DOCUMENTOS DE TRABAJO

BILTOKI

D.T. 2012.02

Scenario Cluster Lagrangian Decomposition
in two stage stochastic mixed 0-1 optimization

Laureano F. Escudero, M.Araceli Garín, Gloria Pérez y Aitziber Unzueta

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 DT2012.02
Editado por el Departamento de Economía Aplicada III (Econometría y Estadística)
de la Universidad del País Vasco.

ISSN: 1134-8984

Scenario Cluster Lagrangian Decomposition in two-stage stochastic mixed 0-1 optimization

Laureano F. Escudero[1], M. Araceli Garín[2], Gloria Pérez[3] and Aitziber Unzueta[4]

[1] Dpto. Estadística e Investigación Operativa. Universidad Rey Juan Carlos, Spain. laureano.escudero@urjc.es

[2] Dpto. de Economía Aplicada III. Universidad del País Vasco, Spain. mariaaraceli.garin@ehu.es

[3] Dpto. de Matemática Aplicada y Estadística e I.O. Universidad del País Vasco, Spain. gloria.perez@ehu.es

[4] Dpto. de Economía Aplicada III. Universidad del País Vasco, Spain. aitziber.unzueta@ehu.es

March 15, 2012

Abstract

In this paper we introduce four scenario Cluster based Lagrangian Decomposition (CLD) procedures for obtaining strong lower bounds to the (optimal) solution value of two-stage stochastic mixed 0-1 problems. At each iteration of the Lagrangian based procedures, the traditional aim consists of obtaining the solution value of the corresponding Lagrangian dual via solving scenario submodels once the nonanticipativity constraints have been dualized. Instead of considering a splitting variable representation over the set of scenarios, we propose to decompose the model into a set of scenario clusters. We compare the computational performance of the four Lagrange multiplier updating procedures, namely the Subgradient Method, the Volume Algorithm, the Progressive Hedging Algorithm and the Dynamic Constrained Cutting Plane scheme for different numbers of scenario clusters and different dimensions of the original problem. Our computational experience shows that the CLD bound and its computational effort depend on the number of scenario clusters to consider. In any case, our results show that the CLD procedures outperform the traditional LD scheme for single scenarios both in the quality of the bounds and computational effort. All the procedures have been implemented in a C++ experimental code. A broad computational experience is reported on a test of randomly generated instances by using the MIP solvers COIN-OR [17] and CPLEX [16] for the auxiliary mixed 0-1 cluster submodels, this last solver within the open source engine COIN-OR. We also give computational evidence of the model tightening effect that the preprocessing techniques, cut generation and appending and parallel computing tools have in stochastic integer optimization. Finally, we have observed that the plain use of both solvers does not provide the optimal solution of the instances included in the testbed with which we have experimented but for two toy instances in affordable elapsed time. On the other hand the proposed procedures provide strong lower bounds (or the same solution value) in a considerably shorter elapsed time for the quasi-optimal solution obtained by other means for the original stochastic problem.

Keywords: Two-stage stochastic integer programming, nonanticipativity constraints, Cluster Lagrangian decomposition, scenario cluster model, Subgradient Method, Volume Algorithm, Progressive Hedging Algorithm, Dynamic Constrained Cutting Plane scheme.

1 Introduction

In this work we consider a general two-stage stochastic mixed 0-1 problem. The uncertainty is modeled via a finite set of scenarios $\omega = 1, ..., |\Omega|$, each with an associated probability of occurrence w^{ω}, $\omega \in \Omega$. The traditional aim in this type of problems is to solve the so-called Deterministic Equivalent Model (DEM), which is a mixed 0-1 problem with a special structure, see e.g., [21] for a good survey of some mayor results in this area obtained during the 90s and beyond. A Branch-and-Bound algorithm for solving problems having mixed-integer variables in both stages is designed in [6], among others, by using Lagrangian relaxation for obtaining lower bounds to the optimal solution of the original problem. A Branch-and-Fix Coordination (BFC) methodology for solving such DEM in production planning under uncertainty is given in [1, 2], but the approach does not allow continuous first stage variables or 0-1 second stage variables. We propose in [7, 8] a BFC algorithmic framework for obtaining the optimal solution of the two-stage stochastic mixed 0-1 integer problem, where the uncertainty appears anywhere in the coefficients of the 0-1 and continuous variables in both stages. Recently, a general algorithm for two-stage problems has been presented in [22].

We study in [10] several solution methods for solving the dual problem corresponding to the Lagrangian Decomposition (LD) of two-stage stochastic mixed 0-1 models. At each iteration of these Lagrangian based procedures, the traditional aim consists of obtaining the solution value of the corresponding parametric mixed 0-1 Lagrangian dual problem via solving single scenario submodels once the nonanticipativity constraints (NAC) have been dualized, and the parameters (i.e., the Lagrange multipliers) are updated by using different subgradient and cutting plane based methodologies.

Instead of considering a splitting variable representation over the set of scenarios, in this paper we propose a new approach so named Cluster Lagrangian Decomposition (for short, CLD) to decompose the model into a set of scenario clusters. So, we computationally compare the performance of the Subgradient Method (SM) [15], the Volume Algorithm (VA) [4], the Progressive Hedging Algorithm (PHA) [20] and the Dynamic Constrained Cutting Plane (DCCP) scheme [18] for Lagrange multipliers updating while solving large-scale stochastic mixed 0-1 problems in an algorithmic framework based on scenario clusters decomposition. A successful result may open up the possibility for tightening the lower bounds of the solution value at the candidate Twin Node Families in the exact BFC scheme for both two-stage and multistage types of problems, see e.g., [9].

For different choices of the number of scenario clusters we report the computational experience by using CPLEX, integrated in the COIN-OR environment, to verify the effectiveness of the proposal. In this sense, we also give computational evidence of the model tightening effect and their computational cost that preprocessing, cut generation and appending and parallel computing tools have in stochastic integer optimization too, see [19]. We also computationally compare the new with the cluster singleton approach (i.e., the LD for single scenarios) outperforming it, as well as outperforming the plain use of the MIP solver of choice, CPLEX. The proposed approach provides a tight lower bound such that the quasi-optimality gap of the upper solution bound obtained by other means on large-scale instances is very small and frequently, guarantees its optimality. However, the plain use of CPLEX cannot guarantee the optimality of the incumbent solution in a somewhat large elapsed time limit, its objective function value being simply an upper bound of the solution

value of the original stochastic problem in some cases. In other cases, we can prove in very much smaller elapsed time that the incumbent CPLEX solution is the optimal one, since our CLD procedures provide lower bounds identical to the value of that solution. Additionally, that incumbent solution is also frequently even worse than that which we have obtained when both the quality and the small elapsed time are good enough.

The remainder of the paper is organized as follows: Section 2 presents the two-stage stochastic mixed 0-1 problem in compact and splitting variable representations over the scenarios and scenario clusters. Section 3 summarizes the theoretical results on Lagrangian decomposition and presents the Cluster Lagrangian Decomposition approach. Section 4 presents the four procedures mentioned above for updating the Lagrange multipliers. Section 5 reports the results of the computational experiment. Section 6 concludes.

2 Two-stage stochastic mixed 0-1 problem

In many real cases a two-stage deterministic mixed 0-1 optimization model must be extended to consider the uncertainty in some of the main parameters. In our case, these are the objective function, the right and left hand-side vectors and the constraint matrix coefficients. This uncertainty is introduced by using the scenario analysis approach, such that a scenario consists of a realization of all random parameters in both stages through a scenario tree. When a finite number of scenarios is considered, a general two-stage program can be expressed in terms of the first stage decision variables being equivalent to a large, dual block-angular programming problem, introduced in [25] and known as Deterministic Equivalent Model (DEM).

Let us consider the *compact* representation of the DEM of a two-stage stochastic integer problem (MIP),

$$
\begin{array}{lll}
(MIP)^c: \; z_{MIP} = \min & c_1\delta + c_2 x + \sum\limits_{\omega\in\Omega} w^\omega [q_1^\omega \gamma^\omega + q_2^\omega y^\omega] & \\
\text{s.t.} & b_1 \leq A \begin{pmatrix} \delta \\ x \end{pmatrix} \leq b_2 & \\
& h_1^\omega \leq T^\omega \begin{pmatrix} \delta \\ x \end{pmatrix} + W^\omega \begin{pmatrix} \gamma^\omega \\ y^\omega \end{pmatrix} \leq h_2^\omega, \quad \forall \omega \in \Omega & \\
& \delta, \gamma^\omega \in \{0,1\}, x, y^\omega \geq 0, \quad \forall \omega \in \Omega, &
\end{array}
\tag{1}
$$

where the uncertainty in the parameters is introduced by using a scenario analysis approach. c_1 and c_2 are known vectors of the objective function coefficients for the δ and x variables in the first stage, respectively, b_1 and b_2 are the known left and right hand side vectors for the first stage constraints, respectively, and A is the known matrix of coefficients for the first stage constraints. For each scenario ω, w^ω is the likelihood attributed to the scenario, such that $\sum_{\omega\in\Omega} w^\omega = 1$, h_1^ω and h_2^ω are the left and right hand side vectors for the second stage constraints, respectively, and q_1^ω and q_2^ω are the objective function coefficients for the second stage γ and y variables, respectively, while T^ω and W^ω are the technology constraint matrices under scenario ω, for $\omega \in \Omega$, where Ω is the set of scenarios to consider. Notice that there are two types of decision variables at each stage, namely, the set of δ 0-1 and x continuous variables for the first stage, and the set of γ^ω 0-1 and y^ω continuous variables for the second stage.

Notice also that for the purpose of simplification, the objective function to optimize in the models dealt with in this paper is the expected value over the set of scenarios Ω, i.e., the risk neutral strategy. For a survey of coherent risk averse measures as opposed to the risk neutral strategy considered in this work, see e.g., [3].

The structure of the uncertain information can be visualized as a tree, where each root-to-leaf path represents one specific scenario, ω, and corresponds to one realization of the whole set of the uncertain parameters. In the example depicted in Figure 1, there are $|\Omega| = 10$ root-to-leaf possible paths, i.e., scenarios. Following the nonanticipativity principle, stated in [25] and restated in [20], see [5] among others, all scenarios should have the same value for the related first stage variables in the two-stage problem.

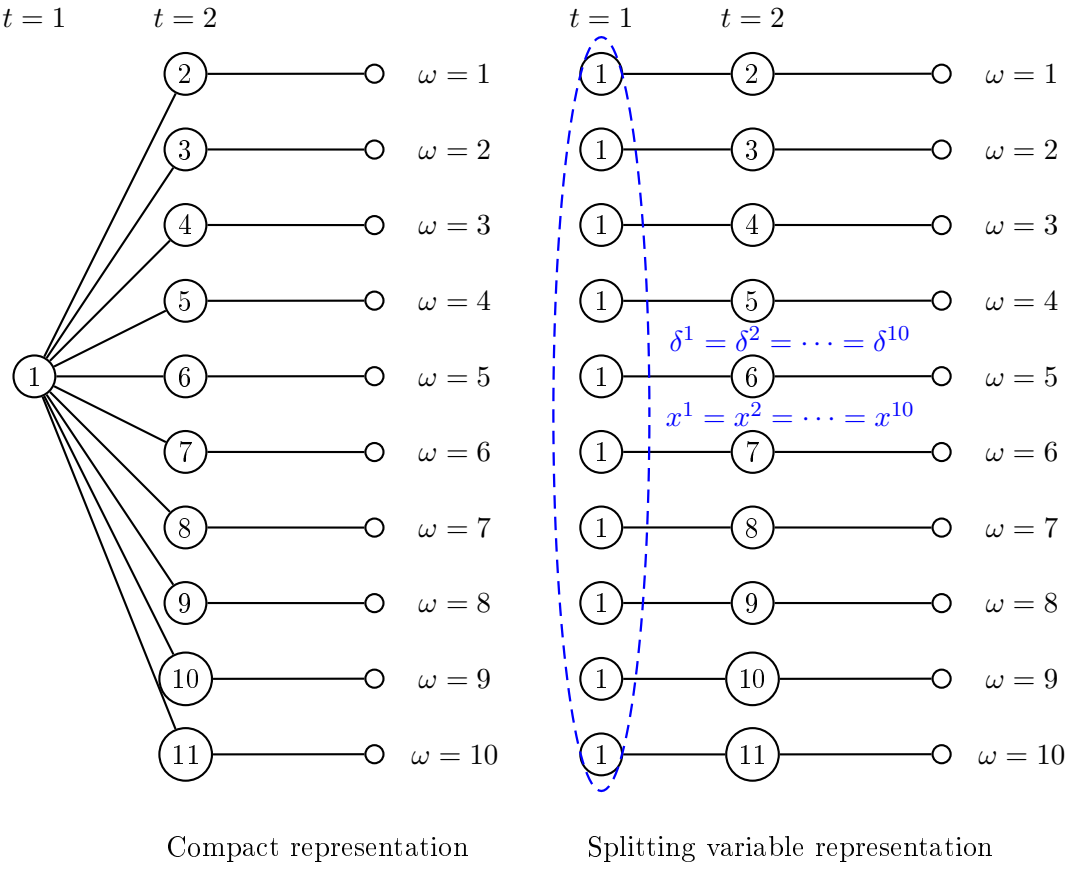

Figure 1: Scenario tree

The left section of Figure 1 implicitly represents the non-anticipativity constraints (NAC, for short). This is the compact representation shown in model (1). The right section of Figure 1 gives the same information as the compact representation but using a splitting variable scheme and noticing that it explicitly represents the NAC (i.e., imposing the equality) on the first stage variables δ^ω x^ω and for all the scenarios ω.

Let us consider the *splitting variable* representation of the DEM of the two-stage stochastic

mixed 0-1 problem.

$$
\begin{aligned}
(MIP)^s: \quad & z_{MIP} = \min \sum_{\omega \in \Omega} w^\omega [c_1 \delta^\omega + c_2 x^\omega + q_1^\omega \gamma^\omega + q_2^\omega y^\omega] \\
\text{s.t.} \quad & b_1 \leq A \begin{pmatrix} \delta^\omega \\ x^\omega \end{pmatrix} \leq b_2 \quad \forall \omega \in \Omega \\
& h_1^\omega \leq T^\omega \begin{pmatrix} \delta^\omega \\ x^\omega \end{pmatrix} + W^\omega \begin{pmatrix} \gamma^\omega \\ y^\omega \end{pmatrix} \leq h_2^\omega \quad \forall \omega \in \Omega \\
& \delta^\omega = \delta^{\omega'} \quad \forall \omega, \omega' \in \Omega, \quad \omega \neq \omega' \\
& x^\omega = x^{\omega'} \quad \forall \omega, \omega' \in \Omega, \quad \omega \neq \omega' \\
& x^\omega, y^\omega \geq 0 \quad \forall \omega \in \Omega \\
& \delta^\omega, \gamma^\omega \in \{0,1\} \quad \forall \omega \in \Omega.
\end{aligned} \tag{2}
$$

In addition to these two formulations, we propose a scenario-cluster partitioning to allow a combination of compact and splitting variable representations into and inter the scenario cluster submodels. A scenario *cluster* is a set of scenarios where the NAC are implicitly considered. By slightly abusing the notation from now on, throughout the paper the upperindex in boldface $\mathbf{p}$ will denote the cluster of scenarios instead of the single one. Let $\hat{\mathbf{p}}$ denote the number of scenario cluster partitions to consider. As an illustrative example, let us consider again the scenario tree depicted in Figure 1.

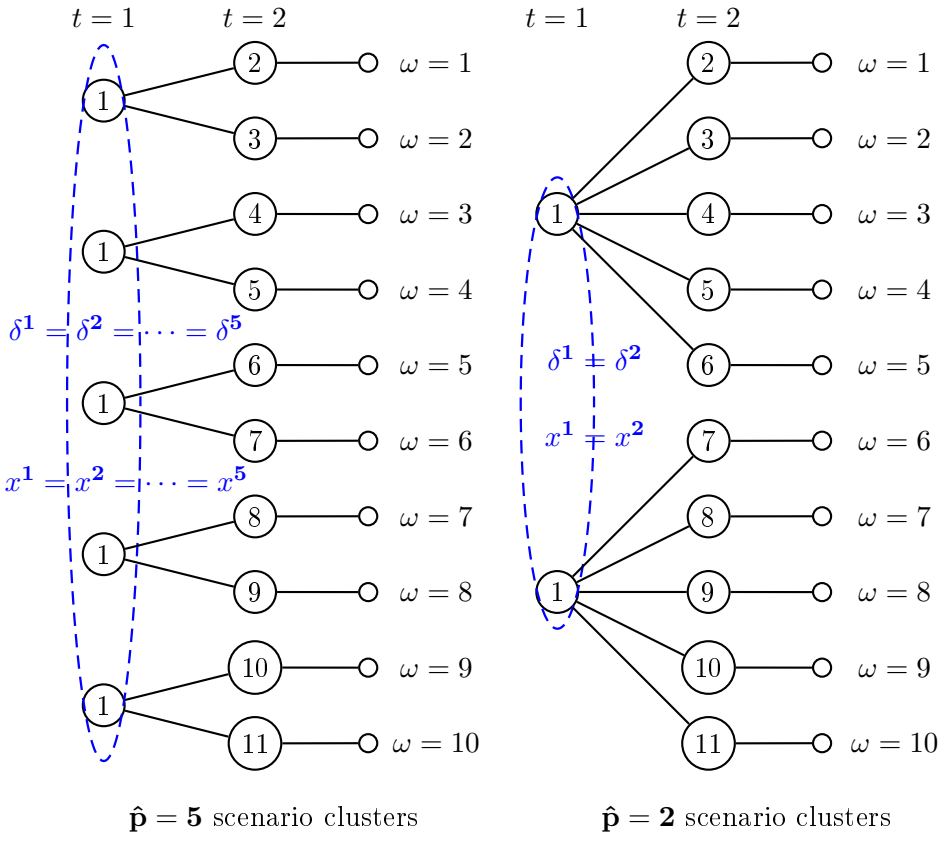

Figure 2: Scenario cluster partitioning

Figure 2 shows the problem decomposition in $\hat{\mathbf{p}} = \mathbf{5}$ (left tree) and $\hat{\mathbf{p}} = \mathbf{2}$ (right tree) scenario clusters into which the set of scenarios is split. Observe that the NAC for the first stage vectors of variables are given by $x^{\mathbf{1}} = \cdots = x^{\mathbf{5}}$ and $\delta^{\mathbf{1}} = \cdots = \delta^{\mathbf{5}}$ for the left side of the figure, and they are given by $x^{\mathbf{1}} = x^{\mathbf{2}}$ and $\delta^{\mathbf{1}} = \delta^{\mathbf{2}}$ for the right side of the figure, where by abusing the notation $x^{\mathbf{p}}$ and $\delta^{\mathbf{p}}$ are the x and δ vectors of the first stage continuous and 0-1 variables for scenario cluster $\mathbf{p}$, respectively.

In general, given a scenario tree, $\hat{\mathbf{p}}$ can be chosen as any value between 1 and $|\Omega|$, so that we can represent the DEM (1) by a mixture of the splitting variable representation where the NAC are explicitly stated for the $\hat{\mathbf{p}}$ cluster submodels, and a compact representation for the set $\Omega^{\mathbf{p}}$ of scenarios into each cluster model $\mathbf{p}$, where the NAC are implicitly stated such that $p \in \{1, ..., \hat{\mathbf{p}}\}$, $\Omega^{\mathbf{p}}$ defines the set of scenarios in cluster $\mathbf{p}$, and $|\Omega^{\mathbf{p}}|$ is its size.

Without loss of generality (wlog, for short) and for computational purposes, the number of clusters $\hat{\mathbf{p}}$ can be calculated as a divisor of the number of scenarios, $|\Omega|$ and, then, we have that $l = |\Omega^{\mathbf{p}}| = \frac{|\Omega|}{\hat{\mathbf{p}}}$ defines the size of each scenario cluster $\mathbf{p}$, for $\mathbf{p} = \mathbf{1}, ..., \hat{\mathbf{p}}$. The scenario clusters are defined in terms of consecutive scenarios, $\Omega^{\mathbf{1}} = \{1, ..., |\Omega^1|\}$, $\Omega^{\mathbf{2}} = \{|\Omega^1| + 1, ..., |\Omega^1| + |\Omega^2|\}$,..., $\Omega^{\hat{\mathbf{p}}} = \{|\Omega^1| + ... + |\Omega^{\hat{\mathbf{p}}-\mathbf{1}}| + 1, ..., |\Omega|\}$.

The mixed 0-1 submodel to consider for each scenario cluster $\mathbf{p}$ can be expressed by the *compact* representation,

$$
\begin{array}{lll}
(MIP^{\mathbf{p}}): \ z^{\mathbf{p}} = \min & \mathbf{w}^{\mathbf{p}}(c_1\delta^{\mathbf{p}} + c_2 x^{\mathbf{p}}) + \displaystyle\sum_{\omega \in \Omega^{\mathbf{p}}} w^{\omega}(q_1^{\omega}\gamma^{\omega} + q_2^{\omega} y^{\omega}) & \\
\text{s.t.} & b_1 \leq A \begin{pmatrix} \delta^{\mathbf{p}} \\ x^{\mathbf{p}} \end{pmatrix} \leq b_2 & \\
& h_1^{\omega} \leq T^{\omega} \begin{pmatrix} \delta^{\mathbf{p}} \\ x^{\mathbf{p}} \end{pmatrix} + W^{\omega} \begin{pmatrix} \gamma^{\omega} \\ y^{\omega} \end{pmatrix} \leq h_2^{\omega} \quad \forall \omega \in \Omega^{\mathbf{p}} & \\
& \delta^{\mathbf{p}} \in \{0,1\}, x^{\mathbf{p}} \geq 0 & \\
& \gamma^{\omega} \in \{0,1\}, y^{\omega} \geq 0 \quad \forall \omega \in \Omega^{\mathbf{p}}, &
\end{array}
\tag{3}
$$

where $\mathbf{w}^{\mathbf{p}} = \sum_{\omega \in \Omega^{\mathbf{p}}} w^{\omega}$ denotes the likelihood for scenario cluster $\mathbf{p}$, and $\delta^{\mathbf{p}}$ and $x^{\mathbf{p}}$ are the vectors of the first stage δ and x variables for scenario cluster $\mathbf{p}$. Moreover, the $\hat{\mathbf{p}}$ submodels (3) are linked by the NAC,

$$
\delta^{\mathbf{p}} - \delta^{\mathbf{p}'} = 0 \tag{4}
$$

$$
x^{\mathbf{p}} - x^{\mathbf{p}'} = 0, \tag{5}
$$

for $\mathbf{p}, \mathbf{p}' = \mathbf{1}, \ldots, \hat{\mathbf{p}} : \mathbf{p} \neq \mathbf{p}'$. Observe that the NAC (4)-(5) can been represented as a set of inequalities in order to avoid the use of non-signed vectors of Lagrange multipliers in the dualization of such constraints. They will be expressed as follows,

$$
\delta^{\mathbf{p}} - \delta^{\mathbf{p}+\mathbf{1}} \leq 0 \quad \forall \mathbf{p} = \mathbf{1}, ..., \hat{\mathbf{p}} - \mathbf{1}, \quad \delta^{\hat{\mathbf{p}}} \leq \delta^{\mathbf{1}}, \tag{6}
$$

$$
x^{\mathbf{p}} - x^{\mathbf{p}+\mathbf{1}} \leq 0 \quad \forall \mathbf{p} = \mathbf{1}, ..., \hat{\mathbf{p}} - \mathbf{1}, \quad x^{\hat{\mathbf{p}}} \leq x^{\mathbf{1}} \tag{7}
$$

So, the mixed 0-1 DEM (1) is equivalent to the splitting-compact variable representation

over the set of scenario clusters.

$$
\begin{array}{lll}
(MIP): & z_{MIP} = \min \displaystyle\sum_{\mathbf{p}=1}^{\hat{\mathbf{p}}} [\mathbf{w^p}(c_1\delta^{\mathbf{p}} + c_2 x^{\mathbf{p}}) + \sum_{\omega\in\Omega^{\mathbf{p}}} w^\omega(q_1^\omega\gamma^\omega + q_2^\omega y^\omega)] & \\
\text{s.t.} & b_1 \le A\begin{pmatrix}\delta^{\mathbf{p}}\\ x^{\mathbf{p}}\end{pmatrix} \le b_2 \quad \forall \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}} & \\
& h_1^\omega \le T^\omega\begin{pmatrix}\delta^{\mathbf{p}}\\ x^{\mathbf{p}}\end{pmatrix} + W^\omega\begin{pmatrix}\gamma^\omega\\ y^\omega\end{pmatrix} \le h_2^\omega \quad \forall \omega\in\Omega^{\mathbf{p}}, \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}} & \\
& \delta^{\mathbf{p}} - \delta^{\mathbf{p+1}} \le 0 \quad \forall \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}}-\mathbf{1} & \\
& \delta^{\hat{\mathbf{p}}} \le \delta^{\mathbf{1}}, & \\
& x^{\mathbf{p}} - x^{\mathbf{p+1}} \le 0 \quad \forall \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}}-\mathbf{1}, & \\
& x^{\hat{\mathbf{p}}} \le x^{\mathbf{1}} & \\
& x^{\mathbf{p}} \ge 0, \delta^{\mathbf{p}} \in \{0,1\} \quad \forall \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}} & \\
& y^\omega \ge 0, \gamma^\omega \in \{0,1\} \quad \forall \omega\in\Omega^{\mathbf{p}}, \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}}. &
\end{array}
\tag{8}
$$

Additionally, notice that model (8) for $\hat{\mathbf{p}} = \mathbf{1}$ coincides with the mixed 0-1 DEM in the compact representation (1), and we obtain the splitting variable representation (2) for $\hat{\mathbf{p}} = |\mathbf{\Omega}|$.

3 Scenario Cluster Lagrangian Decomposition

The scenario Cluster Lagrangian Decomposition (CLD) of the mixed 0-1 DEM (8) for a given number of scenario clusters $\hat{\mathbf{p}}$ and a given nonnegative vector of weights (i.e., Lagrange multipliers) $\mu^{\mathbf{p}} = (\mu_\delta^{\mathbf{p}}, \mu_x^{\mathbf{p}})$, is the μ-parametric mixed 0-1 minimization model (9) in $(\delta^{\mathbf{p}}, x^{\mathbf{p}}, \gamma^\omega, y^\omega)$, $\omega\in\Omega^{\mathbf{p}}$, $\mathbf{p}=\mathbf{1},\cdots,\hat{\mathbf{p}}$, with the objective function value $z_{LD}(\mu,\hat{\mathbf{p}})$, such that it can be expressed as follows,

$$
\begin{array}{lll}
(MIP_{LD}^{\hat{\mathbf{p}}}(\mu)): & z_{LD}(\mu,\hat{\mathbf{p}}) = & \min \displaystyle\sum_{\mathbf{p}=\mathbf{1}}^{\hat{\mathbf{p}}} [\mathbf{w^p}(c_1\delta^{\mathbf{p}} + c_2 \mathbf{x^p}) + \sum_{\omega\in\Omega^{\mathbf{p}}} w^\omega(q_1^\omega\gamma^\omega + q_2^\omega y^\omega)] \\
& & + \displaystyle\sum_{\mathbf{p}=\mathbf{1}}^{\hat{\mathbf{p}}-\mathbf{1}} \mu_\delta^{\mathbf{p}}(\delta^{\mathbf{p}} - \delta^{\mathbf{p+1}}) + \mu_\delta^{\hat{\mathbf{p}}}(\delta^{\hat{\mathbf{p}}} - \delta^{\mathbf{1}}) + \\
& & + \displaystyle\sum_{\mathbf{p}=\mathbf{1}}^{\hat{\mathbf{p}}-\mathbf{1}} \mu_x^{\mathbf{p}}(x^{\mathbf{p}} - x^{\mathbf{p+1}}) + \mu_x^{\hat{\mathbf{p}}}(x^{\hat{\mathbf{p}}} - x^{\mathbf{1}}) \\
& \text{s.t.} \quad b_1 \le & A\begin{pmatrix}\delta^{\mathbf{p}}\\ x^{\mathbf{p}}\end{pmatrix} \le b_2 \quad \forall \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}} \\
& h_1^\omega \le & T^\omega\begin{pmatrix}\delta^{\mathbf{p}}\\ x^{\mathbf{p}}\end{pmatrix} + W^\omega\begin{pmatrix}\gamma^\omega\\ y^\omega\end{pmatrix} \le h_2^\omega \quad \forall \omega\in\Omega^{\mathbf{p}}, \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}} \\
& & x^{\mathbf{p}} \ge 0, \delta^{\mathbf{p}} \in \{0,1\} \quad \forall \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}} \\
& & y^\omega \ge 0, \gamma^\omega \in \{0,1\} \quad \forall \omega\in\Omega^{\mathbf{p}}, \mathbf{p}=\mathbf{1},...,\hat{\mathbf{p}}.
\end{array}
\tag{9}
$$

It is well known that model $(MIP_{LD}^{\hat{\mathbf{p}}}(\mu))$ is a relaxation of model (MIP), since (i) the feasible set of $(MIP_{LD}^{\hat{\mathbf{p}}}(\mu))$ contains the feasible set of (MIP), and (ii) for any (δ, x, γ, y) feasible solution for (MIP), any $\mu \ge 0$ and $\mathbf{1} < \hat{\mathbf{p}} \le |\Omega|$, it results that $z_{LD}(\mu,\hat{\mathbf{p}}) \le z_{MIP}$. Notice that if $\hat{\mathbf{p}} = \mathbf{1}$, for any $\mu \ge 0$ $z_{LD}(\mu,\mathbf{1}) = z_{MIP}$ by definition of the compact representation. Then, it follows that the value $z_{LD}(\mu,\hat{\mathbf{p}})$, which depends on μ is, a lower bound on the solution value of (MIP), z_{MIP} for any choice of $\hat{\mathbf{p}}$, with $\mathbf{1} < \hat{\mathbf{p}} \le |\Omega|$.

Definition 1 *For any choice of* $\hat{\mathbf{p}}$ *such that* $\mathbf{1} < \hat{\mathbf{p}} \leq |\Omega|$*, the problem of finding the tightest Lagrangian lower bound on* z_{MIP} *is*

$$(MIP_{LD}): \quad z_{LD} = max_{\mu \geq 0} z_{LD}(\mu, \hat{\mathbf{p}}).$$

It is called the Lagrangian dual of (MIP) *relative to the NAC.*

By LP duality, z_{LD} can be obtained by using a mixture of linear and mixed 0-1 programs. (MIP_{LD}) is a linear problem in the dual space of the Lagrange multipliers, whereas $(MIP_{LD}^{\hat{\mathbf{p}}}(\mu))$ is a μ-parametric mixed 0-1 problem in the vector of variables (δ, x, γ, y). Let $(\delta(\mu^{\hat{\mathbf{p}}}), x(\mu^{\hat{\mathbf{p}}}), \gamma(\mu^{\hat{\mathbf{p}}}), y(\mu^{\hat{\mathbf{p}}}))$ denote an optimal solution of $(MIP_{LD}^{\hat{\mathbf{p}}}(\mu))$ for some μ and $\hat{\mathbf{p}}$, i.e., a Lagrangian solution.

It is also known that, unless (MIP_{LD}) does have the integrality property, the LD can yield an equal or stronger bound than the LP relaxation. If it has the integrality property then $z_{LP} = z_{LD} \leq z_{MIP}$. In the other case, $z_{LP} \leq z_{LD} \leq z_{MIP}$. See the seminal work [12], and a good survey in [13].

Let the following proposition state that the solution values of nonsingleton scenario cluster Lagrangian decomposition (CLD) problems are stronger than the solution values of singleton CLD problems.

Proposition 1 *For all* $\mu \geq 0$*, the following inequalities are satisfied*

$$z_{LD}(\mu, |\Omega|) \leq z_{LD}(\mu, |\Omega| - 1) \leq .... \leq z_{LD}(\mu, 2) \leq z_{LD}(\mu, 1) = z_{MIP}.$$

Proof: Notice that the chain of the related problems only differ on the relaxation of the NAC in some scenarios. So the proof follows.

Our proposal makes use of the expression of the Lagrangian dual z_{LD} as the maximum of the solution values $z_{LD}(\mu, \hat{\mathbf{p}})$ in μ. Previously, we must choose a number of scenario clusters $\hat{\mathbf{p}}$ and the scenario subsets Ω^p, $p = 1, ..., \hat{\mathbf{p}}$ and then, for a given value of μ, say $\mu^{\hat{\mathbf{p}}}$, we must solve the mixed 0-1 problem (9) in $(\delta(\mu^{\hat{\mathbf{p}}}), x(\mu^{\hat{\mathbf{p}}}), \gamma(\mu^{\hat{\mathbf{p}}}), y(\mu^{\hat{\mathbf{p}}}))$ to obtain the optimal solution value, $z_{LD}(\mu^{\hat{\mathbf{p}}}, \hat{\mathbf{p}})$. It consists of computationally comparing the speed of convergence with several iterative methods for updating the Lagrange multipliers and building the sequence $\{\mu^0, \mu^1, ..., \mu^k, ...\}^{\hat{\mathbf{p}}}$, as well as studying the optimal scenario cluster decomposition.

At each iteration k and given the current multiplier vector μ^k, the first step is to obtain $z_{LD}(\mu^k, \hat{\mathbf{p}})$. The second step is to update the Lagrange multipliers μ in a finite number of iterations such that the purpose is to obtain μ^* and $z_{LD}(\mu^*, \hat{\mathbf{p}})$, where

$$\mu^* \in \text{argmax}_{\mu \geq 0}\{z_{LD}(\mu, \hat{\mathbf{p}})\}. \tag{10}$$

Note: The solution $(\delta(\mu^*), x(\mu^*), \gamma(\mu^*), y(\mu^*))$ is the optimal one for DEM (1) provided that it satisfies the NAC (6)-(7).

Notice that the model $MIP_{LD}^{\hat{\mathbf{p}}}(\mu)$ (9) can be decomposed in $\hat{\mathbf{p}}$ smaller submodels, and its solution value can be obtained as the sum of the related $z_{LD}^{\mathbf{p}}(\mu^{\mathbf{p}})$ values, see [10],

$$z_{LD}(\mu, \hat{\mathbf{p}}) = \sum_{\mathbf{p=1}}^{\hat{\mathbf{p}}} z_{LD}^{\mathbf{p}}(\mu^{\mathbf{p}}), \tag{11}$$

where $z^{\mathbf{p}}_{LD}(\mu^{\mathbf{p}})$ is the solution value of the $\mathbf{p}$th scenario cluster model. For $\mathbf{p} = \mathbf{2}, ..., \hat{\mathbf{p}}$, the model is expressed in compact representation as follows,

$$
\begin{array}{rcl}
z^{\mathbf{p}}_{LD}(\mu^{\mathbf{p}}) & = & \min[\mathbf{w^p}c_1 + (\mu^{\mathbf{p}}_{\delta} - \mu^{\mathbf{p-1}}_{\delta})]\delta^{\mathbf{p}} + [\mathbf{w^p}c_2 + (\mu^{\mathbf{p}}_{x} - \mu^{\mathbf{p-1}}_{x})]x^{\mathbf{p}} + \sum_{\omega \in \Omega^{\mathbf{p}}} w^{\omega}(q^{\omega}_1\gamma^{\omega} + q^{\omega}_2 y^{\omega}) \\
\text{s.t} & & \\
& & b_1 \leq A \begin{pmatrix} \delta^{\mathbf{p}} \\ x^{\mathbf{p}} \end{pmatrix} \leq b_2 \\
& & h^{\omega}_1 \leq T^{\omega} \begin{pmatrix} \delta^{\mathbf{p}} \\ x^{\mathbf{p}} \end{pmatrix} + W^{\omega} \begin{pmatrix} \gamma^{\omega} \\ y^{\omega} \end{pmatrix} \leq h^{\omega}_2 \quad \forall \omega \in \Omega^{\mathbf{p}} \\
& & x^{\mathbf{p}} \geq 0, \delta^{\mathbf{p}} \in \{0,1\} \\
& & y^{\omega} \geq 0, \gamma^{\omega} \in \{0,1\} \quad \forall \omega \in \Omega^{\mathbf{p}}.
\end{array}
\tag{12}
$$

For $\mathbf{p} = \mathbf{1}$, the model also in compact representation is as follows,

$$
\begin{array}{rcl}
z^{\mathbf{1}}_{LD}(\mu^{\mathbf{1}}) & = & \min[\mathbf{w^1}c_1 + (\mu^{\mathbf{1}}_{\delta} - \mu^{\hat{\mathbf{p}}}_{\delta})]\delta^{\mathbf{1}} + [\mathbf{w^1}c_2 + (\mu^{\mathbf{1}}_{x} - \mu^{\hat{\mathbf{p}}}_{x})]x^{\mathbf{1}} + \sum_{\omega \in \Omega^{\mathbf{1}}} w^{\omega}(q^{\omega}_1\gamma^{\omega} + q^{\omega}_2 y^{\omega}) \\
\text{s.t.} & & \\
& & b_1 \leq A \begin{pmatrix} \delta^{\mathbf{1}} \\ x^{\mathbf{1}} \end{pmatrix} \leq b_2 \\
& & h^{\omega}_1 \leq T^{\omega} \begin{pmatrix} \delta^{\mathbf{1}} \\ x^{\mathbf{1}} \end{pmatrix} + W^{\omega} \begin{pmatrix} \gamma^{\omega} \\ y^{\omega} \end{pmatrix} \leq h^{\omega}_2 \quad \forall \omega \in \Omega^{\mathbf{1}} \\
& & x^{\mathbf{1}} \geq 0, \delta^{\mathbf{1}} \in \{0,1\} \\
& & y^{\omega} \geq 0, \gamma^{\omega} \in \{0,1\} \quad \forall \omega \in \Omega^{\mathbf{1}}.
\end{array}
\tag{13}
$$

Observe in expression (11) that the bound value and the computational effort to compute it depend on how many scenario cluster submodels are considered in the decomposition, i.e., $\hat{\mathbf{p}}$. We computationally study in Section 5 the influence of the number of scenario clusters into the bounds tightening and the related computational effort to compute the bounds.

4 Lagrange multipliers updating procedures for CLD

In this section the specialization of different Lagrange multiplier procedures for scenario cluster decomposition is presented.

Let us assume in the rest of the work that the scenario set is broken down into $\hat{\mathbf{p}}$ clusters. Let also $\overline{z}_{LD}$ be an upper bound of the solution value of the original (MIP). It can be obtained efficiently as a quasioptimal solution, $\overline{z}(\rho)$ with a given $\rho\%$ of quasi-optimality tolerance, see Section 5. Let μ^0 be the initial multiplier vector and, finally, let α_k be a real parameter related to the steplength of the Lagrange multiplier updating procedure, where $\alpha_k \in (0, 2)$, see below.

4.1 Subgradient method

This is one of the most popular approaches to solve the Lagrangian dual. The subgradient procedure was proposed in [15]. It is an iterative approach method in which at iteration k,

given the current multipliers vector μ^k, a step is taken along a subgradient of $z_{LD}(\mu^k, \hat{\mathbf{p}})$. The procedure for updating the Lagrange multipliers of the NAC (6)-(7) is given in Figure 3.

Step 0: We start with a vector μ^0, and solve the $\hat{\mathbf{p}}$ submodels (12)-(13) to obtain $(\delta^{(0)}, x^{(0)}, \gamma^{(0)}, y^{(0)})$ and $z_{LD}(\mu^0, \hat{\mathbf{p}})$ as the sum given in (11). Set $k := 0$.

Step 1: Compute the step direction $s^k = \begin{pmatrix} (\delta^{(k)\mathbf{1}} - \delta^{(k)\mathbf{2}}) \\ \vdots \\ (\delta^{(k)\hat{\mathbf{p}}-\mathbf{1}} - \delta^{(k)\hat{\mathbf{p}}}) \\ (\delta^{(k)\hat{\mathbf{p}}} - \delta^{(k)\mathbf{1}}) \\ (x^{(k)\mathbf{1}} - x^{(k)\mathbf{2}}) \\ \vdots \\ (x^{(k)\hat{\mathbf{p}}-\mathbf{1}} - x^{(k)\hat{\mathbf{p}}}) \\ (x^{(k)\hat{\mathbf{p}}} - x^{(k)\mathbf{1}}) \end{pmatrix}$,
check the stopping criteria given in Sec. 4.5 and if they are not satisfied, set $\mu^{k+1} := \mu^k + \alpha_k \cdot \frac{(\overline{z}_{LD} - z_{LD}(\mu^k, \hat{\mathbf{p}}))}{||s^k||^2} \cdot s^k$.
Solve the $\hat{\mathbf{p}}$ problems (12)-(13) with μ^{k+1}, and let $(\delta^{(k+1)}, x^{(k+1)}, \gamma^{(k+1)}, y^{(k+1)})$ and $z_{LD}(\mu^{k+1}, \hat{\mathbf{p}})$ be the optimal solution and solution value of (9), respectively. Set $k := k + 1$ and go to Step 1.

Figure 3: Subgradient Method (SM)

4.2 Volume Algorithm

We present a version of the Volume Algorithm given in [4] for updating the Lagrange multipliers of the NAC (6)-(7). This procedure only updates the multipliers when there is an improvement in the incumbent solution value $z_{LD}(\mu, \hat{\mathbf{p}})$ of the Lagrangian problem. Additionally, the feasible solution is replaced by a convex combination of solutions obtained in previous iterations. Let f_k be a real parameter related to the incumbent solution updating, where $f_k \in (0, 1)$, see in Sec. 4.6 the procedure for obtaining it. The procedure for updating the Lagrange multipliers of the NAC (6)-(7) is given in Figure 4.

Step 0: We start with a multiplier vector μ^0, and solve the $\hat{\mathbf{p}}$ problems (12)-(13) to obtain $(\delta^{(0)}, x^{(0)}, \gamma^{(0)}, y^{(0)})$ and $z_{LD}(\mu^0, \hat{\mathbf{p}})$ as the sum given in (11). Set: $(\overline{\delta}, \overline{x}, \overline{\gamma}, \overline{y}) := (\delta^{(0)}, x^{(0)}, \gamma^{(0)}, y^{(0)})$, $\overline{\mu} := \mu^0$, and $\overline{z}(\overline{\mu}, \hat{\mathbf{p}}) := z_{LD}(\overline{\mu}, \hat{\mathbf{p}})$ where $z_{LD}(\overline{\mu}, \hat{\mathbf{p}}) = \sum_{\mathbf{p=1}}^{\hat{\mathbf{p}}} z_{LD}^{\mathbf{p}}(\overline{\mu})$. Set $k := 1$.

Step 1: Compute $s^k = \begin{pmatrix} (\delta^{(k)\mathbf{1}} - \delta^{(k)\mathbf{2}}) \\ \vdots \\ (\delta^{(k)\hat{\mathbf{p}}-\mathbf{1}} - \delta^{(k)\hat{\mathbf{p}}}) \\ (\delta^{(k)\hat{\mathbf{p}}} - \delta^{(k)\mathbf{1}}) \\ (x^{(k)\mathbf{1}} - x^{(k)\mathbf{2}}) \\ \vdots \\ (x^{(k)\hat{\mathbf{p}}-\mathbf{1}} - x^{(k)\hat{\mathbf{p}}}) \\ (x^{(k)\hat{\mathbf{p}}} - x^{(k)\mathbf{1}}) \end{pmatrix}$ and $\overline{s}^k = \begin{pmatrix} (\overline{\delta}^{\mathbf{1}} - \overline{\delta}^{\mathbf{2}}) \\ \vdots \\ (\overline{\delta}^{\hat{\mathbf{p}}-\mathbf{1}} - \overline{\delta}^{\hat{\mathbf{p}}}) \\ (\overline{\delta}^{\hat{\mathbf{p}}} - \overline{\delta}^{\mathbf{1}}) \\ (\overline{x}^{\mathbf{1}} - \overline{x}^{\mathbf{2}}) \\ \vdots \\ (\overline{x}^{\hat{\mathbf{p}}-\mathbf{1}} - \overline{x}^{\hat{\mathbf{p}}}) \\ (\overline{x}^{\hat{\mathbf{p}}} - \overline{x}^{\mathbf{1}}) \end{pmatrix}$,
check the stopping criteria given in Sec. 4.5 and if they are not satisfied, set $\mu^k := \overline{\mu} + \alpha_k \cdot \frac{(\overline{z}_{LD} - \overline{z}(\overline{\mu}, \hat{\mathbf{p}}))}{||\overline{s}^k||^2} \cdot \overline{s}^k$.
Solve the $\hat{\mathbf{p}}$ problems (12)-(13) with μ^k, and let $(\delta^{(k)}, \mathbf{x}^{(k)}, \gamma^{(k)}, y^{(k)})$ and $z_{LD}(\mu^k, \hat{\mathbf{p}})$ be the optimal solution and the solution value of (9), respectively. Then, update $(\overline{\delta}, \overline{x}, \overline{\gamma}, \overline{y}) := f_k \cdot (\delta^{(k)}, x^{(k)}, \gamma^{(k)}, y^{(k)}) + (1 - f_k) \cdot (\overline{\delta}, \overline{x}, \overline{\gamma}, \overline{y})$

Step 2: If $z_{LD}(\mu^k, \hat{\mathbf{p}}) > \overline{z}(\overline{\mu}, \hat{\mathbf{p}})$, update $\overline{\mu} := \mu^k$ and $\overline{z}(\overline{\mu}, \hat{\mathbf{p}}) := z_{LD}(\mu^k, \hat{\mathbf{p}})$.
Set $k := k + 1$ and go to Step 1.

Figure 4: Volume Algorithm (VA)

Note: The step directions s^k and $\overline{s}^k$ are used for obtaining the weighting parameter f_k and chosing the convergence parameters.

4.3 Progressive Hedging Algorithm

The Progressive Hedging Algorithm for problems with continuous variables alone was introduced in [20], see also [24] for a recent innovation. Our procedure for updating the Lagrange multipliers of the NAC (6)-(7) is given in Figure 5. The basic features are as follows: Let $(\delta^{(k)}, x^{(k)}, \gamma^{(k)}, y^{(k)})$ be an optimal solution of problem $(MIP_{LD}^{\hat{\mathbf{p}}}(\mu^k))$ (9) at iteration k. A new non-necessarily feasible solution can be defined as $\hat{\delta} = \sum_{\mathbf{p=1}}^{\hat{\mathbf{p}}} \mathbf{w^p} \delta^{(k)\mathbf{p}}$ and $\hat{x} = \sum_{\mathbf{p=1}}^{\hat{\mathbf{p}}} \mathbf{w^p} x^{(k)\mathbf{p}}$. These expressions represent an estimation of the expected value over the set of scenario clusters of the optimal solution obtained at iteration k.

Step 0: Given the Lagrange multipliers vector, μ^0, solve the $\hat{\mathbf{p}}$ problems (12)-(13) to obtain $(\delta^{(0)}, x^{(0)}, \gamma^{(0)}, y^{(0)})$ and $z_{LD}(\mu^0, \hat{\mathbf{p}})$ as the sum given in (11). Set $k := 0$.

Step 1: Compute $s^k = \begin{pmatrix} (\delta^{(k)\mathbf{1}} - \delta^{(k)\mathbf{2}}) \\ \vdots \\ (\delta^{(k)\hat{\mathbf{p}}-\mathbf{1}} - \delta^{(k)\hat{\mathbf{p}}}) \\ (\delta^{(k)\hat{\mathbf{p}}} - \delta^{(k)\mathbf{1}}) \\ (x^{(k)\mathbf{1}} - x^{(k)\mathbf{2}}) \\ \vdots \\ (x^{(k)\hat{\mathbf{p}}-\mathbf{1}} - x^{(k)\hat{\mathbf{p}}}) \\ (x^{(k)\hat{\mathbf{p}}} - x^{(k)\mathbf{1}}) \end{pmatrix}$ and $\hat{s}^k = \begin{pmatrix} (\delta^{(k)\mathbf{1}} - \hat{\delta}^{(k)}) \\ \vdots \\ (\delta^{(k)\hat{\mathbf{p}}-\mathbf{1}} - \hat{\delta}^{(k)}) \\ (\delta^{(k)\hat{\mathbf{p}}} - \hat{\delta}^{(k)}) \\ (x^{(k)\mathbf{1}} - \hat{x}^{(k)}) \\ \vdots \\ (x^{(k)\hat{\mathbf{p}}-\mathbf{1}} - \hat{x}^{(k)}) \\ (x^{(k)\hat{\mathbf{p}}} - \hat{x}^{(k)}) \end{pmatrix}$,

check the stopping criteria given in Sec. 4.5 and if they are not satisfied, set $\mu^{k+1} := \mu^k + \alpha_k \cdot \frac{(\overline{z}_{LD} - z_{LD}(\mu^k, \hat{\mathbf{p}}))}{||\hat{s}^k||^2} \cdot \hat{s}^k$.

Solve the $\hat{\mathbf{p}}$ problems (12)-(13) with μ^{k+1}, and let $(\delta^{(k+1)}, x^{(k+1)}, \gamma^{(k+1)}, y^{(k+1)})$ and $z_{LD}(\mu^{k+1}, \hat{\mathbf{p}})$ be the optimal solution and solution value, respectively.

Compute $\hat{\delta}^{k+1}$ and $\hat{x}^{k+1}$.

Set $k := k + 1$ and go to Step 1.

Figure 5: Progressive Hedging Algorithm (PHA)

Note: The step direction $\hat{s}^k$ is used for choosing the convergence parameters, see Sec. 4.6.

4.4 Dynamic Constrained Cutting Plane method

The DCCP is a Cutting Plane Method, see [18], in which the Lagrange multiplier at iteration k are updated by solving the following maximization problem

$$\begin{aligned} \overline{z}_{LD}(\mu^k, \hat{\mathbf{p}}) &= \max_{\mu \in C^k(\mu)} z \\ z &\leq \overline{z}_{LD}(\mu^i, \hat{\mathbf{p}}) \quad \forall i \in I, \end{aligned}$$

where $C^k(\mu)$ is the dynamically updated Lagrange multipliers feasible region and $\overline{z}_{LD}(\mu^i, \hat{\mathbf{p}})$ is a truncation of Taylor series expansion of the function $z_{LD}(\mu, \hat{\mathbf{p}})$ around the point μ^i, i.e.,

$$\begin{aligned} \overline{z}_{LD}(\mu^k) &= \max_{\mu \in C^k(\mu)} z \\ \text{s.t.} \quad z &\leq z_{LD}(\mu^i, \hat{\mathbf{p}}) + \sum_{\mathbf{p}=\mathbf{1}}^{\hat{\mathbf{p}}} (\mu^{\mathbf{p}} - \mu^{i,\mathbf{p}}) s^i \quad \forall i \in I, \end{aligned} \tag{14}$$

where I is the set of cutting planes, see (16), $z_{LD}(\mu^i, \hat{\mathbf{p}})$ is the Langrangean bound obtained at iteration i, and s^i is the subgradient vector of $z_{LD}(\mu, \hat{\mathbf{p}})$ at μ^i, for $i \in I$.

Notice that the number of constraints in model (14) grows with the number of iterations. To prevent the excessive size of the problem, $\hat{n}$ denotes the maximum number of cutting planes, i.e, the maximum number of constraints in model (14), so $|I| = \min\{k, \hat{n}\}$. Then, if

the number of iterations is lower than or equal to the maximum number of cutting planes, $k \leq \hat{n}$, all the cutting planes are considered in the model (14). Whereas, if the iteration number is larger than the maximum number of constraints, $k > \hat{n}$, the difference, say, d_i between the ith hyperplane $z_{LD}(\mu^i, \hat{\mathbf{p}}) + \sum_{\mathbf{p=1}}^{\hat{\mathbf{p}}}(\mu^{k,\mathbf{p}} - \mu^{i,\mathbf{p}})s^i$ and the Lagrangian bound obtained at iteration k is computed as follows,

$$d_i = z_{LD}(\mu^i, \hat{\mathbf{p}}) + \sum_{\mathbf{p=1}}^{\hat{\mathbf{p}}}(\mu^{k,\mathbf{p}} - \mu^{i,\mathbf{p}})s^i - z_{LD}(\mu^k, \hat{\mathbf{p}}). \tag{15}$$

The most distant hyperplanes are deleted from I. It should be noted that the residual d_i is always positive, since the cutting plane reconstruction of the dual function overestimates the actual dual function.

The feasible region $C^k(\mu)$ has the expression

$$C^k(\mu) = \{\mu, \underline{\mu}^k \leq \mu \leq \overline{\mu}^k\}, \tag{16}$$

where $\underline{\mu}^k$ and $\overline{\mu}^k$ denote the lower and the upper bound of the Lagrange multipliers vector at iteration k, respectively, such that they are updated at each iteration and can be expressed

$$\underline{\mu}_j^{k+1} = \mu_j^k - \alpha_k \cdot \beta^k \cdot |s_j^k| \quad \text{and} \quad \overline{\mu}_j^{k+1} = \mu_j^k + \alpha_k \cdot \beta^k \cdot |s_j^k|, \tag{17}$$

where μ_j^k is the jth component of the multipliers vector obtained as optimal solution of model (14) at iteration k and $\beta^k = \dfrac{(\overline{z}_{LD} - z_{LD}(\mu^k, \hat{\mathbf{p}}))}{||s^k||^2}$. Therefore, at iteration $k+1$ the feasible region $C^{k+1}(\mu)$ is defined around the optimal multipliers vector obtained in the previous iteration. The procedure for updating the Lagrange multipliers of the NAC (6)-(7) is given in Figure 6.

Step 0: Given the Lagrange multipliers vector, μ^0, solve the $\hat{\mathbf{p}}$ problems (12)-(13) to obtain $(\delta^{(0)}, x^{(0)}, \gamma^{(0)}, y^{(0)})$ and $z_{LD}(\mu^0, \hat{\mathbf{p}})$ as the sum given in (11). Set $k := 0$.

Step 1: Compute $s^k = \begin{pmatrix} (\delta^{(k)\mathbf{1}} - \delta^{(k)\mathbf{2}}) \\ \vdots \\ (\delta^{(k)\hat{\mathbf{p}}-\mathbf{1}} - \delta^{(k)\hat{\mathbf{p}}}) \\ (\delta^{(k)\hat{\mathbf{p}}} - \delta^{(k)\mathbf{1}}) \\ (x^{(k)\mathbf{1}} - x^{(k)\mathbf{2}}) \\ \vdots \\ (x^{(k)\hat{\mathbf{p}}-\mathbf{1}} - x^{(k)\hat{\mathbf{p}}}) \\ (x^{(k)\hat{\mathbf{p}}} - x^{(k)\mathbf{1}}) \end{pmatrix}$
check the stopping criteria given in Sec. 4.5 and if they are not satisfed, set
$\underline{\mu}_j^{k+1}$ and $\overline{\mu}_j^{k+1}$ as (17) where $\beta^k = \dfrac{(\overline{z}_{LD} - z_{LD}(\mu^k, \hat{\mathbf{p}}))}{||s^k||^2}$.
Solve the model (14) to obtain the new Lagrangian multiplier vector, μ^{k+1}.
If $k > \hat{n}$, compute d_i as (15) and delete $\iota \in argmax_{i \in I}\{d_i\}$ from I.

Step 2: Solve the $\hat{\mathbf{p}}$ problems (12)-(13) with μ^{k+1}, and let $(\delta^{(k+1)}, x^{(k+1)}, \gamma^{(k+1)}, y^{(k+1)})$ and $z_{LD}(\mu^{k+1}, \hat{\mathbf{p}})$ be the optimal solution and solution value, respectively.
Set $k := k + 1$ and go to Step 1.

Figure 6: Dynamic Constrained Cutting Plane method (DC-CP)

4.5 Stopping criteria

In this section we present the stopping criteria that are common to the four procedures described above. At Step 1 of each procedure, and after computing the subgradient vector s^k (SM) and (DC-CP), $\overline{s}^k$ (VA), or $\hat{s}^k$ (PHA), respectively, we compute its norm.

The stopping criterion 1, requires that the norm of the subgradient vector is near to zero (say, less than $\epsilon_s = 0.01$). We have used the ℓ_2 norm, but it could be possible to compute the ℓ_∞, with a little more computational effort and the solution would perhaps have been more accurate. If this criterion is satisfied, then the NAC (6)-(7) are satisfied as well and the optimal solution to the MIP model has been obtained. So, the Lagrangian bound coincides with the optimal solution value of the original stochastic integer problem.

The stopping criterion 2 common to the four procedures has two parts. The first is as follows,

$$\frac{|\sum_{\mathbf{p}=\mathbf{1}}^{\hat{\mathbf{p}}}[\mathbf{w}^{\mathbf{p}}(c_1\tilde{\delta}^{(k)\mathbf{p}} + c_2\tilde{x}^{(k)\mathbf{p}}) + \sum_{\omega\in\Omega_{\mathbf{p}}} w^{\omega}[q_1^{\omega}\tilde{\gamma}^{(k)\omega} + q_2^{\omega}\tilde{y}^{(k)\omega}]] - z_{LD}(\mu^k, \hat{\mathbf{p}})|}{|z_{LD}(\mu^k, \hat{\mathbf{p}})|} < \epsilon_z \qquad (18)$$

where $(\tilde{\delta}^{(k)\mathbf{p}}, \tilde{x}^{(k)\mathbf{p}}, \tilde{\gamma}^{(k)\omega}, \tilde{y}^{(k)\omega})$ denotes the incumbent solution, being $(\delta^{(k)\mathbf{p}}, x^{(k)\mathbf{p}}, \gamma^{(k)\omega}, y^{(k)\omega})$ for SM, PHA and DCCP and $(\overline{\delta}, \overline{x}, \overline{\gamma}, \overline{y})$ for VA, and ϵ_z is a given tolerance. In particular, we use $\epsilon_z = 0.008$.

The second part is given by

$$\frac{\sum_{\mathbf{p}=\mathbf{1}}^{\hat{\mathbf{p}}} |\tilde{s}_{\mathbf{p}\delta}|}{\hat{\mathbf{p}} \cdot n_\delta} < \epsilon_\delta \quad \text{and} \quad \frac{\sum_{\mathbf{p}=\mathbf{1}}^{\hat{\mathbf{p}}} |\tilde{s}_{\mathbf{p}x}|}{\hat{\mathbf{p}} \cdot n_x} < \epsilon_x, \qquad (19)$$

where $\hat{\mathbf{p}} \cdot n_\delta$ and $\hat{\mathbf{p}} \cdot n_x$ are the number of NAC for the δ and x variables, respectively, $\tilde{s}_{\mathbf{p}\delta}$ and $|\tilde{s}_{\mathbf{p}x}|$ for cluster $\mathbf{p}$ denote the absolute deviations for the corresponding δ and x rows of vector s^k for SM and DCCP, $\overline{s}^k$ for VA and $\hat{s}^k$ for PHA, whereas ϵ_δ and ϵ_x are given tolerances. In particular, we use $\epsilon_\delta = 0.01$ and $\epsilon_x = 0.1$.

Finally, the stopping criterion 3 requires that the incumbent solution value, $z_{LD}(\mu^k, \hat{\mathbf{p}})$ does not improve (given a tolerance, say $\epsilon = 0.0001$) after a sequence of ten consecutive iterations.

When any of the stopping criteria is satisfied, the possible situations are as follow related to the CLD bound $z_{LD}(\mu^k, \hat{\mathbf{p}})$:

1. Stopping criterion 1. The bound is the solution value of the original problem and, additionally, the solution is feasible and then, it is the optimal one. We denote the corresponding results in green in Tables 3-24.

2. Stopping criterion 2. The (strong) bound is the objective function value of a quasi-feasible solution given the optimality tolerances that have been established. We denote the corresponding results in blue in Tables 3-24.

3. Stopping criterion 3. The bound is the strongest bound that can be obtained given the set of tolerances and parameters that have been established. We denote the corresponding results in red in Tables 3-24.

4.6 Choice of the convergence parameters

The performance of the procedures given above is very sensitive to the choice of the given parameters: the initial upper bound $\overline{z}_{LD}$, the initial step size parameter α_0 and moreover the procedure for updating this step size parameter at each iteration α_k; some implementation details are given in [4]. In this sense, and following the notation given in that paper, we have considered three types of iterations for setting the value of α_k. The iteration at which there is no improvement in the value of function $z_{LD}(\mu, \hat{\mathbf{p}})$, such that $z_{LD}(\mu^k, \hat{\mathbf{p}}) \leq z_{LD}(\mu^{k-1}, \hat{\mathbf{p}})$ is called *red*. Otherwise, i.e., $z_{LD}(\mu^k, \hat{\mathbf{p}}) > z_{LD}(\mu^{k-1}, \hat{\mathbf{p}})$, let the vector h^k be computed as follows: $h^k = (s^k)^t \cdot s^{k-1}$ in the Subgradient and Dynamic Constrained Cutting Plane methods; $h^k = (s^k)^t \cdot \overline{s}^k$ in the Volume Algorithm, and $h^k = (s^k)^t \cdot \hat{s}^k$ in the Progressive Hedging Algorithm, where s^k, $\overline{s}^k$ and $\hat{s}^k$ denote, respectively, the subgradient vector calculated in Step 1 of the corresponding procedure. Notice that $h^k < 0$ means that a longer step in the direction of s^k would produce a smaller value for $z_{LD}(\mu^k, \hat{\mathbf{p}})$. In this case, the iteration is called *yellow*. If $h^k \geq 0$ then the iteration is called *green*. At each green iteration we multiply α_k by 1.1. After each sequence of $\#red$ consecutive red iterations we multiply α_k by 0.66.

Note that there is no relationship between the color of the iterations, yellow, red or green color, introduced in [4], to describe the procedure for updating the value of the step size parameter α_k, and that shows the different CLD bounds in the Tables 3-24 showing the stopping criterion has occured.

The optimal values for $\#red$ and α_0 must be adequately tested for each instance and are clearly dependent on the initial upper bound $\overline{z}_{LD}$ considered, see [10]. However, we observed in our computational experimentation (see Sec. 5) that, in general, and for any choice of these parameters, the clustering partition provides stronger lower bounds when computing the Lagrangian bound at iteration zero, i.e., $z_{LD}(\mu^0, \hat{\mathbf{p}})$. Note: The initial vector of the Lagrange multipliers has been taked as a vector of zeros, $\mu^0 = \vec{0}$, given the good results that we have reported in [10] for singleton scenario clusters by computationally comparing this choice with some other alternative.

For each clustering partition, we obtain an interval for the solution value of the original problem, given by $[z_{LD}(\mu^0, \hat{\mathbf{p}}), \overline{z}_{LD}]$. As we will show, the tightness of the Lagrangian bound at iteration zero, $z_{LD}(\mu^0, \hat{\mathbf{p}})$, depends upon the cluster partitioning i.e., $\hat{\mathbf{p}}$ that is considered; while in the case of the upper bound $\overline{z}_{LD}(\rho)$, its goodness depends on the quasi-optimality tolerance $\rho\%$ considered when the MIP solver obtains it. When using the preprocessing and parallel computing tools available by default in CPLEX, stronger bounds are efficiently computed, see Table 2.

In order to homogenize the performance of the two solvers to be used, namely CPLEX within COIN-OR [17] and the LP/MIP functions of it as well as the different cluster partitionings, we have considered $\#red = 1$ in all the instances in the testbed. We have experimented as well as the same initial steplength value α_0, although diminishing it for both solvers in some instances, depending on the extension of the interval that contains the solution value, see Sec. 5.

The parameter f_k in the Volume Algorithm is set to a fixed value for a number of iterations and is decreased afterwards. Let s^k and $\overline{s}^k$ be defined as in Step 1 of the procedure. Let also f_{max} be an upper bound of f_k. Then, we can compute f_{opt} as the value that minimizes

$||f_k \cdot s^k + (1 - f_k) \cdot \overline{s}^k||$. It is easy to verify that this value is $f_{opt} = \frac{\sum_{i=1}^{2\hat{\mathbf{p}}} \overline{s}_i^k (s_i^k - \overline{s}_i^k)}{\sum_{i=1}^{2\hat{\mathbf{p}}} (\overline{s}_i^k - s_i^k)(s_i^k - \overline{s}_i^k)}$. If $f_{opt} < 0$, set $f_k = \frac{1}{10} \cdot f_{max}$. Otherwise, set $f_k = \min\{f_{max}, f_{opt}\}$. In our computational experimentation we have used $f_{max} = f_0 = 0.1$ and we have decreased its value near to the end.

Finally, the maximum number of cutting planes, $\hat{n}$, in the Dynamic Constrained Cutting Plane method has been fixed to $\hat{n} = 30$.

5 Computational experience

We report the results of the computational experience obtained while optimizing the two-stage MIP model (1) over some randomly generated instances. The first two instances of the testbed are small-medium sized, while the other instances are larger, significantly bigger than those normally reported in the literature, e.g., [23].

The computational experiments were conducted in a Workstation Debian Linux (kernel v2.6.32 with 64 bits), 2 processors Xeon 5355 (Quad Core with 2x4 cores), 2.664 Ghz and 16 Gb of RAM.

The four procedures given above have been implemented in a C++ experimental code. It uses alternatively two of the state-of-the-art optimization engines, in particular CPLEX v12.2 within the open source engine COIN-OR and the LP/MIP default functions Clp and Cbc of the same COIN-OR system. Both optimizers are used by the CLD algorithm for solving the LP relaxation of the whole model and the related mixed 0-1 cluster submodels.

We will compare the results obtained by using both optimizers, COIN-OR and CPLEX. The use of the latter is denoted with the upperindex ^{ppc}, since this solver uses (by default) the state-of-the-art preprocessing and parallel computing (in our case with a parallel scheme of eight threads, one per core). The four Lagrange multipliers updating procedures presented above can be enriched by providing a variety of specialized preprocessing, cut generation and appending, probing and parallel computation tools, see [19], that can customize the experimental code to achieve maximum efficiency.

The structure of the DEM in compact representation for the instances, which is inspired in model (38) of [23], can be expressed

$$\begin{array}{ll} \min & c_1\delta + c_2 x + \sum_{\omega=1}^{|\Omega|} w^\omega (q_1^\omega \gamma^\omega + q_2^\omega y^\omega) \\ \text{s.t.} & b_1 \leq A \begin{pmatrix} \delta \\ x \end{pmatrix} \leq b_2 \\ & h_1^\omega \leq T^\omega \begin{pmatrix} \delta \\ x \end{pmatrix} + W^\omega \begin{pmatrix} \gamma^\omega \\ y^\omega \end{pmatrix} \leq h_2^\omega \quad \forall \omega \in \Omega \\ & x, y^\omega \in [0,1] \quad \forall \omega \in \Omega \\ & \delta, \gamma^\omega \in \{0,1\} \quad \forall \omega \in \Omega, \end{array} \tag{20}$$

Note that the variables in both stages are bounded. The vectors of variables δ and γ are integer, moreover they are binary, whereas the vectors of continuous variables, x

Table 1: Model dimensions

	Compact representation							Splitting variable representation					
Instance	m^c	n^c_δ	n^c_x	n_γ	n_y	nel^c	$dens^c$	m^s	n^s_δ	n^s_x	nel^s	$dens^s$	$\|\Omega\|$
P1	136	4	4	128	128	2112	5.88	640	128	128	4608	1.41	32
P2	148	10	10	128	128	3984	9.75	1408	320	320	17664	1.40	32
P3	288	5	10	280	420	70120	3.40	4410	350	700	80500	1.04	70
P4	1290	30	15	1280	256	73410	3.59	8320	3840	1920	142080	0.23	128
P5	1935	25	10	2560	1920	134925	1.54	8320	3200	1280	210560	0.28	128
P6	2010	20	20	2000	2000	120400	1.48	12000	4000	4000	216000	0.15	200
P7	2010	20	40	3000	2000	170600	1.68	16000	4000	8000	314000	0.11	200
P8	2005	12	15	6000	4000	104135	0.52	14800	4800	6000	179600	0.06	400
P9	2005	10	15	3600	3600	86125	0.59	14000	4000	6000	156000	0.06	400
P10	2520	30	40	5000	2500	213900	2.76	47500	15000	20000	982500	0.05	500
P11	2520	50	50	5000	2500	289500	1.51	62500	25000	25000	1387500	0.04	500

and y, are scaled onto [0,1]. The likelihood attributed to the scenarios is equal under each scenario, i.e, $w^\omega = \frac{1}{|\Omega|}\ \forall \omega \in \Omega$, being Ω the set of scenarios. The vectors of the objective function coefficients, $c_1, c_2, (q^\omega_1), (q^\omega_2)$ are generated using the uniform distribution over $[-2.5, -1.5], [-2.5, -1.5], [-30+\frac{\omega}{|\Omega|}, -10+\frac{\omega}{|\Omega|}]$ and $[-30+\frac{\omega}{|\Omega|}, -10+\frac{\omega}{|\Omega|}]$, respectively. The left-hand-side vectors, $b_1, (h^\omega_1)$ are fixed to $\frac{1}{2}\cdot k_1$ and $\frac{1}{2}\cdot k_1 + \frac{\omega}{|\Omega|}$, respectively. The right-hand-side vectors, $b_2, (h^\omega_2)$, are generated using the uniform distribution over $[k_2, k_2 + k_1 \cdot (n_\delta + n_x)]$ and $[k_3 + \frac{\omega}{|\Omega|}, k_3 + \frac{\omega}{|\Omega|} + k_1 \cdot (n_\delta + n_x + n_\gamma + n_y)]$, respectively, where $k_1 \in [0, 1]$, $k_2 \in [0, 41.5]$ and $k_3 \in [0, 30.5]$. n_δ and n_x are the number of 0-1 and continuous first stage variables, and n_γ and n_y are the corresponding number of 0-1 and continuous second stage variables. A is the matrix of coefficients for the first stage constraints, and the technology matrices T^ω and W^ω for the second stage variables are generated using the uniform distribution over $[0, 2], [-0.1 \cdot \frac{\omega}{|\Omega|}, -0.1\ \cdot \frac{\omega}{|\Omega|} + 0.3]$ and $[1.5 \cdot \frac{\omega}{|\Omega|}, 1.5 \cdot \frac{\omega}{|\Omega|} + 8.0]$, respectively.

Table 1 gives the dimensions of the mixed 0-1 DEM for the 11 instances of the testbed that we have experimented with in compact and splitting variable representations. The headings are as follows: m^c, m^s, number of constraints; n^c_δ, n^s_δ, number of 0-1 first stage variables; and n^c_x, n^s_x number of continuous first stage variables in compact and splitting variable respresentation, respectively. n_γ, number of 0-1 second stage variables; n_y, number of continuous second stage variables. nel^c, nel^s, number of nonzero coefficients in the constraint matrix; and $dens^c$, $dens^s$, constraint matrix density (in %) in compact and splitting variable representation, respectively. Finally, $|\Omega|$ denotes number of scenarios. Notice that the numbers of second stage variables n_γ and n_y, are the same under both representations. It is worth pointing out that the testbed has 4 types of instances from the DEM dimensions point of view, namely the instances P1 and P2 are toy ones, P3 up to P7 are medium sized instances, P8 and P9 are large-scale instances, and P10 and P11 are very large-scale instances given the state-of-the art of general MIP solvers.

Table 2 shows some of the main results obtained by plain use of the two optimizers COIN-OR and CPLEX for solving the original problem (20). The headings are as follows: z^{ppc}_{LP}, LP solution value; z^{ppc}_{MIP}, objective function value of the CPLEX incumbent solution (but it is the solution value for the toy instances P1 and P2) of problem (20); T^{ppc}_{LP} and T^{ppc}_{MIP}, elapsed times (in secs.) to obtain the z^{ppc}_{LP} and z^{ppc}_{MIP} values, respectively, by plain use of CPLEX in the compact representation of problem (20). Upper bounds $\overline{z}(\rho)$ and $\overline{z}^{ppc}(\rho)$ of the solution

Table 2: LP relaxation lower bound and upper bound for the optimal MIP solution value

Case	z_{LP}^{ppc}	z_{MIP}^{ppc}	T_{LP}^{ppc}	T_{MIP}^{ppc}	$\overline{z}(\rho)$	$T_{\overline{z}(\rho)}$	$\overline{z}^{ppc}(\rho)$	$T_{\overline{z}(\rho)}^{ppc}$
P1	-81.14	-80.4820	0.01	1	-80.1945(1)	0.27	-80.3516(1)	0
P2	-100.42	-99.8996	0.01	2	-99.3327(1)	0.25	-99.6225(1)	0
P3	-61.40	-59.4645(*)	0.28	–	-58.6387(10)	70	-59.46(0.1)	28
P4	-76.05	-70.7906(*)	0.09	–	-68.5212(10)	24	-70.7906(1)	40
P5	-86.70	-84.2161(*)	0.21	–	-82.3986(5)	20	-84.1637(0.5)	156
P6	-69.30	-66.0478(*)	0.29	–	-65.955(5)	125	-66.0315(0.5)	49
P7	-83.50	-79.8772(*)	0.41	–	-77.326(10)	111	-79.8045(1)	87
P8	-116.32	-114.318(*)	0.28	–	-113.235(5)	61	-114.044(0.5)	37
P9	-95.81	-94.1302(*)	0.26	–	-92.9241(5)	37	-94.1227(0.1)	89
P10	-301.54	-300.456(*)	0.62	–	-300.166(0.5)	114	300.425(0.05)	27
P11	-321.29	-320.283(*)	0.80	–	-317.724(5)	54	-320.249(0.05)	61

–: Time limit exceeded (3 hours = 10800 secs.)
(*): Incumbent solution value at the time limit

value of the original problem that have been obtained as quasi-optimal solution values with a ρ% tolerance computed by plain use of COIN-OR and CPLEX, respectively; and, finally, $T_{\overline{z}(\rho)}$ and $T_{\overline{z}(\rho)}^{ppc}$, elapsed times (in secs.) for obtaining the corresponding upper bounds.

Table 2 shows relevant information concerning the difficulty of the instances we were experimenting with, in particular the larger ones (i.e., from P3 to P11). None of them are solved up to optimality by plain use of solvers COIN-OR and CPLEX within the three hours elapsed time limit. Therefore, the objective function value of the incumbent solution provided by CPLEX, is in some instances just an upper bound of the solution value of the original stochastic instance, i.e., P4. In some other instances (i.e., P6, P7, P8, P10 and P11) the incumbent solution coincides with the optimal one. However, this fact is not known by the solver, but we can guarantee this after having obtained a green solution with our procedures, evidently requiring a total elapsed time much less than three hours. Finally, there some other instances (i.e., P3, P5 and P9) for which the plain use of CPLEX provides an incumbent solution with an objective function value slightly higher than the CLD bound provided by our procedures, but with a much greater computational effort. Note: In these situations the quasi-optimality gap between the CPLEX incumbent solution and the best CLD bound, defined as $|\frac{z_{MIP}^{ppc}-z_{LD}}{z_{LD}}|$, is for instance P3, $|\frac{-59.4645+59.4647}{-59.4647}| = 3.36 \cdot 10^{-6}$, for instance P5, $|\frac{-84.2161+84.2243}{-84.2243}| = 9.73 \cdot 10^{-5}$, and for instance P9, $|\frac{-94.1302+94.1407}{-94.1407}| = 1.11 \cdot 10^{-4}$. Very small in both of them. However, the traditional optimality gap defined as $|\frac{z_{MIP}^{ppc}-z_{LP}^{ppc}}{z_{LP}^{ppc}}|$, of value $0.031, 0.028$ and 0.017 for instances P3, P5 and P9, respectively, is substantially greater. The details of this conclusion are shown in the results presented in the rest of the section.

Tables 3-4 until 23-24 show in detail the main results of our computational experience for each of the instances P1 until P11, without and with sophisticated preprocessing and parallel computation tools (i.e., by using COIN-OR functions and CPLEX, respectively). In all of them, the heading $\hat{\mathbf{p}}$ denotes the cluster partition i.e., the number of scenario clusters that are considered. In all the instances we have considered four scenario cluster partitions. For each cluster partition (i.e., at each column in the tables) we present the interval of the solution value (i.e., the objective function value of the optimal solution of the original stochastic mixed 0-1 instance) given by $[z_{LD}(\mu^0, \hat{\mathbf{p}}), \overline{z}(\rho)]$. Additionally, α_0 denotes the initial step size

parameter; $z_{SM}[ite]$, $z_{VA}[ite]$, $z_{PHA}[ite]$ and $z_{DCCP}[ite]$ denote the CLD bounds obtained in $[ite]$, the corresponding number of iterations required by using the procedures SM, VA, PHA and DCCP, respectively; T_S, T_V, T_P and T_D denote the related elapsed times (in secs.) by using the COIN-OR functions. Finally, the upperindex ^{ppc} in these headings indicate the case of using CPLEX.

Table 3: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P1

$z_{MIP} \in$	$\hat{\mathbf{p}} = 32$ clusters $[-81.0529, -80.1945]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 16$ clusters $[-80.7393, -80.1945]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-80.6329, -80.1945]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 4$ clusters $[-80.553, -80.1945]$ $\alpha_0 = 1.9$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[63] = -80.5098$	6	$z[60] = -80.4873$	5	$z[29] = -80.4834$	4	$z[34] = -80.4825$	6
							$z[52] = -80.482$	10
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[104] = -80.494$	12	$z[58] = -80.4843$	5	$z[61] = -80.4845$	6	$z[45] = -80.482$	10
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[213] = -80.4861$	21	$z[89] = -80.4886$	7	$z[108] = -80.4827$	10	$z[60] = -80.482$	12
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[82] = -80.5033$	9	$z[68] = -80.4857$	7	$z[32] = -80.4836$	3	$z[14] = -80.4827$	3
							$z[23] = -80.482$	5

Table 4: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P1

$z_{MIP} \in$	$\hat{\mathbf{p}} = 32$ clusters $[-81.0529, -80.3516]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 16$ clusters $[-80.7393, -80.3516]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-80.6329, -80.3516]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 4$ clusters $[-80.553, -80.3516]$ $\alpha_0 = 1.9$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[62] = -80.5113$	16	$z[63] = -80.4917$	10	$z[29] = -80.4843$	4	$z[33] = -80.482$	3
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[133] = -80.4952$	34	$z[58] = -80.4847$	10	$z[69] = -80.4842$	6	$z[37] = -80.482$	4
							$z[56] = -80.482$	11
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[180] = -80.4857$	48	$z[105] = -80.4852$	17	$z[115] = -80.482$	12	$z[63] = -80.482$	7
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[80] = -80.5137$	25	$z[49] = -80.4863$	19	$z[21] = -80.4839$	6	$z[17] = -80.482$	2

Tables 3-4 show the results reported for instance P1. The CLD bounds obtained by using both solvers are very similar, but with a higher computational effort in case of using CPLEX, perhaps due to the small dimensions of the instance. Notice that this happens for all the four procedures and the four cluster partitions that we have experimented with, but for the column corresponding to the partition in $\hat{\mathbf{p}} = 4$ clusters, where each cluster submodel has 8 scenarios and the four procedures are more efficient when using CPLEX. The first column in both tables corresponds to the traditional LD, where the number of clusters is the number of scenarios. Notice that in this column the color of the solutions is red (i.e., the third stopping criterion has been satisfied) or blue (second stopping criterion), which indicates that the CLD bound is, at least, the strongest bound that can be obtained for the given tolerances. The color of the solutions in both tables is green (first stopping criterion), which means that the CLD bound is the solution value of the original problem. Notice also that for some cases, although the CLD bound is equal to the solution value of the original problem, the color of

the results is not green, see VA for $\hat{\mathbf{p}}$ =4 in both tables. This is due to the fact that the CLD bound does not satisfy the NAC, i.e., the norm of the corresponding subgradient vector $\overline{s}^k$ is higher than the given tolerance.

Table 5: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P2

$z_{MIP} \in$	$\hat{\mathbf{p}} = 32$ clusters $[-100.289, -99.3327]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 16$ clusters $[-100.15, -99.3327]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-99.9725, -99.3327]$ $\alpha_0 = 1.5$		$\hat{\mathbf{p}} = 4$ clusters $[-99.944, -99.3327]$ $\alpha_0 = 1.5$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[51] = -99.9233$	3	$z[31] = -99.9017$	2	$z[28] = -99.8997$	2	$z[8] = -99.9002$	1
							$z[59] = -99.8996$	10
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[30] = -99.9436$	2	$z[17] = -99.9578$	1	$z[7] = -99.9537$	0	$z[5] = -99.944$	1
							$z[48] = -99.8996$	8
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[39] = -99.9003$	7	$z[73] = -99.8996$	4	$z[55] = -99.8996$	4	$z[27] = -99.9009$	5
							$z[89] = -99.8996$	16
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[24] = -99.9504$	2	$z[35] = -99.9091$	3	$z[21] = -99.9002$	2	$z[8] = -99.9122$	2
							$z[38] = -99.8996$	7

It can be observed in the results for the larger instances that sometimes the optimal cluster partitioning is not the smallest. In these situations, it may be more efficient to consider a great number of clusters and then, more manageable sized cluster submodels.

Tables 5-6 show the results obtained for instance P2. In order to eliminate the oscillatory behavior of the iterative procedures for narrow solution value intervals, we have reduced the initial step size parameter for the cases with partitions in $\hat{\mathbf{p}}$ =8 and 4 clusters. Again the optimal partition is the one shown in the last column of both tables where the solution value is found. The quality of the CLD bounds obtained for the small instances P1 and P2 is very similar, but the elapsed time is smaller for the procedures SM and DCCP, followed by VA and PHA. SM is even more efficient than the plain use of CPLEX for instance P2.

Table 6: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P2

$z_{MIP} \in$	$\hat{\mathbf{p}} = 32$ clusters $[-100.289, -99.6225]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 16$ clusters $[-100.15, -99.6225]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-99.9725, -99.6225]$ $\alpha_0 = 1.5$		$\hat{\mathbf{p}} = 4$ clusters $[-99.944, -99.6225]$ $\alpha_0 = 1.5$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[28] = -99.9378$	11	$z[32] = -99.9023$	8	$z[21] = -99.8996$	10	$z[8] = -89.8996$	1
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[31] = -99.9397$	13	$z[18] = -99.9348$	3	$z[9] = -99.9663$	2	$z[5] = -99.944$	1
							$z[55] = -99.8996$	5
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[116] = -99.9014$	51	$z[101] = -99.8996$	23	$z[50] = -99.8996$	6	$z[32] = -99.8996$	3
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[41] = -99.9346$	17	$z[30] = -99.9057$	7	$z[20] = -99.9013$	3	$z[11] = -99.9010$	2
							$z[29] = -99.8996$	3

P3 is one of the most difficult instances in our testbed, in spite of the dimensions of its model. Tables 7-8 report the main results. The COIN-OR functions (see Table 7) require more than three hours to obtain the CLD bounds in case of considering cluster partitions in

$\hat{\mathbf{p}}$ =10 or less clusters and then, we cannot provide the interval of the solution value, due to exceeding the time limit . However, this is obtained at the first iteration when using CPLEX (see Table 8). After the blue solution is obtained at the first iteration, the procedures continue iterating until obtaining the strongest CLD bound by satisfying the third stopping criterion, i.e., a red solution.

Table 7: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P3

$z_{MIP} \in$	$\hat{\mathbf{p}} = 70$ clusters $[-59.5529, -58.6387]$ $\alpha_0 = 0.5$		$\hat{\mathbf{p}} = 35$ clusters $[-59.5142, -58.6387]$ $\alpha_0 = 0.5$		$\hat{\mathbf{p}} = 10$ clusters $[-, -58.6387]$ $\alpha_0 = -$		$\hat{\mathbf{p}} = 5$ clusters $[-, -58.6387]$ $\alpha_0 = -$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[18] = -59.487$	117	$z[4] = -59.4902$	107	–	–	–	–
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[29] = -59.481$	180	$z[17] = -59.4858$	367	–	–	–	–
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[53] = -59.4955$	329	$z[60] = -59.4815$	1233	–	–	–	–
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[18] = -59.4863$	116	$z[20] = -59.4798$	423	–	–	–	–

Table 8: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P3

$z_{MIP} \in$	$\hat{\mathbf{p}} = 70$ clusters $[-59.5529, -59.46]$ $\alpha_0 = 0.5$		$\hat{\mathbf{p}} = 35$ clusters $[-59.5142, -59.46]$ $\alpha_0 = 0.5$		$\hat{\mathbf{p}} = 10$ clusters $[-59.4821, -59.46]$ $\alpha_0 = 0.5$		$\hat{\mathbf{p}} = 5$ clusters $[-59.4763, -59.46]$ $\alpha_0 = 0.1$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[10] = -59.4925$	29	$z[11] = -59.4863$	34	$z[0] = -59.4821$	6	$z[0] = -59.4763$	10
							$z[54] = -59.4669$	457
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[26] = -59.4847$	71	$z[23] = -59.4828$	68	$z[0] = -59.4821$	6	$z[0] = -59.4763$	10
							$z[50] = -59.4649$	432
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[24] = -59.483$	64	$z[9] = -59.4888$	28	$z[0] = -59.4821$	6	$z[0] = -59.4763$	10
							$z[117] = -59.4647$	967
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[13] = -59.4934$	35	$z[12] = -59.4848$	37	$z[0] = -59.4821$	5	$z[0] = -59.4763$	10
							$z[62] = -59.4647$	489

Tables 9-10 and 11-12 show the results for the instances P4 and P5, respectively, being very similar for both instances. As in instance P3, more than three hours are required to obtain the CLD bounds by using COIN-OR in case of considering cluster partitions in $\hat{\mathbf{p}} =$ 8 or less clusters (see Tables 9 and 11). In both instances, the strongest CLD bounds are obtained by using CPLEX in case of considering a partition in $\hat{\mathbf{p}} = 4$ clusters (see Tables 10 and 12).

Table 9: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P4

$z_{MIP} \in$	$\hat{\mathbf{p}} = 128$ clusters $[-73.9588, -68.5212]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 32$ clusters $[-70.108, -68.5212]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-, -68.5212]$ $\alpha_0 = -$		$\hat{\mathbf{p}} = 4$ clusters $[-, -68.5212]$ $\alpha_0 = -$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[104] = -71.5568$	207	$z[90] = -71.0946$	539	–	–	–	–
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[101] = -71.3968$	206	$z[63] = -71.0582$	386	–	–	–	–
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[213] = -71.3772$	438	$z[154] = -71.0031$	954	–	–	–	–
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[135] = -71.4846$	285	$z[108] = -71.1141$	669	–	–	–	–

Table 10: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P4

$z_{MIP} \in$	$\hat{\mathbf{p}} = 128$ clusters $[-73.9588, -70.7906]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 32$ clusters $[-70.108, -70.7906]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-71.3013, -70.7906]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 4$ clusters $[-71.0679, -70.7906]$ $\alpha_0 = 1.9$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[106] = -71.5566$	310	$z[75] = -71.1511$	204	$z[21] = -70.911$	345	$z[7] = -70.8615$	346
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[142] = -71.4157$	440	$z[57] = -71.0701$	161	$z[30] = -70.897$	483	$z[29] = -70.8533$	872
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[188] = -71.3789$	606	$z[149] = -71.0235$	442	$z[16] = -71.0144$	218	$z[14] = -70.895$	484
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[150] = -71.584$	486	$z[97] = -71.1523$	268	$z[20] = -70.9028$	328	$z[14] = -70.895$	484

Table 11: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P5

$z_{MIP} \in$	$\hat{\mathbf{p}} = 128$ clusters $[-89.1014, -82.3986]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 32$ clusters $[-86.7169, -82.3986]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-, -82.3986]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 4$ clusters $[-, -82.3986]$ $\alpha_0 = -$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[99] = -85.4933$	170	$z[99] = -84.7134$	583	–	–	–	–
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[228] = -85.2149$	512	$z[156] = -84.5161$	997	–	–	–	–
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[240] = -84.9909$	403	$z[193] = -84.4516$	1076	–	–	–	–
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[163] = -85.5784$	296	$z[107] = -84.785$	651	–	–	–	–

Table 12: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P5

$z_{MIP} \in$	$\hat{\mathbf{p}} = 128$ clusters $[-89.1014, -84.1637]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 32$ clusters $[-86.7169, -84.1637]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-85.4198, -84.1637]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 4$ clusters $[-85.1652, -84.1637]$ $\alpha_0 = 1.9$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[123] = -85.5389$	546	$z[88] = -84.7792$	247	$z[70] = -84.2606$	1151	$z[21] = -84.2370$	1069
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[228] = -85.2448$	941	$z[189] = -84.5286$	593	$z[65] = -84.2433$	1148	$z[47] = -84.2243$	1487
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[291] = -84.9872$	1241	$z[220] = -84.4464$	949	$z[107] = -84.2429$	1881	$z[46] = -84.2259$	1383
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[145] = -85.576$	578	$z[110] = -84.9279$	313	$z[92] = -84.3066$	1511	$z[28] = -84.2349$	1730

P6 and P7 are instances with similar dimensions and the results are also similar to those obtained for the instances P4 and P5 in the sense that the behavior of the four procedures is analogous when using COIN-OR for partitions in $\hat{\mathbf{p}}$ =8 and 4 clusters (see Tables 13 and 15). However when using CPLEX the optimal partition is $\hat{\mathbf{p}}$ =4 clusters for P6 and $\hat{\mathbf{p}}$ =8 clusters for P7 (see Tables 14 and 16), i.e., the smallest and then, reaching the optimal solution in a more efficient way for the four procedures. Notice that for instance P7 with $\hat{\mathbf{p}}$ =8 and 4 clusters, a feasible CLD bound is obtained at iteration zero for all the procedures by using CPLEX. The efficiency of the four procedures is lower for $\hat{\mathbf{p}}$ =4 clusters and, in particular, PHA requires more than 15000 secs. to reach the optimal solution.

Table 13: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P6

$z_{MIP} \in$	$\hat{\mathbf{p}} = 200$ clusters $[-68.0453, -65.955]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-66.639, -65.955]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-, -65.955]$ $\alpha_0 = -$		$\hat{\mathbf{p}} = 4$ clusters $[-, -65.955]$ $\alpha_0 = -$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[101] = -66.2772$	281	$z[30] = -66.1435$	176	–	–	–	–
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[110] = -66.205$	309	$z[73] = -66.0966$	444	–	–	–	–
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[189] = -66.1759$	498	$z[95] = -66.088$	543	–	–	–	–
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[149] = -66.2739$	443	$z[104] = -66.1349$	602	–	–	–	–

Table 15: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P7

$z_{MIP} \in$	$\hat{\mathbf{p}} = 200$ clusters $[-81.934, -77.326]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-80.5159, -77.326]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-, -77.326]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 4$ clusters $[-, -77.326]$ $\alpha_0 = -$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[37] = -80.1946$	128	$z[26] = -79.9765$	256	–	–	–	–
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[55] = -80.1311$	128	$z[32] = -79.969$	384	–	–	–	–
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[213] = -80.0999$	502	$z[198] = -79.9803$	1623	–	–	–	–
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[126] = -80.1823$	312	$z[34] = -79.9836$	328	–	–	–	–

Table 14: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P6

$z_{MIP} \in$	$\hat{\mathbf{p}} = 200$ clusters $[-68.0453, -66.0315]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-66.639, -66.0315]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-66.1605, -66.0315]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 4$ clusters $[-66.1153, -66.0315]$ $\alpha_0 = 1.9$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[107] = -66.2753$	558	$z[63] = -66.1358$	177	$z[18] = -66.0601$	302	$z[8] = -66.0503$	315
							$z[31] = -66.0478$	935
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[88] = -66.2136$	461	$z[65] = -66.094$	178	$z[25] = -66.0615$	421	$z[7] = -66.106$	194
							$z[48] = -66.0478$	1282
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[181] = -66.1928$	963	$z[106] = -66.0885$	279	$z[37] = -66.0582$	582	$z[16] = -66.0488$	437
							$z[51] = -66.0478$	1468
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[51] = -66.9962$	227	$z[46] = -66.5703$	111	$z[9] = -66.062$	160	$z[6] = -66.0519$	188
							$z[18] = -66.0478$	491

Table 16: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P7

$z_{MIP} \in$	$\hat{\mathbf{p}} = 200$ clusters $[-81.934, -79.8045]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-80.5159, -79.8045]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-79.9739, -79.8045]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 4$ clusters $[-79.917, -79.8045]$ $\alpha_0 = 1.9$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[45] = -80.2291$	201	$z[36] = -79.9984$	143	$z[0] = -79.9739$	33	$z[0] = -79.917$	130
					z[62]=-79.8772	1848	$z[40] = -79.8772$	6319
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[51] = -80.157$	237	$z[36] = -79.9837$	145	$z[0] = -79.9739$	33	$z[0] = -79.917$	130
					$z[73] = -79.8772$	2293	$z[45] = -79.8772$	6689
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[131] = -80.0797$	602	$z[45] = -79.949$	180	$z[0] = -79.9739$	33	$z[0] = -79.917$	129
					$z[150] = -79.8772$	4435	–	–
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[117] = -80.1835$	566	$z[54] = -80.0096$	219	$z[0] = -79.9739$	33	$z[0] = -79.917$	129
					$z[134] = -79.8772$	4262	$z[45] = -79.8772$	8419

P8 and P9 are large instances both with 400 scenarios. Again, when the procedures are implemented by using COIN-OR for partitions in a small number of clusters, say $\hat{\mathbf{p}}$ =20 and 8 in instance P8 (see Table 17) and $\hat{\mathbf{p}}$ = 8 in instance P9 (see Table 19), no CLD bounds have been obtained within the elapsed time limit, 10800 secs. However, the optimal partition is obtained by using CPLEX for partitions in $\hat{\mathbf{p}}$ =8 clusters in both instances (see Tables 18 and 20), i.e., the smallest. The optimal CLD bound is obtained by all the four procedures in instance P8 for partitions in $\hat{\mathbf{p}}$ =8 clusters by using CPLEX.

Table 17: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P8

$z_{MIP} \in$	$\hat{\mathbf{p}} = 400$ clusters $[-116.043, -113.235]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-114.531, -113.235]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 20$ clusters $[-, -113.235]$ $\alpha_0 = -$		$\hat{\mathbf{p}} = 8$ clusters $[-, -113.235]$ $\alpha_0 = -$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[73] = -114.622$	128	$z[74] = -114.371$	1152	–	–	–	–
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[57] = -114.568$	68	$z[39] = -114.361$	538	–	–	–	–
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[217] = -114.457$	296	$z[153] = -114.362$	2437	–	–	–	–
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[123] = -114.6$	165	$z[106] = -114.367$	1560	–	–	–	–

Table 18: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P8

$z_{MIP} \in$	$\hat{\mathbf{p}} = 400$ clusters $[-116.043, -114.044]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-114.689, -114.044]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 20$ clusters $[-114,531, -114.044]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-114.427, -114.044]$ $\alpha_0 = 1.9$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[126] = -114.626$	795	$z[72] = -114.382$	340	$z[22] = -114.342$	303	$z[16] = -114.324$	362
							$z[33] = -114.318$	910
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[55] = -114.573$	346	$z[43] = -114.368$	198	$z[28] = -114.34$	406	$z[24] = -114.325$	560
							$z[61] = -114.318$	1411
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[191] = -114.457$	1236	$z[99] = -114.345$	460	$z[58] = -114.333$	760	$z[57] = -114.318$	1127
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[156] = -114.623$	1026	$z[91] = -114.387$	440	$z[24] = -114.486$	312	$z[15] = -114.318$	325
							$z[37] = -114.318$	958

Table 19: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P9

$z_{MIP} \in$	$\hat{\mathbf{p}} = 400$ clusters $[-95.8124, -92.9241]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-94.4468, -92.9241]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 20$ clusters $[-94.2895, -92.9241]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-, -92.9241]$ $\alpha_0 = -$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[79] = -94.3658$	69	$z[32] = -94.1975$	176	$z[41] = -94.1482$	1278	–	–
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[53] = -94.3311$	47	$z[31] = -94.2037$	175	$z[21] = -94.1799$	601	–	–
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[247] = -94.2356$	238	$z[142] = -94.1893$	805	$z[88] = -94.1646$	2546	–	–
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[134] = -94.2895$	139	$z[60] = -94.1901$	351	$z[56] = -94.1478$	1720	–	–

Table 20: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P9

$z_{MIP} \in$	$\hat{\mathbf{p}} = 400$ clusters $[-95.8124, -94.1227]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-94.4468, -94.1227]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 20$ clusters $[-94.2895, -94.1227]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 8$ clusters $[-94.22, -94.1227]$ $\alpha_0 = 1.9$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[79] = -94.3964$	436	$z[14] = -94.2129$	44	$z[52] = -94.1726$	176	$z[11] = -94.1461$	81
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[56] = -94.3434$	302	$z[33] = -94.2048$	100	$z[29] = -94.1675$	92	$z[17] = -94.1634$	115
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[181] = -94.2522$	997	$z[107] = -94.1595$	474	$z[49] = -94.1535$	158	$z[30] = -94.1407$	192
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[129] = -94.3973$	755	$z[30] = -94.209$	93	$z[73] = -94.1793$	226	$z[12] = -94.1502$	90

P10 and P11 are the largest instances both with 500 scenarios. Tables 21-22 and 23-24 report the results. As in previous situations, when the procedures are implemented by using COIN-OR for partitions in a small number of clusters, say $\hat{\mathbf{p}}$ =10 for all procedures, but $\hat{\mathbf{p}}$ =50 for PHA in instance P11, no CLD bounds have been obtained within the elapsed time limit, 10800 secs (see Tables 21 and 23). However when using CPLEX (see Tables 22 and 24), the results are slightly different in both instances. By considering the partition in $\hat{\mathbf{p}}$ =5 clusters, the four procedures obtain the optimal solution in both instances, but VA and DCCP require more than three hours of elapsed time for instance P11. Notice that the best results for P11 are obtained for partitions in $\hat{\mathbf{p}}$ =10 clusters (see Table 24).

By considering the partition in $\hat{\mathbf{p}}$ =5 clusters, the four procedures obtain the optimal solution for instance P10 when using CPLEX (see Table 22). The optimal solution is obtained more efficiently in procedures SM, PHA and DCCP for partitions in $\hat{\mathbf{p}}$ =10 clusters, but VA stops in a red solution given just the strongest CLD bound since it coincides with the solution value of the original problem. Notice that the norm of the subgradient vector for this CLD bound is 0.015 which is slightly greater than the given tolerance ϵ_s =0.01 for the stopping criterion 1.

Table 21: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P10

$z_{MIP} \in$	$\hat{\mathbf{p}} = 500$ clusters $[-301.865, -300.166]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-300.546, -300.166]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 10$ clusters $[-, -300.166]$ $\alpha_0 = -$		$\hat{\mathbf{p}} = 5$ clusters $[-, -300.166]$ $\alpha_0 = -$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[50] = -300.506$	342	$z[18] = -300.462$	801	–	–	–	–
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[52] = -300.494$	290	$z[36] = -300.464$	1528	–	–	–	–
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[126] = -300.479$	912	$z[65] = -300.462$	2811	–	–	–	–
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[89] = -300.535$	624	$z[20] = -300.468$	906	–	–	–	–

Table 22: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P10

$z_{MIP} \in$	$\hat{\mathbf{p}} = 500$ clusters $[-301.865, -300.425]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-300.546, -300.425]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 10$ clusters $[-300.468, -300.425]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 5$ clusters $[-300.461, -300.425]$ $\alpha_0 = 1.9$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[63] = -300.5$	724	$z[13] = -300.465$	151	$z[3] = -300.459$	63	$z[0] = -300.461$	24
					$z[88] = -300.456$	1238	$z[61] = -300.456$	1249
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[56] = -300.508$	669	$z[20] = -300.473$	238	$z[6] = -300.465$	107	$z[0] = -300.461$	24
					$z[55] = -300.456$	804	$z[37] = -300.456$	780
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[117] = -300.48$	1358	$z[25] = -300.467$	292	$z[21] = -300.458$	295	$z[0] = -300.461$	24
					$z[77] = -300.456$	1065	$z[69] = -300.456$	1412
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[101] = -300.512$	1223	$z[10] = -300.474$	113	$z[3] = -300.463$	63	$z[0] = -300.461$	23
					$z[42] = -300.456$	618	$z[38] = -300.456$	817

Table 23: CLD bounds without preprocessing and parallel computation tools (COIN-OR). Instance P11

$z_{MIP} \in$	$\hat{\mathbf{p}} = 500$ clusters $[-322.35, -317.724]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-320.479, -317.724]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 10$ clusters $[-, -317.724]$ $\alpha_0 = -$		$\hat{\mathbf{p}} = 5$ clusters $[-, -317.724]$ $\alpha_0 = -$	
SM	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S	$z_{SM}[ite]$	T_S
	$z[84] = -320.416$	2324	$z[46] = -320.297$	3035.31	–	–	–	–
VA	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V	$z_{VA}[ite]$	T_V
	$z[87] = -320.391$	2562	$z[26] = -320.309$	1644.04	–	–	–	–
PHA	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P	$z_{PHA}[ite]$	T_P
	$z[186] = -320.383$	5302	–	–	–	–	–	–
DC-CP	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D	$z_{DCCP}[ite]$	T_D
	$z[107] = -320.453$	3043	$z[52] = -320.299$	3652	–	–	–	–

Table 24: CLD bounds with preprocessing and parallel computation tools (CPLEX). Instance P11

$z_{MIP} \in$	$\hat{\mathbf{p}} = 500$ clusters $[-322.35, -320.249]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 50$ clusters $[-320.479, -320.249]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 10$ clusters $[-320.326, -320.249]$ $\alpha_0 = 1.9$		$\hat{\mathbf{p}} = 5$ clusters $[-320.31, -320.249]$ $\alpha_0 = 1.9$	
SM	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S	$z^{ppc}_{SM}[ite]$	T^{ppc}_S
	$z[81] = -320.4$	1920	$z[38] = -320.301$	1100	$z[25] = -320.283$	702	$z[18] = -320.283$	2294
					$z[26] = -320.283$	732		
VA	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V	$z^{ppc}_{VA}[ite]$	T^{ppc}_V
	$z[137] = -320.393$	3528	$z[33] = -320.305$	983	$z[11] = -320.32$	307	$z[12] = -320.297$	2159
					$z[73] = -320.283$	2375	–	–
PHA	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P	$z^{ppc}_{PHA}[ite]$	T^{ppc}_P
	$z[170] = -320.361$	4519	$z[168] = -320.284$	5115	$z[59] = -320.283$	1520	$z[47] = -320.283$	5032
DC-CP	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D	$z^{ppc}_{DCCP}[ite]$	T^{ppc}_D
	$z[42] = -320.313$	773	$z[58] = -320.301$	1783	$z[27] = -320.283$	752	$z[8] = -320.283$	1343
					$z[81] = -320.283$	2806	–	–

6 Conclusions

In this paper we have presented four scenario Cluster based Lagrangian Decomposition (CLD) procedures for obtaining strong lower bounds to the solution value of two-stage stochastic mixed 0-1 problems, where the uncertainty appears anywhere in the coefficients of the 0-1 and continuous variables in the objective function and constraints in both stages. For obtaining the CLD bounds we have used three popular subgradient based procedures, namely, the traditional Subgradient Method (SM), the Volume Algorithm (VA) and the Progressive Hedging Algorithm (PHA). Additionally, we have also used the procedure Dynamic Constrained Cutting Plane (DCCP). We have used the same scheme in all of them. Two new main ideas have been incorporated in the implementation of the procedures. The first consists of the scenario cluster partitioning that allows us to compute at iteration zero of each Lagrange multiplier updating procedure, a strong lower bound for tightening the interval of the solution value of the original problem. The second idea consists of obtaining a good upper bound of this interval that is efficiently computed by the MIP solver of choice as a quasi-optimal solution for a given tolerance in relation to the best LP relaxation value in its branch-and-cut phase.

Moreover, we have given computational evidence of the model tightening effect that sophisticated preprocessing, cut generating and appending and parallel computation tools have in stochastic integer programming, by using, in this case, the MIP solver CPLEX versus the tools implemented in the COIN-OR LP/MIP functions.

The extensive computational experience reported in the paper has used small, medium, large and very large sized instances in the testbed we have experimented with (in total, 11 instances), by considering four sizes of cluster partitions. The instances are so difficult that the plain use of CPLEX cannot guarantee the optimality of the incumbent solution within the three-hour time limit, but for two toy instances. We can draw the following conclusions: (1) Very frequently the four procedures for obtaining the CLD bound give the solution value of the original stochastic mixed 0-1 problem and, in the other situations they provide a narrow interval of its solution value; (2) The performance of the CLD procedures outperforms the traditional LD scheme based on single scenarios in both the quality of the bounds and computational effort; (3) The CLD bounds obtained by both solvers (being used as auxiliary tools for solving LP/MIP submodels) are very similar for small problems, but with a higher computational effort in case of using a more sophisticated preprocessing, cut generation and appending tools, i.e., using CPLEX (where parallel computing tools are also used); (4) CPLEX outperforms COIN-OR for medium, large and very large instances, both by plain use for problem solving and as auxiliary solvers of submodels, mainly for partitions in a small number of clusters (and, then, larger MIP submodels); and (5) The efficiency of the four procedures, as contrasted in the testbed we have experimented with, is very similar in quality (i.e., tightness) to the CLD bound, but the elapsed time for obtaining it is smaller for the procedures SM and DCCP followed by VA and PHA.

As a future work, we are studying how to extend these CLD procedures to the multistage case for tightening the lower bound of the solution value of the submodels attached to a subset of the set of active Twin Node Families (TNFs) in the Branch-and-Fix phase of our Branch-and-Fix Coordination algorithm, see [9], for solving large-scale multi-stage stochastic mixed 0-1 problems. So, the LP relaxation bound (usually, a non very strong one) is to be replaced

by the CLD bound in the subset of active TNFs so-called super candidate TNFs (á là super node concept in Branch-and-Bound terminology for solving deterministic MIP problems).

Acknowledgments

This research has been partially supported by the projects PLANIN MTM2009-14087-C04-01 from Ministry of Science and Innovation, Grupo de Investigación IT-347-10 from the Basque Government, RIESGOS CM from Comunidad de Madrid and grant FPU ECO-2006 from the Ministry of Education, Spain.

We would like to express our gratitude to Prof. Dr. Fernando Tusell for making it easier to access the Laboratory of Quantitative Economics from the University of the Basque Country (UPV/EHU, Bilbao, Spain) to perform and check the computational experience reported in the paper.

References

[1] A. Alonso-Ayuso, L.F.Escudero and M.T.Ortuño. BFC, a Branch-and-Fix Coordination algorithmic framework for solving some types of stochastic pure and mixed 0-1 programs. *European Journal of Operational Research*, 151:503-519, 2003.

[2] A. Alonso-Ayuso, L.F. Escudero, A.Garín, M.T. Ortuño and G. Pérez. An approach for strategic supply chain planning based on stochastic 0–1 programming. *Journal of Global Optimization*, 26:97-124, 2003.

[3] L. Aranburu, L.F. Escudero, A. Garín and G. Pérez. Risk management for mathematical optimization under uncertainty. Part I: Multistage mixed 0-1 modeling schemes. *In preparation.*

[4] F. Barahona and R. Anbil. The volume algorithm: Producing primal solutions with a subgradient methhod. *Mathematical Programming*, 87:385–399, 2000.

[5] J.R. Birge and F.V. Louveaux. *Introduction to Stochastic Programming.* Springer, 1997.

[6] C.C. Carøe and R. Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24:37-45, 1999.

[7] L.F. Escudero, A. Garín, M. Merino and G. Pérez. A general algorithm for solving two-stage stochastic mixed 0-1 first stage problems. *Computers and Operations Research*, 36:2590-2600, 2009.

[8] L.F. Escudero, A. Garín, M. Merino and G. Pérez. An exact algorithm for solving large-scale two-stage stochastic mixed integer problems: some theoretical and experimental aspects *European Journal of Operational Research*, 204:105-116, 2010.

[9] L.F. Escudero, A. Garín, M. Merino and G. Pérez. An algorithmic framework for solving large scale stochastic mixed 0-1 problems with nonsymmetric scenario trees. *Computers and Operations Research*, 39:1133-1144, 2012.

[10] L.F. Escudero, A. Garín, G. Pérez and A. Unzueta. Lagrangian decomposition for large-scale two-stage stochastic mixed 0-1 problems *TOP*, doi: 10.1007/s11750-011.0237-1, 2012.

[11] L.F. Escudero, M. Landete and A. Rodriguez-Chia. Stochastic Set Packing Problem. *European Journal of Operational Research*, 211:232-240, 2011.

[12] A.M. Geoffrion. Lagrangean relaxation for integer programming. *Mathematical Programming Studies*, 2:82-114, 1974.

[13] M. Guignard, Lagrangian relaxation. *TOP*, 11:151–228, 2003.

[14] M. Guignard and S. Kim. Lagrangean decomposition. A model yielding stronger Lagrangean bounds. *Mathematical Programming*, 39:215-228, 1987.

[15] M. Held and R.M. Karp. The traveling salesman problem and minimum spanning trees: part II. *Mathematical Programming*, 1:6-25, 1971.

[16] IBM ILOG. CPLEX v12.2. `http://www.ilog.com/products/cplex`; 2009.

[17] INFORMS. COIN-OR. www.coin-or.org, 2010.

[18] N. Jimenez Redondo and A.J. Conejo. Short-term hydro-thermal coordination by Lagrangean relaxation: solution of the dual problem. *IEEE Transactions on Power Systems*, 14:89–95, 1997.

[19] M. Jünger, T. Liebling, D. Naddef, G. Nemhauser, W. Pulleyblank, G. Reinelt, G. Rinaldi and L. Wolsey (eds.). 50 years of Integer Programming 1958-2008. *Springer*, 2010.

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

[21] R. Schultz. Stochastic programming with integer variables. *Mathematical Programming S. B*, 97:285-309, 2003.

[22] H.D. Sherali and J.C. Smith. Two-stage hierarchical multiple risk problems: Models and algorithms. *Mathematical Programming S. A*, 120:403-427, 2009.

[23] H.D. Sherali and X. Zhu. On solving discrete two stage stochastic programs having mixed-integer first and second stage variables. *Mathematical Programming*, Series A 108:597–611, 2006.

[24] J.P. Watson and D. Woodruff. Progressive hedging innovations for a class os stochastic mixed-integer resource allocation problems. *Computational Management Science*, doi:10.1007/s10287-010-0125-4, 2012.

[25] R. Wets. Programming under uncertainty: the equivalent convex program. *SIAM Journal on Applied Mathematics*, 14:89-105, 1966.