario tree matrix reproduces the structure given by the scenario tree $\mathscr{R}$. This matrix has been built by using the sets Ω_g and $\mathscr{G}_t$, i.e., the scenario tree $\mathscr{R}$, but these sets can be also generated from the matrix. For each stage $t \in \mathscr{T}$, we can obtain the set of scenario groups in such stage, $\mathscr{G}_t$, as the column of the position (ω,g), for which the corresponding element in the scenario tree matrix is equal to t; then $\mathscr{G}_t = \{g \in \mathscr{G} \mid \exists\omega \in \Omega :\ ST(\omega,g) = t\}$. See also that the set of scenarios related to group g is $\Omega_g = \{\omega \in \Omega \mid ST(\omega,g) \neq 0\}$. For our example, the scenario tree matrix, $ST(\omega,g)$, is given in (6).

$$ST(\omega,g) = \left(\begin{array}{c|cc|cccc|cccccccc}
1 & 2 & 0 & 3 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\
1 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4
\end{array}\right). \tag{6}$$

We will decompose the scenario tree into a subset of scenario clusters subtrees, each one for a scenario cluster in the set denoted as $\mathscr{C} = \{1,...,C\}$ with $C = |\mathscr{C}|$, see below the reason for it. Let Ω^c denote the set of scenarios that belongs to cluster c, where $c \in \mathscr{C}$ and $\sum_{c=1}^{C} |\Omega^c| = |\Omega|$. It is clear that the criterion for scenario clustering is instance dependent. In any case, notice that $\Omega^c \bigcap \Omega^{c'} = \emptyset$, $c, c' = 1, \ldots, C : c \neq c'$ and $\Omega = \cup_{c=1}^{C} \Omega^c$. Let also $\mathscr{G}^c \subset \mathscr{G}$ denote the set of scenario groups for cluster c, such that $\Omega_g \cap \Omega^c \neq \emptyset$ means that $g \in \mathscr{G}^c$, and let $\mathscr{G}_t^c = \mathscr{G}_t \cap \mathscr{G}^c$ denote the set of scenario groups for cluster $c \in \mathscr{C}$ in stage $t \in \mathscr{T}$.

We propose to choose the number of scenario clusters C as any value from the subset $\mathscr{Q} = \{|\mathscr{G}_1|, |\mathscr{G}_2|, \ldots, |\mathscr{G}_T|\}$. As we will see below, the parameter C will be associated with the number of stages with explicit NAC between scenario clusters.

2 Illustrative stochastic example in MPS format

Let us consider the illustrative example of financial planning and control given in section 1.2 of the book [Birge and Louveaux, 2011]. As it is explained in the book, there are 55 thousand dollars to invest in any of $\mathscr{I} = \{1,2\}$ investments. After $T-1 = 3$ investment periods, we will have a wealth that we would like to have a exceed a tuition goal of 80 thousand dollars. We suppose that exceeding the goal would be equivalent to our having an income of 1% of the excess while not meeting the goal would lead to borrowing for a cost 4% of the amount short. The major uncertainty in this model is the return on each investment i within each period t. The decisions of investments are the y_{ti} variables, where $i \in \mathscr{I}$ and $t \leq T-1$, the deficit or shortage is denoted y_{T1} and the excess or surplus variable is y_{T2}, see Figure 2.

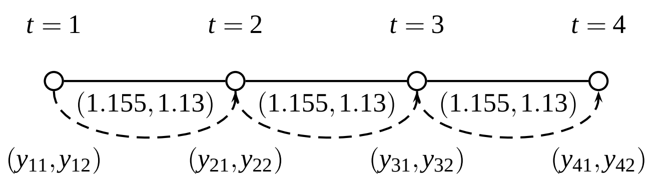

Figure 2: Deterministic problem

If we consider the expected returns (1.155, 1.13), we can formulate the deterministic model as follows:

$$
\begin{array}{rrl}
\max z = & y_{41} - 4y_{42} & \\
\text{s.t.} & y_{11} + y_{21} & = 55 \\
 & -1.155y_{11} - 1.13y_{12} & +y_{21} + y_{22} = 0 \\
 & -1.155y_{21} - 1.13y_{22} & +y_{31} + y_{32} = 0 \\
 & 1.155y_{31} + 1.13y_{32} & -y_{41} + y_{42} = 80 \\
 & y_{ti} \geq 0,\ \forall i \in \mathscr{I},\ t \leq T-1, & y_{41}, y_{42} \geq 0
\end{array}
\tag{7}
$$

To explain this technique we consider a full model with a symmetric scenario tree of 2 branches in each stage, being $|\mathscr{T}| = 4$ the number of stages, that is, with $|\Omega| = 8$ scenarios. The multistage stochastic problem can be formulated in compact representation as follows:

$$
\begin{array}{rll}
\max z = & \sum_{\omega=1}^{8} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^{1} + y_{12}^{1} = 55 & \\
& -1.25y_{11}^{1} - 1.14y_{12}^{1} & +y_{21}^{1} + y_{22}^{1} = 0 \\
& -1.06y_{11}^{1} - 1.12y_{12}^{1} & +y_{21}^{5} + y_{22}^{5} = 0 \\
& -1.25y_{21}^{1} - 1.14y_{22}^{1} & +y_{31}^{1} + y_{32}^{1} = 0 \\
& -1.06y_{21}^{1} - 1.12y_{22}^{1} & +y_{31}^{3} + y_{32}^{3} = 0 \\
& -1.25y_{21}^{5} - 1.14y_{22}^{5} & +y_{31}^{5} + y_{32}^{5} = 0 \\
& -1.06y_{21}^{5} - 1.12y_{22}^{5} & +y_{31}^{7} + y_{32}^{7} = 0 \\
& 1.25y_{31}^{1} + 1.14y_{32}^{1} & -y_{41}^{1} + y_{42}^{1} = 80 \\
& 1.06y_{31}^{1} + 1.12y_{32}^{1} & -y_{41}^{2} + y_{42}^{2} = 80 \\
& 1.25y_{31}^{3} + 1.14y_{32}^{3} & -y_{41}^{3} + y_{42}^{3} = 80 \\
& 1.06y_{31}^{3} + 1.12y_{32}^{3} & -y_{41}^{4} + y_{42}^{4} = 80 \\
& 1.25y_{31}^{5} + 1.14y_{32}^{5} & -y_{41}^{5} + y_{42}^{5} = 80 \\
& 1.06y_{31}^{5} + 1.12y_{32}^{5} & -y_{41}^{6} + y_{42}^{6} = 80 \\
& 1.25y_{31}^{7} + 1.14y_{32}^{7} & -y_{41}^{7} + y_{42}^{7} = 80 \\
& 1.06y_{31}^{7} + 1.12y_{32}^{7} & -y_{41}^{8} + y_{42}^{8} = 80 \\
& y_{ti}^{\omega} \geq 0, & \forall i = 1,2, t \in \mathscr{T}, \omega \in \Omega
\end{array}
\tag{8}
$$

This problem can be represented with MPS format as follows:

```
NAME          TOTAL
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
 E  R0000004
 E  R0000005
 E  R0000006
 E  R0000007
 E  R0000008
 E  R0000009
 E  R0000010
 E  R0000011
 E  R0000012
 E  R0000013
 E  R0000014
COLUMNS
    C0000000  OBJROW    0.125          R0000007  -1.
    C0000001  OBJROW    0.125          R0000008  -1.
    C0000002  OBJROW    0.125          R0000009  -1.
    C0000003  OBJROW    0.125          R0000010  -1.
    C0000004  OBJROW    0.125          R0000011  -1.
    C0000005  OBJROW    0.125          R0000012  -1.
    C0000006  OBJROW    0.125          R0000013  -1.
    C0000007  OBJROW    0.125          R0000014  -1.
    C0000008  OBJROW    -0.5           R0000007  1.
    C0000009  OBJROW    -0.5           R0000008  1.
    C0000010  OBJROW    -0.5           R0000009  1.
    C0000011  OBJROW    -0.5           R0000010  1.
    C0000012  OBJROW    -0.5           R0000011  1.
    C0000013  OBJROW    -0.5           R0000012  1.
    C0000014  OBJROW    -0.5           R0000013  1.
    C0000015  OBJROW    -0.5           R0000014  1.
    C0000016  R0000000  1.             R0000001  -1.25
    C0000016  R0000002  -1.06
    C0000017  R0000001  1.             R0000003  -1.25
    C0000017  R0000004  -1.06
    C0000018  R0000002  1.             R0000005  -1.25
    C0000018  R0000006  -1.06
    C0000019  R0000003  1.             R0000007  1.25
    C0000019  R0000008  1.06
    C0000020  R0000004  1.             R0000009  1.25
    C0000020  R0000010  1.06
    C0000021  R0000005  1.             R0000011  1.25
    C0000021  R0000012  1.06
    C0000022  R0000006  1.             R0000013  1.25
    C0000022  R0000014  1.06
    C0000023  R0000000  1.             R0000001  -1.14
    C0000023  R0000002  -1.12
    C0000024  R0000001  1.             R0000003  -1.14
    C0000024  R0000004  -1.12
    C0000025  R0000002  1.             R0000005  -1.14
    C0000025  R0000006  -1.12
    C0000026  R0000003  1.             R0000007  1.14
    C0000026  R0000008  1.12
    C0000027  R0000004  1.             R0000009  1.14
    C0000027  R0000010  1.12
    C0000028  R0000005  1.             R0000011  1.14
    C0000028  R0000012  1.12
    C0000029  R0000006  1.             R0000013  1.14
    C0000029  R0000014  1.12
RHS
    RHS       R0000000  55.
    RHS       R0000001  0.             R0000002   0.
    RHS       R0000003  0.             R0000004   0.
    RHS       R0000005  0.             R0000006   0.
    RHS       R0000007  80.            R0000008   80.
    RHS       R0000009  80.            R0000010   80.
    RHS       R0000011  80.            R0000012   80.
    RHS       R0000013  80.            R0000014   80.
ENDATA
```

Total.mps (8)

3 Basic requeriments

For the aim of obtaining the scenario clustering partition the following two files are needed:

- `Total.mps`, a (symmetric or not) multistage stochastic mixed integer optimization model in compact representation without cross-scenario constraints in MPS format and
- `inputData.dat`, a input file with the following information:
  1. t^*, break stage
  2. T, number of stages
  3. $(\mathscr{G}_t)_{t\in\mathscr{T}}$, number of scenario groups for each stage $t \in \mathscr{T}$
  4. $\mathscr{R}$, number of branches along the scenario tree (symmetric or not) ordered by scenario group
  5. $(nx_t)_{t\in\mathscr{T}}$, number of 0-1 variables by stage (number of variables in any scenario group)
  6. $(ny_t)_{t\in\mathscr{T}}$, number of continuous variables by stage (number of variables in any scenario group)
  7. $(w^\omega)_{w\in\Omega}$, the vector of likelihood for scenarios; if all scenarios have the same probability of occurrence, 0 value appears in the corresponding line.
  8. $o(x,y)$, order of variables. Firstly 0-1 and then continuous; in each variable type ordered by stage; in each stage ordered by scenario group. If the original ordering in the MPS file is the expected one, 0 value appears.

```
1
4
1  2  4  8
2     2  2     2  2  2  2
0  0  0  0
2  2  2  2
0
16  23
17  24     18  25
19  26     20  27     21  28     22  29
 0   8      1   9      2  10      3  11      4  12     5  13      6  14      7  15
```

inputData.dat

The first data is the initial decision number of break stage, $t^* = 1$ and consequently, the number of submodels $C = |\mathscr{G}_{t^*+1}| = 2$. The number of stages is $T = 4$ and there are $|\mathscr{G}_1| = 1$, $|\mathscr{G}_2| = 2$, $|\mathscr{G}_3| = 4$ and $|\mathscr{G}_4| = 8$ scenario groups. The tree associated with the figure 1 has 2 branches in each scenario group $g \in \mathscr{G}_1 \cup \mathscr{G}_2 \cup \ldots \mathscr{G}_{T-1}$. Notice that nx_t or ny_t is the same number of 0-1 or continuous variables for each scenario group of the same stage; for example, in stage $t = 3$ there are $(nx_3, ny_3) = (0,2)$ variables in each scenario group $g \in \mathscr{G}_3 = \{4,5,6,7\}$; so, in the third stage there are no 0-1 variables and 8 continuous variables. The likelihood for each scenario $\omega \in \Omega$ is $w^\omega = \frac{1}{|\Omega|} = 0.125$. Finally, the variables in the Total.mps file appear in the following order: $y^1_{41}, y^2_{41}, \ldots, y^8_{41}, y^1_{42}, y^2_{42}, \ldots, y^8_{42}, y^1_{11}, y^1_{21}, y^5_{21}, y^1_{31}, y^3_{31}, y^5_{31}, y^7_{31}, y^1_{12}, y^1_{22}, y^5_{22}, y^1_{32}, y^3_{32}, y^5_{32}, y^7_{32}$, while the wanted order in Output.mps file is y^1_{11}, y^1_{12} for the first stage (in blue), $y^1_{21}, y^1_{22}, y^5_{21}, y^5_{22}$ for the second stage (in green), $y^1_{31}, y^1_{32}, y^3_{31}, y^3_{32}, y^5_{31}, y^5_{32}, y^7_{31}, y^7_{32}$ for the third stage (in red) and $y^1_{41}, y^1_{42}, y^2_{41}, y^2_{42}, \ldots, y^8_{41}, y^8_{42}$ for the fourth stage (in black), see Total.mps, inputData.dat and Figure 3.

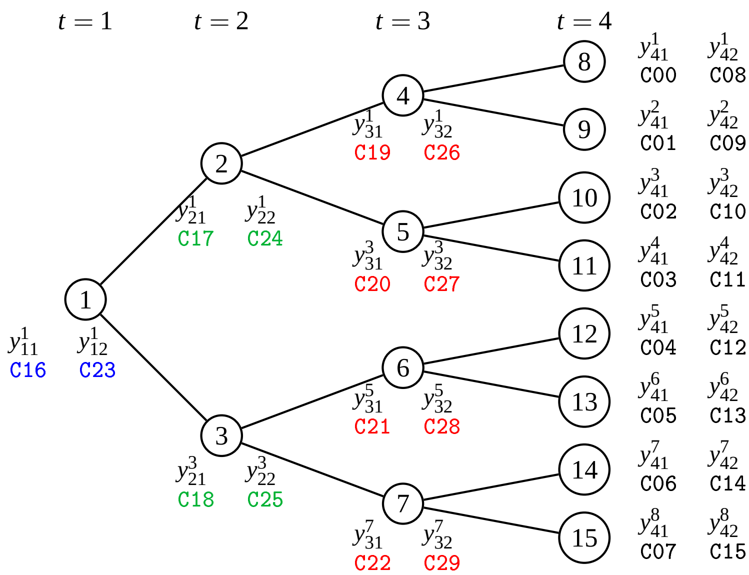

Figure 3: Variables stage ordering

4 Scenario Cluster Partitioning

The aim is to break this problem in scenario submodels according to the selected break stage and create the corresponding mps cluster models and full model (8) ordered by stages. About scenario cluster partitioning, see [Escudero *et al.*, 2010a; Escudero *et al.*, 2010b; Escudero *et al.*, 2012].

Definition 5 *A* ***break stage*** t^* *is a stage t such that the number of scenario clusters is* $C = |\mathscr{G}_{t^*+1}|$, *where* $t^*+1 \in \mathscr{T}$. *In this case, any cluster* $c \in \mathscr{C}$ *is induced by a group* $g \in \mathscr{G}_{t^*+1}$ *and contains all scenarios belonging to that group, i.e.,* $\Omega^c = \Omega_g$.

Definition 6 *The* scenario cluster *models are those that result from the relaxation of the NAC until some break stage* t^* *in model (2), called* t^****-decomposition***.

Recall that the choice of $t^* = 0$ corresponds to the full model and $t^* = T-1$ corresponds to the scenario partitioning.

Definition 7 *The* ***cluster tree matrix*** *associated with the* t^*-*decomposition,* $CT^{t*} \in M_{C\times|\mathscr{G}|}$, *is a matrix where the corresponding value for the pair* (p,g) *gives the related stage t, such that*

$$CT^{t^*}(p,g) = \begin{cases} t, & \text{if } g \in \mathscr{G}_t^p \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Notice that $\mathscr{G}_t^p = \mathscr{G}_t \cap \mathscr{G}^p$, is the set of scenario groups for cluster $c \in \mathscr{C}$ in stage $t \in \mathscr{T}$.

Once decided the break stage, t^*, the corresponding cluster partition is given, and its structure is defined by the related cluster tree matrix.

Notice that the subsets $\mathscr{G}^p$ and $\mathscr{G}_t$ and, consequently, $\mathscr{G}_t^p$ can be obtained from the cluster tree matrix given above. For each cluster $c \in \mathscr{C}$ (i.e. $c-$row in matrix CT^{t^*}), the set of scenario groups $\mathscr{G}^p$ can be obtained as the set of columns in the t^*-cluster tree matrix with a nonzero element, i.e., $\mathscr{G}^c = \{g \in \mathscr{G} \mid CT^{t^*}(c,g) \neq 0\}$. Similarly, the set $\mathscr{G}_t$ of scenario groups in each stage $t \in \mathscr{T}$ can be obtained as $\mathscr{G}_t = \{g \in \mathscr{G} \mid \exists c \in \mathscr{C} : CT^{t^*}(c,g) = t\}$.

In the illustrative example depicted in Figure 1, three cases can be considered for generating the C cluster submodels where C can be chosen from the set of values $\{|\mathscr{G}_2|,|\mathscr{G}_3|,|\mathscr{G}_4|\} = \{2,4,8\}$. We can consider the break stage $t^* = 1$ and then $|\mathscr{C}| = 2$ cluster submodels are obtained (the nonanticipativity constraints have been relaxed for the first stage), the break stage $t^* = 2$ (the NAC have been relaxed for the first and second stages) and then $|\mathscr{C}| = 4$ cluster submodels are obtained or the break stage $t^* = 3$ (all the NAC have been relaxed) and then $|\mathscr{C}| = 8$ cluster or scenario submodels are obtained, see Figure 4.

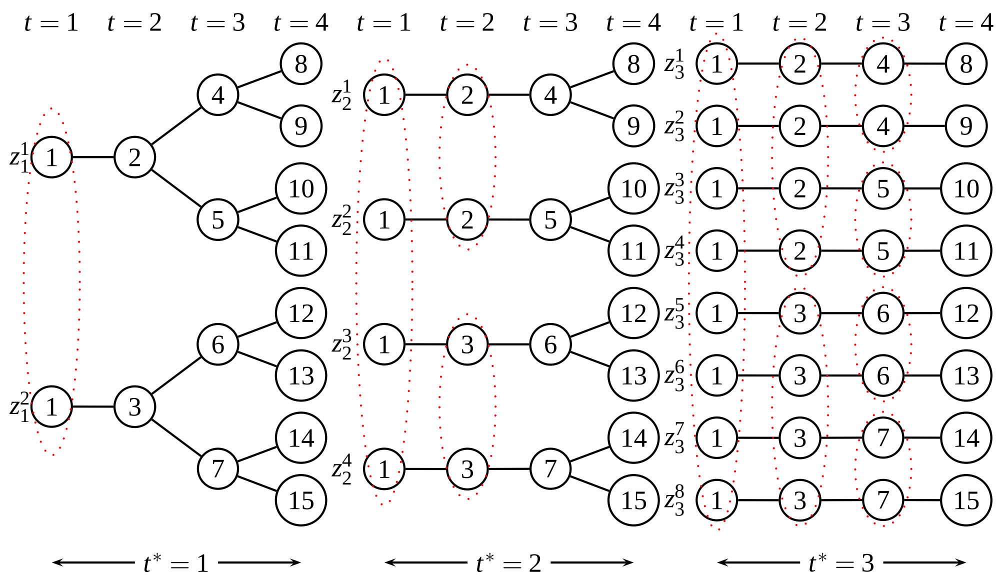

Figure 4: Scenario cluster partitioning, for $t^* = 1$ (left), $t^* = 2$ (central) and $t^* = 3$ (right)

The algorithm is detailed in file `mainmps.cpp`, see Appendix A. A scheme of the procedure is detailed in Algorithm 1:

Step 1: Input file: read full model **Total.mps** with original variable order
Input data: read t^*, T, $\mathscr{G}_t$, $\mathscr{R}$, n_{xt}, n_{yt}, w^ω $o(x,y)$.
Step 2: Calculate additional vectors.
Step 3: Reorder objective function coefficients a and b
Reorder columns of constraints matrices A',A,B', and B
Reorder bounds of the continuous variables x and y according to the order of variables vector.
Step 4: Generate the stage ordered full model **Output.mps**.
Step 5: Link full model variables with the corresponding cluster using the cluster tree matrix.
Step 6: Link rows to clusters. By default all are assigned.
For $i = 0$ to nelements **do**.
  For $j = 0$ to C cluster submodel **do**.
    If Column of element i does not belong to Cluster j **then**.
      Unlink row of element i from Cluster j.
Step 7: Renumber cluster rows.
Reorder the right-hand-side vector.
Renumber the element vector and update corresponding row index.
Step 8: Generate stage ordered **Clusterc.mps** files

Algorithm 1: mainmps.cpp scheme

- **Case 1**. Let the break stage $t^* = 1$, then there are $C = |\mathscr{G}_2| = 2$ clusters, see left decomposition in Figure 4 and, then, two subsets of scenario groups, say $\mathscr{G}^1 = \{1,2,4,5,8,9,10,11\}$ and $\mathscr{G}^2 = \{1,3,6,7,12,13,14,15\}$, where the scenarios in each set are $\Omega^1 = \{1,2,3,4\}$ and $\Omega^2 = \{5,6,7,8\}$.

  The 1-cluster tree matrix is given in (10).

$$CT^1(p,g) = \left(\begin{array}{c|cc|cccc|cccccccc} 1 & 2 & 0 & 3 & 3 & 0 & 0 & 4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 3 & 3 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 \end{array}\right). \tag{10}$$

  The $|\mathscr{C}| = 2$ cluster submodels obtained for the break stage $t^* = 1$, (11) and (12) are:

$$\begin{array}{lrl} \max z_1^1 & = \sum_{\omega=1}^4 \frac{1}{8}(y_{41}^\omega - 4y_{42}^\omega) & \\ \text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\ & -1.25y_{11}^1 - 1.14y_{12}^1 & +y_{21}^1 + y_{22}^1 = 0 \\ & -1.25y_{21}^1 - 1.14y_{22}^1 & +y_{31}^1 + y_{32}^1 = 0 \\ & -1.06y_{21}^1 - 1.12y_{22}^1 & +y_{31}^3 + y_{32}^3 = 0 \\ & 1.25y_{31}^1 + 1.14y_{32}^1 & -y_{41}^1 + y_{42}^1 = 80 \\ & 1.06y_{31}^1 + 1.12y_{32}^1 & -y_{41}^2 + y_{42}^2 = 80 \\ & 1.25y_{31}^3 + 1.14y_{32}^3 & -y_{41}^3 + y_{42}^3 = 80 \\ & 1.06y_{31}^3 + 1.12y_{32}^3 & -y_{41}^4 + y_{42}^4 = 80 \\ & y_{ti}^\omega \geq 0, \forall i = 1,2, t \in \mathscr{T}, & \omega \in \Omega_2 \end{array} \tag{11}$$

$$
\begin{array}{rll}
\max z_1^2 & = \sum_{\omega=5}^{8} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & \quad y_{11}^{1} + y_{12}^{1} = 55 & \\
& -1.06y_{11}^{1} - 1.12y_{12}^{1} & +y_{21}^{5} + y_{22}^{5} = 0 \\
& -1.25y_{21}^{5} - 1.14y_{22}^{5} & +y_{31}^{5} + y_{32}^{5} = 0 \\
& -1.06y_{21}^{5} - 1.12y_{22}^{5} & +y_{31}^{7} + y_{32}^{7} = 0 \\
& 1.25y_{31}^{5} + 1.14y_{32}^{5} & -y_{41}^{5} + y_{42}^{5} = 80 \\
& 1.06y_{31}^{5} + 1.12y_{32}^{5} & -y_{41}^{6} + y_{42}^{6} = 80 \\
& 1.25y_{31}^{7} + 1.14y_{32}^{7} & -y_{41}^{7} + y_{42}^{7} = 80 \\
& 1.06y_{31}^{7} + 1.12y_{32}^{7} & -y_{41}^{8} + y_{42}^{8} = 80 \\
& y_{ti}^{\omega} \geq 0, \forall i = 1,2, & t \in \mathscr{T}, \omega \in \Omega_3
\end{array} \tag{12}
$$

The corresponding submodels for break stage $t^* = 1$ in MPS format are in Appendix B.

- **Case 2**. Let the break stage $t^* = 2$, then there are $C = |\mathscr{G}_3| = 4$ clusters, see central decomposition in Figure 4 and, then, four subsets of scenario groups, say $\mathscr{G}^1 = \{1,2,4,8,9\}$, $\mathscr{G}^2 = \{1,2,5,10,11\}$, $\mathscr{G}^3 = \{1,3,6,12,13\}$, and $\mathscr{G}^4 = \{1,3,7,12,14,15\}$, where the scenarios in each set are $\Omega^1 = \{1,2\}$, $\Omega^2 = \{3,4\}$, $\Omega^3 = \{5,6\}$ and $\Omega^4 = \{7,8\}$.

  The 2-cluster tree matrix is given in (13).

$$
CT^2(p,g) = \left(\begin{array}{c|cc|cccc|cccccccc}
1 & 2 & 0 & 3 & 0 & 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4
\end{array} \right). \tag{13}
$$

  The $|\mathscr{C}| = 4$ cluster submodels obtained for the break stage $t^* = 2$, (14)-(17) are:

$$
\begin{array}{rll}
\max z_2^1 & = \sum_{\omega=1}^{2} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & \quad y_{11}^{1} + y_{12}^{1} = 55 & \\
& -1.25y_{11}^{1} - 1.14y_{12}^{1} & +y_{21}^{1} + y_{22}^{1} = 0 \\
& -1.25y_{21}^{1} - 1.14y_{22}^{1} & +y_{31}^{1} + y_{32}^{1} = 0 \\
& 1.25y_{31}^{1} + 1.14y_{32}^{1} & -y_{41}^{1} + y_{42}^{1} = 80 \\
& 1.06y_{31}^{1} + 1.12y_{32}^{1} & -y_{41}^{2} + y_{42}^{2} = 80 \\
& y_{ti}^{\omega} \geq 0, \forall i = 1,2, t \in \mathscr{T}, & \omega \in \Omega_4
\end{array} \tag{14}
$$

$$
\begin{array}{rll}
\max z_2^2 & = \sum_{\omega=3}^{4} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & \quad y_{11}^{1} + y_{12}^{1} = 55 & \\
& -1.25y_{11}^{1} - 1.14y_{12}^{1} & +y_{21}^{1} + y_{22}^{1} = 0 \\
& -1.06y_{21}^{1} - 1.12y_{22}^{1} & +y_{31}^{3} + y_{32}^{3} = 0 \\
& 1.25y_{31}^{3} + 1.14y_{32}^{3} & -y_{41}^{3} + y_{42}^{3} = 80 \\
& 1.06y_{31}^{3} + 1.12y_{32}^{3} & -y_{41}^{4} + y_{42}^{4} = 80 \\
& y_{ti}^{\omega} \geq 0, \forall i = 1,2, t \in \mathscr{T}, & \omega \in \Omega_5
\end{array} \tag{15}
$$

$$
\begin{array}{rll}
\max z_2^3 & = \sum_{\omega=5}^{6} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
 & -1.06y_{11}^1 - 1.12y_{12}^1 & +y_{21}^5 + y_{22}^5 = 0 \\
 & -1.25y_{21}^5 - 1.14y_{22}^5 & +y_{31}^5 + y_{32}^5 = 0 \\
 & 1.25y_{31}^5 + 1.14y_{32}^5 & -y_{41}^5 + y_{42}^5 = 80 \\
 & 1.06y_{31}^5 + 1.12y_{32}^5 & -y_{41}^6 + y_{42}^6 = 80 \\
 & y_{ti}^{\omega} \geq 0, \forall i = 1,2, & t \in \mathscr{T}, \omega \in \Omega_6
\end{array} \tag{16}
$$

$$
\begin{array}{rll}
\max z_2^4 & = \sum_{\omega=7}^{8} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
 & -1.06y_{11}^1 - 1.12y_{12}^1 & +y_{21}^5 + y_{22}^5 = 0 \\
 & -1.06y_{21}^5 - 1.12y_{22}^5 & +y_{31}^7 + y_{32}^7 = 0 \\
 & 1.25y_{31}^7 + 1.14y_{32}^7 & -y_{41}^7 + y_{42}^7 = 80 \\
 & 1.06y_{31}^7 + 1.12y_{32}^7 & -y_{41}^8 + y_{42}^8 = 80 \\
 & y_{ti}^{\omega} \geq 0, \forall i = 1,2, & t \in \mathscr{T}, \omega \in \Omega_7
\end{array} \tag{17}
$$

The corresponding submodels for break stage $t^* = 2$ in MPS format are in Appendix C, notice that the first line of inputData.dat file must be updated to 2.

- **Case 3**. Let the break stage $t^* = 3$, then there are $C = |\mathscr{G}_4| = 8$ clusters, see right decomposition in Figure 4 and, then, seven sets of scenario groups, say $\mathscr{G}^1 = \{1,2,4,8\}$, $\mathscr{G}^2 = \{1,2,4,9\}$, $\mathscr{G}^3 = \{1,2,5,10\}$, $\mathscr{G}^4 = \{1,2,5,11\}$, $\mathscr{G}^5 = \{1,3,6,12\}$, $\mathscr{G}^6 = \{1,3,6,13\}$, $\mathscr{G}^7 = \{1,3,7,14\}$ and $\mathscr{G}^8 = \{1,3,7,15\}$, and seven sets of scenarios: $\Omega^1 = \{1\}$, $\Omega^2 = \{2\}$, $\ldots$, and $\Omega^7 = \{7\}$.

  Notice that the 3-cluster tree matrix CT^3 is ST, see (6).

  The $|\mathscr{C}| = 8$ cluster or scenario submodels obtained for the break stage $t^* = 3$, (18)-(25) are:

$$
\begin{array}{rll}
\max z_3^1 & = \sum_{\omega=1} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
 & -1.25y_{11}^1 - 1.14y_{12}^1 & +y_{21}^1 + y_{22}^1 = 0 \\
 & -1.25y_{21}^1 - 1.14y_{22}^1 & +y_{31}^1 + y_{32}^1 = 0 \\
 & 1.25y_{31}^1 + 1.14y_{32}^1 & -y_{41}^1 + y_{42}^1 = 80 \\
 & y_{ti}^{\omega} \geq 0, \forall i = 1,2, t \in \mathscr{T}, & \omega \in \Omega_8
\end{array} \tag{18}
$$

$$
\begin{array}{rll}
\max z_3^2 & = \sum_{\omega=2} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
 & -1.25y_{11}^1 - 1.14y_{12}^1 & +y_{21}^1 + y_{22}^1 = 0 \\
 & -1.25y_{21}^1 - 1.14y_{22}^1 & +y_{31}^1 + y_{32}^1 = 0 \\
 & 1.06y_{31}^1 + 1.12y_{32}^1 & -y_{41}^2 + y_{42}^2 = 80 \\
 & y_{ti}^{\omega} \geq 0, \forall i = 1,2, t \in \mathscr{T}, & \omega \in \Omega_9
\end{array} \tag{19}
$$

$$
\begin{array}{rll}
\max z_3^3 & = \sum_{\omega=3} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
 & -1.25y_{11}^1 - 1.14y_{12}^1 & +y_{21}^1 + y_{22}^1 = 0 \\
 & -1.06y_{21}^1 - 1.12y_{22}^1 & +y_{31}^3 + y_{32}^3 = 0 \\
 & 1.25y_{31}^3 + 1.14y_{32}^3 & -y_{41}^3 + y_{42}^3 = 80 \\
 & y_{ti}^{\omega} \geq 0, \forall i = 1,2, t \in \mathscr{T}, & \omega \in \Omega_{10}
\end{array} \tag{20}
$$

$$
\begin{array}{rll}
\max z_3^4 & = \sum_{\omega=4} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
& -1.25y_{11}^1 - 1.14y_{12}^1 & +y_{21}^1 + y_{22}^1 = 0 \\
& -1.06y_{21}^1 - 1.12y_{22}^1 & +y_{31}^3 + y_{32}^3 = 0 \\
& 1.06y_{31}^3 + 1.12y_{32}^3 & -y_{41}^4 + y_{42}^4 = 80 \\
& y_{ti}^{\omega} \geq 0, \forall i = 1,2, t \in \mathscr{T}, & \omega \in \Omega_{11}
\end{array} \tag{21}
$$

$$
\begin{array}{rll}
\max z_3^5 & = \sum_{\omega=5} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
& -1.06y_{11}^1 - 1.12y_{12}^1 & +y_{21}^5 + y_{22}^5 = 0 \\
& -1.25y_{21}^5 - 1.14y_{22}^5 & +y_{31}^5 + y_{32}^5 = 0 \\
& 1.25y_{31}^5 + 1.14y_{32}^5 & -y_{41}^5 + y_{42}^5 = 80 \\
& y_{ti}^{\omega} \geq 0, \forall i = 1,2, & t \in \mathscr{T}, \omega \in \Omega_{12}
\end{array} \tag{22}
$$

$$
\begin{array}{rll}
\max z_3^6 & = \sum_{\omega=6} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
& -1.06y_{11}^1 - 1.12y_{12}^1 & +y_{21}^5 + y_{22}^5 = 0 \\
& -1.25y_{21}^5 - 1.14y_{22}^5 & +y_{31}^5 + y_{32}^5 = 0 \\
& 1.06y_{31}^5 + 1.12y_{32}^5 & -y_{41}^6 + y_{42}^6 = 80 \\
& y_{ti}^{\omega} \geq 0, \forall i = 1,2, & t \in \mathscr{T}, \omega \in \Omega_{13}
\end{array} \tag{23}
$$

$$
\begin{array}{rll}
\max z_3^7 & = \sum_{\omega=7} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
& -1.06y_{11}^1 - 1.12y_{12}^1 & +y_{21}^5 + y_{22}^5 = 0 \\
& -1.06y_{21}^5 - 1.12y_{22}^5 & +y_{31}^7 + y_{32}^7 = 0 \\
& 1.25y_{31}^7 + 1.14y_{32}^7 & -y_{41}^7 + y_{42}^7 = 80 \\
& y_{ti}^{\omega} \geq 0, \forall i = 1,2, & t \in \mathscr{T}, \omega \in \Omega_{14}
\end{array} \tag{24}
$$

$$
\begin{array}{rll}
\max z_3^8 & = \sum_{\omega=8} \frac{1}{8}(y_{41}^{\omega} - 4y_{42}^{\omega}) & \\
\text{s.t.} & y_{11}^1 + y_{12}^1 = 55 & \\
& -1.06y_{11}^1 - 1.12y_{12}^1 & +y_{21}^5 + y_{22}^5 = 0 \\
& -1.06y_{21}^5 - 1.12y_{22}^5 & +y_{31}^7 + y_{32}^7 = 0 \\
& 1.06y_{31}^7 + 1.12y_{32}^7 & -y_{41}^8 + y_{42}^8 = 80 \\
& y_{ti}^{\omega} \geq 0, \forall i = 1,2, & t \in \mathscr{T}, \omega \in \Omega_{15}
\end{array} \tag{25}
$$

The corresponding submodels for break stage $t^* = 3$ in MPS format are in Appendix D, notice that the first line of inputData.dat file must be updated to 3.

The C cluster submodels can be executed in parallel as explained in [Aldasoro *et al.*, 2012].

And the full model (8) in MPS format but ordered as previously explained can be shown in Appendix E.

5 Computational experience

The proposed main program has been implemented in a C++ experimental code. The computational experiments were conducted at the ARINA computational cluster provided by the SGI/IZO-SGIker at the UPV/EHU. ARINA provides 1400 cores divided as follows: 1112 xeon cores, 248 Itanium2 cores and 40 opteron cores. All calculation nodes are connected by an Infiniband network with high bandwidth and low latency. For the present experiments the xeon x86_64 architecture (Xeon Nehalem-EP E5520 @ 2.27GHz) type nodes have been used, consisting on 8 cores with 24 Gb of RAM with an QDR infiniband interconnection. Whereas for the calculation data storage, a 22 Tb high performance file system based on Lustre was used.

For testing the effect of stage ordering, we have computed the times for the testbed presented in [Escudero *et al.*, 2012] with COIN-OR V1.6.0, see [COIN-OR, 2013; Pérez and Garín, 2010] and IBM ILOG CPLEX V12.5, see [IBM, 2013; Pérez and Garín, 2011].

Tables 1 shows the main execution times for CPLEX under COIN-OR optimizer (first three columns) and for plain use of CPLEX (last three columns) with MIP gap of $1.e-6$. The headings are as follows: *by groups*, the elapsed time when variables are ordered by scenario groups; *by stages*, the elapsed time when variables are ordered by stages and *sratio* (%), saving ratio of the elapsed time in the comparison of execution by stages with respect of execution by variables nominal order (in this testbed the original order is by scenario groups).

Table 1: Order effect in execution time

Case	CPLEX under COIN			Plain use of CPLEX		
	by groups	by stages	sratio	by groups	by stages	sratio
P1	21	16	23.81	3	3	0
P2	1690	991	41.36	21	18	14.29
P3	–	–	–	–	–	–
P4	–	–	–	–	–	–
P5	3256	4767	-46.41	4	2	50.00
P6	4315	4160	3.59	3842	1603	58.28
P7	677	492	27.33	530	420	20.75
P8	–	–	–	–	–	–
P9	–	–	–	–	–	–
P10	12	8	33.33	187	102	45.45
P11	253	141	44.27	190	186	2.11
P12	17338	14435	16.74	–	–	–
P13	1277	739	42.13	2220	1203	45.81
P14	923	498	46.05	1678	1039	38.08

– : Time limit reached, 6h.

We can observe that the elapsed time is better in all the cases but one in each case, P5 with CPLEX under COIN-OR. The saving ratio is quite remarkable. So, the main program can be helpful in solving large-scale mixed integer problems because of the advantage of variables ordering and stage ordered cluster partitioning.

Appendix A Main cpp program

The main program `mainmps.cpp` is detailed below (C++ keywords are shown in blue and comments in green).

```
/* 2013-06-28 mainmps.cpp
 *
 *  A code for generating stage ordered full model and cluster submodels from
 *  multistage stochastic mixed integer optimization models using break stage.
 *  U. Aldasoro, M.A. Garï¿½n, M. Merino, G. Pï¿½rez.
 *
 *  Input files: Total.mps, inputData.dat
 *  Output files: Output.mps, outputData.dat, Cluster.mps files
 */

#include "pm.h"
#include "itoa.h"

int ncols,nints,k,j,i,t,nomega,ng,nt,nper=0,nper_max=0,
  nodes,nrows,nmodel,sum,i1,i2,g,w,ip,texpna,it,s,g0,gg,
  in,nintsmax, ncolsmax,nelements,imod,breakstage;

double dens;

int main(int argc, char **argv){

/* ******************************************************************
 Read MPS file
 ****************************************************************** */

  OsiClpSolverInterface solIN;
  OsiClpSolverInterface solOUT;
    solIN.readMps("Total");

  ofstream outputData("outputData.dat");

  nrows=solIN.getNumRows();
  double *drowlow;    drowlow=new double[nrows];
  double *drowup;     drowup=new double[nrows];

  ncols=solIN.getNumCols();
  double *dobj;     dobj=new double[ncols];
  double *dcollow;    dcollow=new double[ncols];
  double *dcolup;     dcolup=new double[ncols];

  const CoinPackedMatrix * A = solIN.getMatrixByRow();
  const bool colordered = A->isColOrdered();
  const int minor=A->getMinorDim();
  const int major=A->getMajorDim();
  const CoinBigIndex numels=A->getNumElements();
  const double * elem = A->getElements();
  const int * ind = A->getIndices();
  const CoinBigIndex * start = A->getVectorStarts();
  const int * len = A->getVectorLengths();

  nelements=solIN.getNumElements();

  nints=0;
  for (j=0;j<ncols;j++) {
    if(solIN.isInteger(j)==1) nints=nints+1;
    dobj[j]=solIN.getObjCoefficients()[j];
    dcollow[j]=solIN.getColLower()[j];
    dcolup[j]=solIN.getColUpper()[j];
  }
```

```
  for (j=0;j<nrows;j++) {
    drowlow[j]=solIN.getRowLower()[j];
    drowup[j]=solIN.getRowUpper()[j];
  }

  outputData<<"\n Number of variables: "<<ncols;
    outputData<<"\n Number of binary variables: "<<nints;
  outputData<<"\n Number of continuous variables: "<<ncols-nints;
  outputData<<"\n Number of constraints: "<<nrows;
  outputData<<"\n Number of nonzero elements: "<<numels;
  dens=(nelements*100.0)/((ncols * 1.0)*(1.0 * nrows));
  outputData<<"\n Density: "<< dens<<"%\n";

/* ****************************************************************
 Read the stochastic tree.
 ****************************************************************  */

    ifstream inputData("inputData.dat");

  int *nrowindx;        nrowindx=new int[nelements];
  int *mcolindx;        mcolindx=new int[nelements];

  outputData<<"\n Stored by rows ";
  k=0;
  for (i=0;i<nrows;i++) {
    for (s=start[i];s<start[i+1];s++) {
      mcolindx[k]=ind[k];
      nrowindx[k]=i;
      k=k+1;
    }
  }

  int *order;       order=new int[ncols];
  int *orderINV;      orderINV=new int[ncols];
  int *varstage;      varstage=new int[ncols];

  inputData>>breakstage;
  inputData>>nt;
  int *nrama;       nrama=new int[nt];
  int *nodes;       nodes=new int[nt];
  int *nodescum;      nodescum=new int[nt];
  int *tsuc;        tsuc=new int[nt];
  int *nummodel;      nummodel=new int[nt+1];
  int *mingt;       mingt=new int[nt+1];
  int *numberofnodes;   numberofnodes=new int[nt]; //number of nodes at each stage t in
      total
  int *ult;       ult=new int[nt+1]; //last node g for each stage t in total
  int *nints_t;     nints_t=new int[nt+1];
  int *ncont_t;     ncont_t=new int[nt+1];

  int **numberofvar = new int*[2];
  for(i=0;i<2;i++)  numberofvar[i] = new int[nt];

  for(t=0;t<nt;t++) inputData>>nodes[t];

  nodescum[0]=1;
  for(t=1;t<nt;t++) nodescum[t]=nodescum[t-1]+nodes[t];
  nmodel=nodes[breakstage];

  int *nrasuc;        nrasuc=new int[nodescum[nt-1]];
  for(i=0;i<(nodescum[nt-1]);i++) nrasuc[i] = 1;
  outputData<<"\n";
  for(t=0;t<nodescum[nt-2];t++){
    inputData>>nrasuc[t];
```

```
    outputData<<" "<<nrasuc[t];
  }

  for(t=0;t<nt;t++) inputData>>nints_t[t];
  for(t=0;t<nt;t++) inputData>>ncont_t[t];

  int *minwp;        minwp=new int[nmodel];
  int *maxwp;        maxwp=new int[nmodel];
  int *ngqp;         ngqp=new int[nmodel];  //number of nodes in each cluster
  int *ncolswqp;       ncolswqp=new int[nmodel];
  int *nintswqp;       nintswqp=new int[nmodel];

  nomega=nodes[nt-1];
  ng=nodescum[nt-1];

  double *p;         p=new double[nomega]; //weight for each scenario w (likelihood)

  inputData>>p[0];
  if(p[0]==0){
    for (w=1; w<=nomega; w++)   p[w-1]=1.0/(1.0*nomega);
  }else{
    for (w=1; w<nomega; w++)    inputData>>p[w];
  }

  int *nwcum;          nwcum=new int[ng]; //last scenario for each index in nrasuc
  int *etapa;          etapa=new int[ng+1]; //stage t of node g in total
  int *minwg;          minwg=new int[ng+1];
  int *fin;          fin=new int[ng+1]; //last binary variable for each node g in total
  int *fincont;        fincont=new int[ng+1]; //last continuous variable for each node g
      in total
  double *pesog;         pesog=new double[ng]; //weight for each group g

  int **scenariotree = new int*[nomega];
  for(i=0;i<nomega;i++) scenariotree[i] = new int[ng+1];

  int **clustertree = new int*[nmodel];
  for(i=0;i<nmodel;i++) clustertree[i] = new int[ng+1];

  int **numberofnodesp = new int*[nt]; //number of nodes at each stage t in clusters
  for(i=0;i<nt;i++) numberofnodesp[i] = new int[nmodel];

  double **pesop = new double*[nt]; //weight ratio for groups at stage t and cluster p
  for(i=0;i<nt;i++) pesop[i] = new double[nmodel];

  int **ultqp = new int*[nt+1]; //last node g for each stage t in clusters
  for(i=0;i<(nt+1);i++) ultqp[i] = new int[nmodel];

  int **finqp = new int*[ng+1]; //last binary variable for each node g in clusters
  for(i=0;i<(ng+1);i++) finqp[i] = new int[nmodel];

  int **finqcontp = new int*[ng+1]; //last continuous variable for each node g in
      clusters
  for(i=0;i<(ng+1);i++) finqcontp[i] = new int[nmodel];

  int **invgrupo = new int*[ng+1];  //group of cluster submodel corresponds to group of
      total model
  for(i=0;i<(ng+1);i++) invgrupo[i] = new int[nmodel];

  int **nrowsindex = new int*[nrows];
  for(i=0;i<nrows;i++)  nrowsindex[i] = new int[nmodel];

  tsuc[0]=0; tsuc[1]=1;
  for(t=2;t<nt;t++){
    for(i=nodescum[t-2];i<nodescum[t-1];i++)
    tsuc[t]=nodescum[t-1];
```

```
  }

  outputData<<"\n\n T="<<nt<<" Omega="<<nomega<<" G="<<ng<<" breakstage="<<breakstage<<"
     q="<<nmodel<<" \n";

  outputData<<"\n Number of leafs at each consecutive node \n";
  for(i=0;i<ng-nomega;i++)  outputData<<" "<<nrasuc[i];

  outputData<<"\n Number of nodes at each stage \n";
  for(t=0;t<nt;t++)     outputData<<" "<<nodes[t];

  outputData<<"\n Number of cummulated nodes at each stage \n";
  for(t=0;t<nt;t++)     outputData<<" "<<nodescum[t];

  outputData<<"\n Index of nrasuc in which a new stage starts \n";
  for(i=0;i<nt;i++)     outputData<<" "<<tsuc[i];

  outputData<<"\n Number of scenarios "<< nodes[nt-1];
  outputData<<"\n Number of groups (tree nodes) "<<nodescum[nt-1];


  //nwcum: last scenario for each index in nrasuc
  for(i=0;i<ng;i++)   nwcum[i]=0;
  //t=T
  for(i=tsuc[nt-1];i<ng;i++){
    nwcum[i]=nwcum[i-1]+nrasuc[i];
  }
  //t<T
  for(t=nt-1;t>=1;t--){
    i1=tsuc[t-1];
    i2=tsuc[t];
    sum=0;
    for(i=i1;i<i2;i++){
      sum=sum+nrasuc[i];
      nwcum[i]=nwcum[i2+sum-1];
    }
  }
  outputData<<"\n\n Last scenario for each index in nrasuc \n";
  for(i=0;i<ng;i++) outputData<<" "<<nwcum[i];

  //ST(w,g)
  outputData<<"\n\n Stage of node g at each scenario w (scenariotreematrix): ";

  //g=0 g=1
  for(i=0;i<nomega;i++){
    scenariotree[i][0]=0;
    scenariotree[i][1]=1;
    for(g=2;g<=ng;g++){
      scenariotree[i][g]=0;
    }
  }
  //g=2 ... g=ng
  i=1;
  g=2;
  for(t=2;t<=nt;t++){
    for(w=1;w<=nomega;w++){
      scenariotree[w-1][g]=t;
      if(w==nwcum[i]){
        g=g+1;
        i=i+1;
      }
    }
  }

/* ******************************************************************
```

```
Print numberofvar and scenariotree
****************************************************************** */

 for(t=0;t<nt;t++){
   numberofvar[0][t]=nints_t[t]*nodes[t];
   numberofvar[1][t]=ncont_t[t]*nodes[t];
 }

 //n_x(t)
 outputData<<"\n Number of binary variables at each stage : \n";
 for(j=1;j<=nt;j++){
   outputData<<" "<<numberofvar[0][j-1];
 }

 //n_y(t)
 outputData<<"\n Number of continuous variables at each stage : \n";
 for(j=1;j<=nt;j++){
   outputData<<" "<<numberofvar[1][j-1];
 }

 //scencariotree ST(p,g)
 outputData<<"\n Node ";
 for(j=0;j<=ng;j++) outputData<<" "<<j;

 for(i=0;i<nomega;i++){
   outputData<<" \n w="<<i+1<<": ";
   for(j=0;j<=ng;j++){
     outputData<<" "<<scenariotree[i][j];
   }
 }

/* ******************************************************************
Stage (etapa) and minimum scenario (minwg) for each group
****************************************************************** */

 // t(g) min_w(g)
   etapa[0]=0;
   minwg[0]=0;
   for(g=1;g<=ng;g++){
        for(w=0;w<nomega;w++){
            if(scenariotree[w][g] != 0){
                etapa[g]=scenariotree[w][g];
                minwg[g]=w;
                w=nomega;
            }
        }
   }
   outputData<<"\n\n Grupo g: ";
   for(g=0;g<=ng;g++) outputData<<" "<<g;
 outputData<<"\n Etapa t: ";
   for(g=0;g<=ng;g++) outputData<<" "<<etapa[g];
   outputData<<"\n Min esc: ";
   for(g=0;g<=ng;g++) outputData<<" "<<minwg[g];

/* ******************************************************************
Number of clusters and minimum group for each stage, break stage
minimum and maximum scenario for each cluster
****************************************************************** */

 nummodel[0]=0;
 mingt[0]=0;mingt[1]=1;
 for(t=1;t<=nt;t++){
   nummodel[t]=0;
   for(g=1;g<=ng;g++){
     if(etapa[g]==t) nummodel[t]=nummodel[t]+1;
```

```
      if(etapa[g]==t+1) {
        mingt[t+1]=g;
        g=ng+1;
      }
    }
    if(nummodel[t]==nmodel){
      texpna=t-1;
      for(ip=0;ip<nmodel;ip++){
        minwp[ip]=minwg[mingt[t]+ip];
        if(ip != nmodel-1) maxwp[ip]=minwg[mingt[t]+ip+1]-1;
          else
        maxwp[ip]=nomega-1;
      }
    }
  }
  outputData<<"\n\n Num clust q: ";
  for(t=0;t<=nt;t++) outputData<<" "<<nummodel[t];
  outputData<<"\n Min grupo g: ";
  for(t=0;t<=nt;t++) outputData<<" "<<mingt[t];
  outputData<<"\n Min-max w: ";
  for(ip=0;ip<nmodel;ip++) outputData<<" "<<minwp[ip]<<"-"<<maxwp[ip];

/* *******************************************************************
Build the cluster tree matrix: clustertree
 ****************************************************************** */

  //CT(p,g) =clustertree: number of stage of group g in cluster p
  for(ip=0;ip<nmodel;ip++){
    clustertree[ip][0]=0;
    for(g=1;g<=ng;g++){
      if(etapa[g] <= texpna+1)
        clustertree[ip][g]=scenariotree[minwp[ip]][g];
      else
      {
        for(w=minwp[ip];w<=maxwp[ip];w++){
          clustertree[ip][g]=0;
          if(scenariotree[w][g] != 0){
            clustertree[ip][g]=scenariotree[w][g];
            w=maxwp[ip]+1;
          }
        }
      }
    }
  }
  outputData<<"\n\n Cluster Tree Matrix: ";
  outputData<<"\n\n Stage of node g at each cluster p : ";
  outputData<<" \n Relation between total and cluster problems ";
  outputData<<"\n clustertree (antiguo res7) ";
  for(ip=0;ip<nmodel;ip++){
    outputData<<"\n p="<<ip+1<<": ";
    for(g=0;g<=ng;g++) outputData<<" "<<clustertree[ip][g];
  }

/* *******************************************************************
Number of nodes at stage t in total (numberofnodes) and clusters (numberofnodesp)
 ****************************************************************** */

  // |G_t|-|G_t-1|
  for(it=0;it<nt;it++){
    numberofnodes[it]=0;
    for(g=1;g<=ng;g++){
      for(ip=0;ip<nmodel;ip++){
        if(clustertree[ip][g]==it+1){
          numberofnodes[it]=numberofnodes[it]+1;
          ip=nmodel;
```

```
        }
      }
    }
  }
  // |G_t(p)|-|G_t-1(p)|
  for(it=0;it<nt;it++){
    for(ip=0;ip<nmodel;ip++){
     numberofnodesp[it][ip]=0;
     for(g=1;g<=ng;g++)
       if(clustertree[ip][g]==it+1)
         numberofnodesp[it][ip]=numberofnodesp[it][ip]+1;
    }
  }
  outputData<<"\n\n Number of nodes at each stage t: ";
  outputData<<"\n numberofnodes= ";
  for (it=0;it<nt;it++) outputData<<" "<<numberofnodes[it];
  for(ip=0;ip<nmodel;ip++){
   outputData<<"\n p="<<ip+1<<": ";
   for (it=0;it<nt;it++)
     outputData<<" "<<numberofnodesp[it][ip];
  }

/* *****************************************************************
Last node g for each stage t in total (ult) and clusters (ultqp)
 ***************************************************************** */

  // |G_t|
  ult[0]=0;
  for (it=1;it<=nt;it++)
    ult[it]=ult[it-1]+numberofnodes[it-1];

  // |G_t(p)|
  for(ip=0;ip<nmodel;ip++){
    ultqp[0][ip]=0;
    for (it=1;it<=nt;it++){
      ultqp[it][ip]=ultqp[it-1][ip]+numberofnodesp[it-1][ip];
    }
  }
  outputData<<"\n\n Last node of each stage t :";
  outputData<<"\n ult=";
  for (j=0;j<=nt;j++)
    outputData<<" "<<ult[j];
    outputData<<"\n ultqp= ";
  for (j=0;j<nmodel;j++){
    outputData<<"\n p="<<j+1<<": ";
    for (i=0;i<=nt;i++)
      outputData<<" "<<ultqp[i][j];
  }


/* *****************************************************************
Number of nodes in clusters (ngqp)
 ***************************************************************** */

  // |G(p)|
  outputData<<"\n\n ngqp=";
  for(ip=0;ip<nmodel;ip++){
    ngqp[ip]=0;
    for (it=0;it<nt;it++){
    ngqp[ip]=ngqp[ip]+numberofnodesp[it][ip];
    }
    outputData<<"\n p="<<ip+1<<" cluster has "<<ngqp[ip]<<" nodes";
  }
  outputData<<"\n\n Explicit NA until stage "<<texpna;
```

```
/* ****************************************************************
Last binary variable for node g in total (fin) and clusters (finqp)
Last continuous variable for node g in total (fincont) and clusters (finqcontp)
 **************************************************************** */

  for (g=0;g<=ng;g++){
    for(ip=0;ip<nmodel;ip++){
      finqp[g][ip]=0;
      finqcontp[g][ip]=0;
    }
  }

  fin[0]=0;
  fincont[0]=0;
  for(it=1;it<=nt;it++)
  for(g=ult[it-1]+1;g<=ult[it];g++){
    fin[g]=fin[g-1]+numberofvar[0][it-1]/numberofnodes[it-1];
    fincont[g]=fincont[g-1]+numberofvar[1][it-1]/numberofnodes[it-1];
  }
  for(ip=0;ip<nmodel;ip++)
    {
        finqp[0][ip]=0;
        finqcontp[0][ip]=0;
        for(it=1;it<=nt;it++)
            for(g=ultqp[it-1][ip]+1;g<=ultqp[it][ip];g++){
                finqp[g][ip]=finqp[g-1][ip]+numberofvar[0][it-1]/numberofnodes[it-1];
                finqcontp[g][ip]=finqcontp[g-1][ip]+numberofvar[1][it-1]/numberofnodes[it
                    -1];
            }
    }
  outputData<<"\n\n Last binary variable for each node g :";
  outputData<<"\n fin= ";
    for (j=0;j<=ng;j++)   outputData<<" "<<fin[j];
  outputData<<"\n finqp= ";
  for (i=0;i<nmodel;i++){
         outputData<<"\n p="<<i+1<<": ";
         for (j=0;j<=ngqp[i];j++)
             outputData<<" "<<finqp[j][i];
  }

    outputData<<"\n\n Last continuous variable for each node g :";
  outputData<<"\n fincont= ";
    for (j=0;j<=ng;j++)   outputData<<" "<<fincont[j];
  outputData<<"\n finqpcont= ";
  for (i=0;i<nmodel;i++){
    outputData<<"\n p="<<i+1<<": ";
    for (j=0;j<=ngqp[i];j++)
      outputData<<" "<<finqcontp[j][i];
  }

/* ****************************************************************
Number of binary (nintswqp) and total (ncolswqp) variables in clusters
 **************************************************************** */

    //n_x(p), n(p)
    nintsmax=0;
    ncolsmax=0;
    outputData<<"\n\n Number of variables by cluster: ";
    for(in=0;in<nmodel;in++){
            nintswqp[in]=finqp[ngqp[in]][in];
            outputData<<"\n p="<<in+1<<": "<<nintswqp[in]<<" integer and ";
        ncolswqp[in]=nintswqp[in]+finqcontp[ngqp[in]][in];
        outputData<<ncolswqp[in]-nintswqp[in]<<" continuous and  ";
        outputData<<ncolswqp[in]<<" ( total variables )";
        if(nintswqp[in]>nintsmax) nintsmax=nintswqp[in];
```

```
        if(ncolswqp[in]>ncolsmax) ncolsmax=ncolswqp[in];
    }
    outputData<<"\n max nintswqp="<<nintsmax;
    outputData<<"\n max ncolswqp="<<ncolsmax;

/* ****************************************************************
Order of variables in total (order) and clusters (ordenqq) BY NODE
 ************************************************************** */

  k=0;
  for(t=0;t<ncols;t++) {
    if(k<2){
      inputData>>order[t];
      orderINV[order[t]]=t;
      if(order[t]==0) k=k+1;
    }
  }

  // if nominal
  if(k>1){
    outputData<<"\n ORDEN NOMINAL DE VARIABLES";
    for(t=0;t<ncols;t++) {
      order[t]=t;
      orderINV[order[t]]=t;
    }
  }

  int **ordenqq = new int*[ncols];
  for(i=0;i<ncols;i++)  ordenqq[i] = new int[nmodel];

  int **ordenqqINV = new int*[ncols];
  for(i=0;i<ncols;i++)  ordenqqINV[i] = new int[nmodel];

  int *binCont;     binCont=new int[nmodel];

  for (ip=0;ip<nmodel;ip++){
    for (j=0;j<ncols;j++) ordenqqINV[j][ip]=-1;
    k=0;
    binCont[ip]=0;
    for (g=1;g<=ng;g++){
      if(clustertree[ip][g]>0){
        if(fin[g-1]!=fin[g]){
          for (i=fin[g-1];i<fin[g];i++){
            ordenqq[k][ip]=order[i];
            ordenqqINV[order[i]][ip]=k;
            k=k+1;
            binCont[ip]=binCont[ip]+1;
          }
        }
      }
    }
  }

  for (ip=0;ip<nmodel;ip++){
    k=binCont[ip];
    for (g=1;g<=ng;g++){
      if(clustertree[ip][g]>0){
        if(fincont[g-1]!=fincont[g]){
          for (i=fincont[g-1];i<fincont[g];i++){
            ordenqq[k][ip]=order[i+nints];
            ordenqqINV[order[i+nints]][ip]=k;
            k=k+1;
          }
        }
      }
```

```
    }
  }

  k=0;
  for(i=0;i<nt;i++){
    if(numberofvar[0][i]>0){
      for(s=0;s<numberofvar[0][i];s++)  varstage[order[s+k]]=i;
      k=k+numberofvar[0][i];
    }
  }

  for(i=0;i<nt;i++){
    if(numberofvar[1][i]>0){
      for(s=0;s<numberofvar[1][i];s++)  varstage[order[s+k]]=i;
      k=k+numberofvar[1][i];
    }
  }

    outputData<<"\n\n Order of integer variables by nodes: ";
    outputData<<"\n\n Order of continuous variables by nodes: ";

    for (ip=0;ip<nmodel;ip++){
        outputData<<"\n Cluster p="<<ip+1;
        for (i=0;i<nintswqp[ip];i++)
            outputData<<"\n Bin: i="<<i<<" ordenqq=("<<ordenqq[i][ip]<<")";
        for (i=nintswqp[ip];i<ncolswqp[ip];i++)
            outputData<<"\n Cont: i="<<i<<" ordenqq=("<<ordenqq[i][ip]<<")";
    }

/* ****************************************************************
p[nomega], pesog[ng+1], pesop[nt][nmodel]: weights
 ****************************************************************  */

  for(i=0;i<nmodel;i++){
    gg=1;
    for(g=1;g<=ng;g++){
       if(clustertree[i][g] != 0){
         invgrupo[gg][i]=g;
         gg=gg+1;
       }
    }
  }

  outputData<<"\n\n Weights (likelihoods) for each scenario w:";
  for (w=1; w<=nomega; w++) {
    outputData<<"\n p["<<w<<"]="<<p[w-1];
  }

  outputData<<"\n\n Weights for each scenario group g:";
  pesog[0]=1.0;
  pesog[1]=1.0;
  outputData<<"\n pesog[1]="<<pesog[1];
  for(t=2;t<=nt;t++){
    for(g=ult[t-1]+1;g<ult[t];g++){
      pesog[g]=0.0;
      for(w=minwg[g];w<minwg[g+1];w++)  pesog[g]=pesog[g]+p[w];
      outputData<<"\n pesog["<<g<<"]="<<pesog[g];
    }
    g=ult[t];
    pesog[g]=0.0;
    for(w=minwg[g];w<nomega;w++)  pesog[g]=pesog[g]+p[w];
    outputData<<"\n pesog["<<g<<"]="<<pesog[g];
  }

  outputData<<"\n\n Weight ratios for groups at stage t and cluster p (pesop)";
```

```
  for(ip=0;ip<nmodel;ip++)
    for(t=0;t<nt;t++)
      pesop[t][ip]=1.0;
  //before explicit NA
  for(ip=0;ip<nmodel;ip++)
    for(t=1;t<=texpna;t++)
      pesop[t-1][ip]=pesog[invgrupo[texpna+1][ip]]/pesog[invgrupo[t][ip]];
  for(ip=0;ip<nmodel;ip++){
    outputData<<"\n Cluster p="<<ip+1;
    for(t=0;t<nt;t++) outputData<<" "<<pesop[t][ip];
  }

//* *****************************************************************
// Reorder MPS problem
// ***************************************************************** */

  double *dobj_st;         dobj_st=new double[ncols];
  double *dcollow_st;        dcollow_st=new double[ncols];
  double *dcolup_st;         dcolup_st=new double[ncols];
  int *mcolindx_st;        mcolindx_st=new int[nelements];

  for (j=0;j<ncols;j++) {
    dobj_st[j]=dobj[order[j]];
    dcollow_st[j]=dcollow[order[j]];
    dcolup_st[j]=dcolup[order[j]];
  }
  for (i=0;i<nelements;i++) mcolindx_st[i]=orderINV[mcolindx[i]];

  CoinPackedMatrix AA(colordered,nrowindx,mcolindx_st,elem,numels);
  solOUT.loadProblem(AA,dcollow_st,dcolup_st,dobj_st,drowlow,drowup);

    for(i=0;i<nints;i++) solOUT.setInteger(i);

  solOUT.writeMps("Output");

///* *****************************************************************
// Create Clusters
// ***************************************************************** */

  OsiClpSolverInterface *solCluster;
  solCluster=new OsiClpSolverInterface[nmodel];

  int *mrowelements;       mrowelements=new int[nrows];
  int *mrowsize;           mrowsize=new int[nmodel];
  int *nelementsize;       nelementsize=new int[nmodel];

  int **mrowbelongs = new int*[nrows];
  for(i=0;i<nrows;i++)  mrowbelongs[i] = new int[nmodel];

  double *dobj_stqp;       dobj_stqp=new double[ncols];
  double *dcollow_stqp;    dcollow_stqp=new double[ncols];
  double *dcolup_stqp;     dcolup_stqp=new double[ncols];
  double *elem_stqp;       elem_stqp=new double[nelements];
  int *mcolindx_stqp;      mcolindx_stqp=new int[nelements];
  int *mrowindx_stqp;      mrowindx_stqp=new int[nelements];
  double *drowlowqp;       drowlowqp=new double[nrows];
  double *drowupqp;      drowupqp=new double[nrows];

  const char* final;
  char buffer [33];
  char modelo[80];

  for(imod=0;imod<nmodel;imod++)  {
    mrowsize[imod]=0;
    nelementsize[imod]=0;
```

```
    for(i=0;i<nrows;i++){
      mrowbelongs[i][imod]=1;
      mrowelements[i]=0;
    }
  }

  for (i=0;i<nelements;i++) {
    for(imod=0;imod<nmodel;imod++){
      if(ordenqqINV[mcolindx[i]][imod]==(-1)){
        mrowbelongs[nrowindx[i]][imod]=0;
      }
    }
    mrowelements[nrowindx[i]]=mrowelements[nrowindx[i]]+1;
  }

  for(i=0;i<nrows;i++){
    for(imod=0;imod<nmodel;imod++){
      if(mrowbelongs[i][imod]==1) {
        mrowsize[imod]=mrowsize[imod]+1;
        nelementsize[imod]=nelementsize[imod]+mrowelements[i];
      }
    }
  }
  for(imod=0;imod<nmodel;imod++){


    for (j=0;j<ncolswqp[imod];j++) {
      dobj_stqp[j]=dobj[ordenqq[j][imod]]*pesop[varstage[ordenqq[j][imod]]][imod];
      dcollow_stqp[j]=dcollow[ordenqq[j][imod]];
      dcolup_stqp[j]=dcolup[ordenqq[j][imod]];
    }

    k=0;
    s=-1;
    gg=0;
    for (i=0;i<nelements;i++) {
      if(mrowbelongs[nrowindx[i]][imod]==1){
        if(s==-1) s=nrowindx[i];

        //Cols
        mcolindx_stqp[gg]=ordenqqINV[mcolindx[i]][imod];

        //Rows
        if(s!=nrowindx[i]) {
          k=k+1;
          s=nrowindx[i];
        }
        mrowindx_stqp[gg]=k;
        drowlowqp[k]=drowlow[nrowindx[i]];
        drowupqp[k]=drowup[nrowindx[i]];

        //Elements
        elem_stqp[gg]=elem[i];

        gg=gg+1;
      }
    }

    outputData<<"\n Cluster "<<imod<<" has "<<ncolswqp[imod]<<" variables ("<<nintswqp[
        imod]<<" integer) "<<mrowsize[imod]<<" rows and "<<nelementsize[imod]<<" nonzero
        elements";

    CoinPackedMatrix AC(colordered,mrowindx_stqp,mcolindx_stqp,elem_stqp,nelementsize[
        imod]);
    solCluster[imod].loadProblem(AC,dcollow_stqp,dcolup_stqp,dobj_stqp,drowlowqp,drowupqp
```

```
        );

    for(i=0;i<nintswqp[imod];i++) solCluster[imod].setInteger(i);

    final="";
    final=itoa(imod+1, buffer,10);
    strcpy (modelo,"Cluster");
    strcat (modelo,final);
    puts (modelo);

    solCluster[imod].writeMps(modelo);
  }

outputData.close();
inputData.close();

return 0;
}
```

mainmps.cpp

Appendix B MPS 1-decomposition

The cluster submodels for break stage $t^* = 1$ in MPS format are as follows:

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
 E  R0000004
 E  R0000005
 E  R0000006
 E  R0000007
COLUMNS
    C0000000  R0000000  1.            R0000001   -1.25
    C0000001  R0000000  1.            R0000001   -1.14
    C0000002  R0000001  1.            R0000002   -1.25
    C0000002  R0000003   -1.06
    C0000003  R0000001  1.            R0000002   -1.14
    C0000003  R0000003   -1.12
    C0000004  R0000002  1.            R0000004  1.25
    C0000004  R0000005  1.06
    C0000005  R0000002  1.            R0000004  1.14
    C0000005  R0000005  1.12
    C0000006  R0000003  1.            R0000006  1.25
    C0000006  R0000007  1.06
    C0000007  R0000003  1.            R0000006  1.14
    C0000007  R0000007  1.12
    C0000008  OBJROW    0.125         R0000004   -1.
    C0000009  OBJROW     -0.5         R0000004  1.
    C0000010  OBJROW    0.125         R0000005   -1.
    C0000011  OBJROW     -0.5         R0000005  1.
    C0000012  OBJROW    0.125         R0000006   -1.
    C0000013  OBJROW     -0.5         R0000006  1.
    C0000014  OBJROW    0.125         R0000007   -1.
    C0000015  OBJROW     -0.5         R0000007  1.
RHS
    RHS       R0000000  55.           R0000004  80.
    RHS       R0000005  80.           R0000006  80.
    RHS       R0000007  80.
ENDATA
```

Cluster1.mps (11)

```
NAME          BLANK
ROWS
```

```
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
 E  R0000004
 E  R0000005
 E  R0000006
 E  R0000007
COLUMNS
    C0000000  R0000000  1.              R0000001   -1.06
    C0000001  R0000000  1.              R0000001   -1.12
    C0000002  R0000001  1.              R0000002   -1.25
    C0000002  R0000003   -1.06
    C0000003  R0000001  1.              R0000002   -1.14
    C0000003  R0000003   -1.12
    C0000004  R0000002  1.              R0000004  1.25
    C0000004  R0000005  1.06
    C0000005  R0000002  1.              R0000004  1.14
    C0000005  R0000005  1.12
    C0000006  R0000003  1.              R0000006  1.25
    C0000006  R0000007  1.06
    C0000007  R0000003  1.              R0000006  1.14
    C0000007  R0000007  1.12
    C0000008  OBJROW    0.125           R0000004   -1.
    C0000009  OBJROW     -0.5           R0000004  1.
    C0000010  OBJROW    0.125           R0000005   -1.
    C0000011  OBJROW     -0.5           R0000005  1.
    C0000012  OBJROW    0.125           R0000006   -1.
    C0000013  OBJROW     -0.5           R0000006  1.
    C0000014  OBJROW    0.125           R0000007   -1.
    C0000015  OBJROW     -0.5           R0000007  1.
RHS
    RHS       R0000000  55.             R0000004  80.
    RHS       R0000005  80.             R0000006  80.
    RHS       R0000007  80.
ENDATA
```

Cluster2.mps (12)

Appendix C MPS 2-decomposition

The cluster submodels for break stage $t^* = 2$ in MPS format are as follows:

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
 E  R0000004
COLUMNS
    C0000000  R0000000  1.              R0000001   -1.25
    C0000001  R0000000  1.              R0000001   -1.14
    C0000002  R0000001  1.              R0000002   -1.25
    C0000003  R0000001  1.              R0000002   -1.14
    C0000004  R0000002  1.              R0000003  1.25
    C0000004  R0000004  1.06
    C0000005  R0000002  1.              R0000003  1.14
    C0000005  R0000004  1.12
    C0000006  OBJROW    0.125           R0000003   -1.
    C0000007  OBJROW     -0.5           R0000003  1.
    C0000008  OBJROW    0.125           R0000004   -1.
    C0000009  OBJROW     -0.5           R0000004  1.
RHS
    RHS       R0000000  55.             R0000003  80.
    RHS       R0000004  80.
ENDATA
```

Cluster1.mps (14)

```
NAME          BLANK
ROWS
```

```
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
 E  R0000004
COLUMNS
    C0000000  R0000000  1.            R0000001   -1.25
    C0000001  R0000000  1.            R0000001   -1.14
    C0000002  R0000001  1.            R0000002   -1.06
    C0000003  R0000001  1.            R0000002   -1.12
    C0000004  R0000002  1.            R0000003  1.25
    C0000004  R0000004  1.06
    C0000005  R0000002  1.            R0000003  1.14
    C0000005  R0000004  1.12
    C0000006  OBJROW    0.125         R0000003   -1.
    C0000007  OBJROW     -0.5         R0000003  1.
    C0000008  OBJROW    0.125         R0000004   -1.
    C0000009  OBJROW     -0.5         R0000004  1.
RHS
    RHS       R0000000  55.           R0000003  80.
    RHS       R0000004  80.
ENDATA
```

Cluster2.mps (15)

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
 E  R0000004
COLUMNS
    C0000000  R0000000  1.            R0000001   -1.06
    C0000001  R0000000  1.            R0000001   -1.12
    C0000002  R0000001  1.            R0000002   -1.25
    C0000003  R0000001  1.            R0000002   -1.14
    C0000004  R0000002  1.            R0000003  1.25
    C0000004  R0000004  1.06
    C0000005  R0000002  1.            R0000003  1.14
    C0000005  R0000004  1.12
    C0000006  OBJROW    0.125         R0000003   -1.
    C0000007  OBJROW     -0.5         R0000003  1.
    C0000008  OBJROW    0.125         R0000004   -1.
    C0000009  OBJROW     -0.5         R0000004  1.
RHS
    RHS       R0000000  55.           R0000003  80.
    RHS       R0000004  80.
ENDATA
```

Cluster3.mps (16)

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
 E  R0000004
COLUMNS
    C0000000  R0000000  1.            R0000001   -1.06
    C0000001  R0000000  1.            R0000001   -1.12
    C0000002  R0000001  1.            R0000002   -1.06
    C0000003  R0000001  1.            R0000002   -1.12
    C0000004  R0000002  1.            R0000003  1.25
    C0000004  R0000004  1.06
    C0000005  R0000002  1.            R0000003  1.14
    C0000005  R0000004  1.12
    C0000006  OBJROW    0.125         R0000003   -1.
    C0000007  OBJROW     -0.5         R0000003  1.
    C0000008  OBJROW    0.125         R0000004   -1.
    C0000009  OBJROW     -0.5         R0000004  1.
RHS
    RHS       R0000000  55.           R0000003  80.
    RHS       R0000004  80.
```

```
ENDATA
```

Cluster4.mps (17)

Appendix D MPS 3-decomposition

The cluster submodels for break stage $t^* = 3$ in MPS format are as follows:

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
COLUMNS
    C0000000  R0000000  1.            R0000001   -1.25
    C0000001  R0000000  1.            R0000001   -1.14
    C0000002  R0000001  1.            R0000002   -1.25
    C0000003  R0000001  1.            R0000002   -1.14
    C0000004  R0000002  1.            R0000003  1.25
    C0000005  R0000002  1.            R0000003  1.14
    C0000006  OBJROW    0.125         R0000003   -1.
    C0000007  OBJROW     -0.5         R0000003  1.
RHS
    RHS       R0000000  55.           R0000003  80.
ENDATA
```

Cluster1.mps (18)

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
COLUMNS
    C0000000  R0000000  1.            R0000001   -1.25
    C0000001  R0000000  1.            R0000001   -1.14
    C0000002  R0000001  1.            R0000002   -1.25
    C0000003  R0000001  1.            R0000002   -1.14
    C0000004  R0000002  1.            R0000003  1.06
    C0000005  R0000002  1.            R0000003  1.12
    C0000006  OBJROW    0.125         R0000003   -1.
    C0000007  OBJROW     -0.5         R0000003  1.
RHS
    RHS       R0000000  55.           R0000003  80.
ENDATA
```

Cluster2.mps (19)

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
COLUMNS
    C0000000  R0000000  1.            R0000001   -1.25
    C0000001  R0000000  1.            R0000001   -1.14
    C0000002  R0000001  1.            R0000002   -1.06
    C0000003  R0000001  1.            R0000002   -1.12
    C0000004  R0000002  1.            R0000003  1.25
    C0000005  R0000002  1.            R0000003  1.14
    C0000006  OBJROW    0.125         R0000003   -1.
    C0000007  OBJROW     -0.5         R0000003  1.
RHS
    RHS       R0000000  55.           R0000003  80.
ENDATA
```

Cluster3.mps (20)

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
COLUMNS
    C0000000  R0000000  1.             R0000001   -1.25
    C0000001  R0000000  1.             R0000001   -1.14
    C0000002  R0000001  1.             R0000002   -1.06
    C0000003  R0000001  1.             R0000002   -1.12
    C0000004  R0000002  1.             R0000003  1.06
    C0000005  R0000002  1.             R0000003  1.12
    C0000006  OBJROW    0.125          R0000003   -1.
    C0000007  OBJROW     -0.5          R0000003  1.
RHS
    RHS       R0000000  55.            R0000003  80.
ENDATA
```

Cluster4.mps (21)

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
COLUMNS
    C0000000  R0000000  1.             R0000001   -1.06
    C0000001  R0000000  1.             R0000001   -1.12
    C0000002  R0000001  1.             R0000002   -1.25
    C0000003  R0000001  1.             R0000002   -1.14
    C0000004  R0000002  1.             R0000003  1.25
    C0000005  R0000002  1.             R0000003  1.14
    C0000006  OBJROW    0.125          R0000003   -1.
    C0000007  OBJROW     -0.5          R0000003  1.
RHS
    RHS       R0000000  55.            R0000003  80.
ENDATA
```

Cluster5.mps (22)

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
COLUMNS
    C0000000  R0000000  1.             R0000001   -1.06
    C0000001  R0000000  1.             R0000001   -1.12
    C0000002  R0000001  1.             R0000002   -1.25
    C0000003  R0000001  1.             R0000002   -1.14
    C0000004  R0000002  1.             R0000003  1.06
    C0000005  R0000002  1.             R0000003  1.12
    C0000006  OBJROW    0.125          R0000003   -1.
    C0000007  OBJROW     -0.5          R0000003  1.
RHS
    RHS       R0000000  55.            R0000003  80.
ENDATA
```

Cluster6.mps (23)

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
```

```
 E  R0000002
 E  R0000003
COLUMNS
    C0000000  R0000000  1.             R0000001   -1.06
    C0000001  R0000000  1.             R0000001   -1.12
    C0000002  R0000001  1.             R0000002   -1.06
    C0000003  R0000001  1.             R0000002   -1.12
    C0000004  R0000002  1.             R0000003  1.25
    C0000005  R0000002  1.             R0000003  1.14
    C0000006  OBJROW    0.125          R0000003   -1.
    C0000007  OBJROW     -0.5          R0000003  1.
RHS
    RHS       R0000000  55.            R0000003  80.
ENDATA
```

Cluster7.mps (24)

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
COLUMNS
    C0000000  R0000000  1.             R0000001   -1.06
    C0000001  R0000000  1.             R0000001   -1.12
    C0000002  R0000001  1.             R0000002   -1.06
    C0000003  R0000001  1.             R0000002   -1.12
    C0000004  R0000002  1.             R0000003  1.06
    C0000005  R0000002  1.             R0000003  1.12
    C0000006  OBJROW    0.125          R0000003   -1.
    C0000007  OBJROW     -0.5          R0000003  1.
RHS
    RHS       R0000000  55.            R0000003  80.
ENDATA
```

Cluster8.mps (25)

Appendix E MPS stage ordered full model

The stage ordered full model (8) in MPS format is as follows:

```
NAME          BLANK
ROWS
 N  OBJROW
 E  R0000000
 E  R0000001
 E  R0000002
 E  R0000003
 E  R0000004
 E  R0000005
 E  R0000006
 E  R0000007
 E  R0000008
 E  R0000009
 E  R0000010
 E  R0000011
 E  R0000012
 E  R0000013
 E  R0000014
COLUMNS
    C0000000  R0000000  1.             R0000001   -1.25
    C0000000  R0000002   -1.06
    C0000001  R0000000  1.             R0000001   -1.14
    C0000001  R0000002   -1.12
    C0000002  R0000001  1.             R0000003   -1.25
    C0000002  R0000004   -1.06
    C0000003  R0000001  1.             R0000003   -1.14
    C0000003  R0000004   -1.12
    C0000004  R0000002  1.             R0000005   -1.25
    C0000004  R0000006   -1.06
    C0000005  R0000002  1.             R0000005   -1.14
    C0000005  R0000006   -1.12
```

```
    C0000006  R0000003  1.          R0000007  1.25
    C0000006  R0000008  1.06
    C0000007  R0000003  1.          R0000007  1.14
    C0000007  R0000008  1.12
    C0000008  R0000004  1.          R0000009  1.25
    C0000008  R0000010  1.06
    C0000009  R0000004  1.          R0000009  1.14
    C0000009  R0000010  1.12
    C0000010  R0000005  1.          R0000011  1.25
    C0000010  R0000012  1.06
    C0000011  R0000005  1.          R0000011  1.14
    C0000011  R0000012  1.12
    C0000012  R0000006  1.          R0000013  1.25
    C0000012  R0000014  1.06
    C0000013  R0000006  1.          R0000013  1.14
    C0000013  R0000014  1.12
    C0000014  OBJROW    0.125       R0000007  -1.
    C0000015  OBJROW    -0.5        R0000007  1.
    C0000016  OBJROW    0.125       R0000008  -1.
    C0000017  OBJROW    -0.5        R0000008  1.
    C0000018  OBJROW    0.125       R0000009  -1.
    C0000019  OBJROW    -0.5        R0000009  1.
    C0000020  OBJROW    0.125       R0000010  -1.
    C0000021  OBJROW    -0.5        R0000010  1.
    C0000022  OBJROW    0.125       R0000011  -1.
    C0000023  OBJROW    -0.5        R0000011  1.
    C0000024  OBJROW    0.125       R0000012  -1.
    C0000025  OBJROW    -0.5        R0000012  1.
    C0000026  OBJROW    0.125       R0000013  -1.
    C0000027  OBJROW    -0.5        R0000013  1.
    C0000028  OBJROW    0.125       R0000014  -1.
    C0000029  OBJROW    -0.5        R0000014  1.
RHS
    RHS       R0000000  55.         R0000007  80.
    RHS       R0000008  80.         R0000009  80.
    RHS       R0000010  80.         R0000011  80.
    RHS       R0000012  80.         R0000013  80.
    RHS       R0000014  80.
ENDATA
```

Output.mps (8)

6 Acknowledgements

This research has been partially supported by the projects MTM2012-31514 from the Spanish Ministry of Economy and Competitiveness, Grupo de Investigación IT-567-13 from the Basque Government, UFI BETS 2011 of the University of Basque Country (UPV/EHU), Spain, and Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED 2011). Technical and human support provided by Dr. E. Ogando and Dr. T. Mercero from IZO-SGI, SGIker (UPV/EHU, MICINN,GV/EJ, ERDF and ESF) is gratefully acknowledged.

7 References

References

[Aldasoro *et al.*, 2012] U. Aldasoro, M. A. Garín, M. Merino, and G. Pérez. MPI parallel programming of mixed integer optimization problem using CPLEX with COIN-OR. Biltoki, Universidad del País Vasco - Departamento de Economía Aplicada III (Econometría y Estadística), 2012.

[Birge and Louveaux, 2011] J.R. Birge and F.V. Louveaux. *Introduction to Stochastic Programming*. 2nd edition, Springer, 2011.

[COIN-OR, 2013] COIN-OR. COmputational INfrastructure for Operations Research. Website, 2013. http://www.coin-or.org/.

[Escudero *et al.*, 2010a] L. F. Escudero, M. A. Garín, M. Merino, and G. Pérez. On BFC-MSMIP strategies for scenario cluster partitioning and Twin Node Families branching selection and bounding for multi-stage stochastic mixed integer programming. *Computers & Operations Research*, 37:738–753, 2010.

[Escudero *et al.*, 2010b] L. F. Escudero, M. A. Garín, M. Merino, and G. Pérez. A note on the implementation of the BFC-MSMIP algorithm in c++ by using COIN-OR as an optimization engine. Biltoki, Universidad del País Vasco - Departamento de Economía Aplicada III (Econometría y Estadística), 2010.

[Escudero *et al.*, 2012] L. F. Escudero, M. A. Garín, M. Merino, and G. Pérez. An algorithmic framework for solving large-scale multistage stochastic mixed 0-1 problems with nonsymmetric scenario trees. *Comput. Oper. Res.*, 39:1133–1144, May 2012.

[IBM, 2013] IBM. ILOG CPLEX optimizer. Website , 2013. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[Pérez and Garín, 2010] G. Pérez and M. A. Garín. On downloading and using COIN-OR for solving linear/integer optimization problems. Biltoki, Universidad del País Vasco - Departamento de Economía Aplicada III (Econometría y Estadística), 2010.

[Pérez and Garín, 2011] G. Pérez and M. A. Garín. On Downloading and Using CPLEX within COIN-OR for Solving Linear/Integer Optimization Problems. Biltoki, Universidad del País Vasco - Departamento de Economía Aplicada III (Econometría y Estadística), 2011.

[Rockafellar and Wets, 1991] R.T. Rockafellar and R. J-B Wets. Scenario and policy aggregation in optimisation under uncertainty. *Mathematics of Operations Research*, 16:119–147, 1991.

[Wets, 1974] R.J-B Wets. Stochastic programs with fixed recourse: The equivalent deterministic program. *SIAM Review*, 16:309–339, 1974.