CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

The Generalized Skeleton Solution: A New Measure of the Quality of the Deterministic Solution in Stochastic Programming

Francesca Maggioni
Teodor Gabriel Crainic
Guido Perboli
Walter Rei

June 2015

CIRRELT-2015-21

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

The Generalized Skeleton Solution: A New Measure of the Quality of the Deterministic Solution in Stochastic Programming

Francesca Maggioni[1], Teodor Gabriel Crainic[2,3,*], Guido Perboli[2,4], Walter Rei[2,3]

[1] Department of Management, Economics and Quantitative Methods, University of Bergamo, Via dei Caniana 2, 24127, Bergamo, Italy

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[3] Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

[4] Department of Control and Computer Engineering, Politecnicò di Torino, Corso Duca degli Abruzzi, 24 - I-10129 Torino, Italy

Abstract. Stochastic programs and stochastic integer ones in particular, are usually hard to solve when applied to realistic sized problems. A common approach is to consider the simpler deterministic program in which random parameters are replaced by their expected values, with a loss in terms of the quality solution. In this paper we investigate the reason of the gap between the deterministic and the stochastic solutions and which information can be inherited by the deterministic solution also in such a cases. In details, this paper provides a comprehensive understanding of the structure of the optimal solution of stochastic problems and its links to the one of the corresponding deterministic version (or its linear relaxation for integer formulations). A new measure of goodness/badness of the deterministic solution with respect to the stochastic one, namely Generalized Loss Using Skeleton Solution, GLUSS, is introduced. It is based on the reduced costs of the deterministic solution, which enable the identification of the good variables to inherit. The possible usage of GLUSS, as well as its interest and value in addressing stochastic programming models, are investigated by means of an extensive experimental campaign.

Keywords: Stochastic programming, value of stochastic solution, skeleton solution, generalized skeleton solution, expected value solution.

Acknowledgements. Partial funding for this project was provided by the Italian University and Research Ministry under the UrbeLOG project-Smart Cities and Communities. The first author acknowledges financial support from the grants "Fondi di Ricerca di Ateneo di Bergamo 2014", managed by Francesca Maggioni. Partial funding for this project has also been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), through its Discovery Grants program. We also gratefully acknowledge the support of the Fonds de recherche du Québec – Nature et technologie (FRQNT) through their infrastructure grants.

---

* Corresponding author: TeodorGabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2015

1 Introduction

Stochastic programs, in particular stochastic integer programs, are often close to impossible to solve for realistically sized problems. Thus, even though a stochastic approach modeling is appropriate, all we may have access to, is the deterministic solution (where all random variables are replaced by their means). In stochastic programming, a standard measure of the expected gain from solving a stochastic model rather than its deterministic counterpart is given by the Value of the Stochastic Solution – *VSS* [4]. A high value of *VSS* is mostly used to argue that stochastic programming models are necessary despite the computational efforts involved, since uncertainty is important for the optimal solution. Consequently, the deterministic solution is considered "bad". However, we think that stopping at this point is a bit simplistic and we should go deeper to understand why the deterministic solution is inappropriate.

A qualitative understanding of the expected value solution with respect to the stochastic one, also in case of high *VSS*, has been analyzed in [12] in terms of its structure and upgradeability, by means of the *Loss Using the Skeleton Solution* $LUSS$ and the *Loss of Upgrading the Deterministic Solution* $LUDS$ in relation to the standard *VSS*. Compared to the *VSS*, $LUSS$ and $LUDS$ give broader information on the structure of the problem and could be of practical relevance to make a fast "good" decision instead of using expensive direct techniques. In detail, they work in the following way: $LUSS$ is obtained by fixing at zero (or at the lower bound) all first stage variables which are at zero (or at the lower bound) in the expected value solution and then solve the stochastic program. Hence, $LUSS$ allows to see if the deterministic model produced the right non-zero variables, but possibly was off on the values of the basic variables. On the other hand, $LUDS$ is measured by first solving a restricted stochastic model obtained by fixing the lower bound of all variables to their corresponding values in the expected value solution. $LUDS$ is then defined as the difference between the optimal value of this restricted stochastic model and the optimal value of the original stochastic problem. Therefore, $LUDS$ tests if the expected value solution is upgradeable to become good (if not optimal) in the stochastic setting. From these results the main causes of badness/goodness of the expected value solution can be summarized as follows:

- Wrong choice of variables, that is, different variables are set to zero (or at the lower bound) in the deterministic and the stochastic solutions, measured by a positive loss using the skeleton solution $0 < LUSS \leq VSS$.
- Wrong values, when the choice of variables is the same but the values of the non-zeros differ; this case is reflected by $LUSS = 0$ and $VSS > 0$. Obviously, a wrong choice of variables leads to wrong values too ($LUSS > 0$). Situations where the skeleton is good, but the deterministic solution is bad, are of particular interest as the deterministic solution is very useful.
- Non-upgradeability of the deterministic solution to the stochastic measured by a positive loss of upgrading the deterministic solution $LUDS > 0$.

In this paper, we seek a deeper understanding of the expected value solution and the possible links between the deterministic and stochastic solutions. What could be identified as "inherited" from the former to the latter? For example, can we identify a subset of variables with zero value in the deterministic solution to fix at zero in the stochastic formulation in order to guide the search for the optimal stochastic solution? In the affirmative, are the *reduced costs* of the continuous relaxation of the deterministic solution a good estimation of bad/good variables? Can we infer a general trend from the cases considered or the behavior of the deterministic solution is problem dependent? Such understanding could help predict how the stochastic model will perform in two important cases, problems actually solvable but that must be run very often, and intractable problems, and thus reduce the computational time of the stochastic solution process. This applies to algorithmic developments as well as practical use of models in industry and government. In essence, we seek the means to identify what is potentially wrong with a solution coming from a deterministic model even when the optimum of the stochastic formulation cannot be found within reasonable computing effort.

To achieve our overall goal, we introduce the *Generalized Loss Using the Skeleton Solution*, *GLUSS*, a measure of the badness/goodness of deterministic solutions based on the information brought by the reduced costs of the continuous relaxation of the deterministic solution. *GLUSS* helps to identify the good variables that should be inherited from the deterministic solution and thus, provides better insights, when compared to *VSS* and *LUSS*, into what defines the structure of the solution to the stochastic programming model. Basic inequalities in relation to *VSS* and *LUSS* are also presented. A large set of problems from the literature are analyzed using *GLUSS*, the results illustrating its performance and interest. To keep the paper length within reasonable limits, our analysis is focused on the two-stage stochastic case. This being said, the proposed approach and investigation can easily be extended to multi-stage models as well.

To sum up, the main contributions of this paper are

1. To provide a more comprehensive understanding of the structure of the optimal solution of stochastic problems and its links to the optimal solution of the corresponding deterministic version (its linear relaxation for integer formulations);
2. To define a new measure of goodness/badness of the deterministic solution with respect to the stochastic formulation (that complements the information provided by *VSS* and *LUSS*), which is based on the reduced costs in the deterministic solution and that enables the identification of the good variables to inherit from the solution;
3. To provide an extensive experimental campaign that illustrates possible utilization of *GLUSS* and underlines its interest and value in addressing stochastic programming models.

The paper is organized as follows. The state of the art is reviewed in Section 2, while

Section 3 defines the $GLUSS$ measure. The experimental plan is described in Section 4, including how we use $GLUSS$ and the problems and formulations considered in the experimentation. Numerical results are presented and analyzed in Section 5. In Section 6, we sum up the highlights and general trends observed from the case studies considered. Finally, we conclude in Section 7.

2 State of the Art

Let us define the standard notation used in this paper. The following mathematical model represents a general formulation of a stochastic program [6]-[3] in which a decision maker needs to determine x in order to minimize (expected) costs or outcomes:

$$\min_{x\in X} E_{\boldsymbol{\xi}} z\left(x,\boldsymbol{\xi}\right) = \min_{x\in X}\left\{f_1(x) + E_{\boldsymbol{\xi}}\left[h_2\left(x,\boldsymbol{\xi}\right)\right]\right\}, \tag{1}$$

where x is a first-stage decision vector restricted to the set $X \subset \mathbb{R}^n_+$ (where $\mathbb{R}^n_+$ is the set of non negative real vectors of dimension n), $E_{\boldsymbol{\xi}}$ denotes the expectation with respect to a random vector $\boldsymbol{\xi}$, defined on some probability space $(\Omega, \mathscr{A}, p)$ with support Ω and given probability distribution p on the $\sigma-$algebra $\mathscr{A}$. The function h_2 is the value function of another optimization problem defined as

$$h_2\left(x,\xi\right) = \min_{y\in Y(x,\xi)} f_2\left(y; x,\xi\right), \tag{2}$$

which is used to reflect the costs associated with adapting to information revealed through a realization ξ of the random vector $\boldsymbol{\xi}$. The term $E_{\boldsymbol{\xi}}\left[h_2\left(x,\boldsymbol{\xi}\right)\right]$ in (1) is referred to as the recourse function. We make the assumption in this paper that functions f_1 and f_2 are linear in their unknowns. The solution x^* obtained by solving problem (1), is called the *here and now solution* and

$$RP = E_{\boldsymbol{\xi}} z(x^*, \boldsymbol{\xi}), \tag{3}$$

is the optimal value of the associated objective function.

A simpler approach is to consider the *Expected Value Problem*, where the decision maker replaces all random variables by their expected values and solves a deterministic program:

$$EV \quad = \quad \min_{x\in X} z(x, \bar{\xi}), \tag{4}$$

where $\bar{\xi} = E(\boldsymbol{\xi})$. Let $\bar{x}(\bar{\xi})$ be an optimal solution to (4), called the *Expected Value Solution* and let EEV be the expected cost when using the solution $\bar{x}(\bar{\xi})$:

$$EEV = E_{\boldsymbol{\xi}}\left(z\left(\bar{x}(\bar{\xi}), \boldsymbol{\xi}\right)\right). \tag{5}$$

The *Value of the Stochastic Solution* is then defined as

$$VSS = EEV - RP, \tag{6}$$

measuring the expected increase in value when solving the simpler deterministic model rather than its stochastic version. Relations and bounds on EV, EEV and RP can be found for instance in [4] and [3].

It is well known that, in general, the expected value solution can behave very badly in a stochastic environment. We studied empirically the structural differences between the two solutions within the context of particular combinatorial optimization problems [7, 15, 17, 16], observing both the general bad behaviour of the expected value solution and hints that some structures from the deterministic solution find their way into the stochastic one. Thus, it is generally not clear where this badness comes from: is it because the *wrong variables* are fixed at non-zero levels or because they have been assigned *wrong values*? We tried to answer this question in [12] by means of the *Loss Using Skeleton Solution LUSS*. The extension to the multi-stage setting is in [8]. We fix at zero (or at the lower bound) all first stage variables which are at zero (or at the lower bound) in the expected value solution (i.e., for linear programs, the non basis variables) and then solve the stochastic program. Hence, we want to see if the deterministic model produced the right non-zero variables (activities), but possibly was off on the values.

Let $\mathcal{J} = \{1, \ldots, J\}$ be the set of indices for which the components of the expected value solution $\bar{x}(\bar{\xi})$ are at zero or at their lower bound (non basis variables). Then let $\hat{x}$ be the solution of:

$$\begin{array}{ll} \min_{x \in X} & E_{\boldsymbol{\xi}} z\left(x, \boldsymbol{\xi}\right) \\ \text{s.t.} & x_j = \bar{x}_j(\bar{\xi}),\ j \in \mathcal{J}. \end{array} \tag{7}$$

We then compute the *Expected Skeleton Solution Value*

$$ESSV = E_{\boldsymbol{\xi}}\left(z\left(\hat{x}, \boldsymbol{\xi}\right)\right), \tag{8}$$

and we compare it with RP by means of the *Loss Using Skeleton Solution*

$$LUSS = ESSV - RP. \tag{9}$$

A $LUSS$ close to zero means that the variables chosen by the expected value solution are the correct ones but their values may be off. We have:

$$RP \leq ESSV \leq EEV, \tag{10}$$

and consequently,

$$VSS \geq LUSS \geq 0. \tag{11}$$

Notice that the case $LUSS = 0$ corresponds to the *perfect skeleton solution* in which the condition $x_j = \bar{x}_j(\bar{\xi}),\ j \in \mathcal{J}$, is satisfied by the stochastic solution x^* even without being enforced by a constraint (i.e., $\hat{x} = x^*$); on the other hand, $0 < LUSS < VSS$ if there exists $j \in \mathcal{J}$ such that $x_j^* \neq \bar{x}_j(\bar{\xi})$. Finally, one observes $LUSS = VSS$, if the $\hat{x} = \bar{x}(\bar{\xi})$.

Notice also that if the original problem is a:

- Linear program, then $ESSV$ leads to solving a linear program but of smaller size than the original one;
- Mixed binary program, then the test implies fixing all the binary variables (at 0 or 1) and solving an easier linear program;
- Mixed integer program (MIP), then we still solve a MIP but of smaller dimension.

3 Does the Stochastic Solution Inherit Properties from the Deterministic One?

One of the main contributions of this paper is to provide a tool to analyze and compare the expected value solution with respect to the stochastic one. The EEV is often far from the optimal stochastic solution, meaning that the expected value and the stochastic solutions are different. The research questions we address are: Can we derive knowledge relative to the stochastic solution from the deterministic one? Can we guide the stochastic problem towards an optimal solution starting with the information given by the expected value model?

To answer these questions, we introduce a new measure, the *Generalized Loss Using the Skeleton Solution*, *GLUSS*, which is a generalization of the *Loss Using Skeleton Solution LUSS* introduced in [12]. $GLUSS$ allows us to investigate, even in the case of a large VSS, what can be inherited from the structure of the expected value solution in its stochastic counterpart, by taking into account the information on reduced costs associated to the variables at zero (or lower bound) in the expected value solution.

Let $\mathscr{R} = \{r_1, \ldots, r_j, \ldots, r_J\}$ be the set of *reduced costs*, with respect to the recourse function, of the components $\bar{x}_j(\bar{\xi}),\ j \in \mathcal{J}$, of the expected value solution $\bar{x}(\bar{\xi})$ at zero or at their lower bound (i.e., out of basis variables). We recall that a reduced cost is the amount by which an objective function coefficient would have to improve (increase, for maximization problems and decrease for minimization ones) before it would be possible for the corresponding variable to assume a positive value in the optimal solution and become a basis variable. Reduced costs of basis variables are zero.

Let $r^{max} = \max_{j \in \mathcal{J}}\{r_j : r_j \in \mathscr{R}\}$ and $r^{min} = \min_{j \in \mathcal{J}}\{r_j : r_j \in \mathscr{R}\}$ be respectively the maximum and the minimum of the reduced costs of the variables $\bar{x}_j(\bar{\xi}),\ j \in \mathcal{J}$. We divide the difference $r^{max} - r^{min}$ into N classes $\mathscr{R}_1, \ldots, \mathscr{R}_N$ of constant width $\frac{r^{max}-r^{min}}{N}$ such that the $p-$class is defined as follows

$$\mathscr{R}_p = \left\{ r_j : r^{min} + (p-1) \cdot \frac{(r^{max} - r^{min})}{N} \leq r_j \leq r^{min} + p \cdot \frac{(r^{max} - r^{min})}{N} \right\}, \quad p = 1, \ldots, N. \tag{12}$$

Let $\mathcal{J}_p$ be the set of indices associated to the variables $\bar{x}_j(\bar{\xi})$ with reduced costs $r_j \in \mathscr{R}_p$. Then let $\tilde{x}_p$ be the solution of

$$\begin{array}{ll} \min_{x \in X} & E_{\boldsymbol{\xi}} z\left(x, \boldsymbol{\xi}\right) \\ \text{s.t.} & x_j = \bar{x}_j(\bar{\xi}), \ j \in \mathcal{J}_p, \ldots, \mathcal{J}_N, \end{array} \tag{13}$$

where we fix at zero or lower bounds only the variables with indices belonging to the last p classes $\mathcal{J}_p, \ldots, \mathcal{J}_N$, i.e., with the highest reduced costs.

We then compute the *Generalized Expected Skeleton Solution Value*

$$GESSV(p, N) = E_{\boldsymbol{\xi}}\left(z\left(\tilde{x}_p, \boldsymbol{\xi}\right)\right) \ , \quad p = 1, \ldots, N, \tag{14}$$

and we compare it with RP by means of the *Generalized Loss Using the Skeleton Solution*

$$GLUSS(p, N) = GESSV(p, N) - RP \ , \quad p = 1, \ldots, N. \tag{15}$$

Notice that $GESSV(1, N) = ESSV$ and consequently $GLUSS(1, N) = LUSS$. Furthermore, the following inequalities hold true:

Proposition 1. *For a fixed $N \in \mathbb{N} \setminus \{0, 1\}$ (where $\mathbb{N}$ is the set of natural numbers),*

$$GLUSS(p, N) \geq GLUSS(p+1, N) \ , \quad p = 1, \ldots, N-1. \tag{16}$$

Proof
Any feasible solution of problem $GESSV(p, N)$ is also a solution of problem $GESSV(p+1, N)$, since the former is more restricted than the latter, and the relation (16) holds true. If $GLUSS(p, N) = \infty$, the inequality is automatically satisfied. □

Proposition 2. *For a given $N \in \mathbb{N} \setminus \{0\}$ and a fixed $p \in \mathbb{N} \setminus \{0\}$ such that $p = 1, \ldots, N$,*

$$GLUSS(p, N+1) \geq GLUSS(p, N). \tag{17}$$

Proof
If $p = 1$ then $GLUSS(p, N+1) = GLUSS(p, N) = LUSS$. Furthermore, any feasible solution of problem $GESSV(p, N+1)$ is also a solution of problem $GESSV(p, N)$, since the former is more restricted than the latter, and the relation (17) holds true. If $GLUSS(p, N+1) = \infty$, the inequality is automatically satisfied. □

The two previous properties can be generalized in the following proposition:

Proposition 3. *For given* $N_1, N_2 \in \mathbb{N}\setminus\{0\}$ *and* $p_1, p_2 \in \mathbb{N}\setminus\{0\}$*, with* $p_1 = 1, \ldots, N_1$*,* $p_2 = 1, \ldots, N_2$ *and such that* $\frac{p_1}{N_1} \leq \frac{p_2}{N_2}$

$$GLUSS(p_1, N_1) \geq GLUSS(p_2, N_2). \tag{18}$$

Proof
If $p_1 = p_2 = 1$ then $GLUSS(p_1, N_1) = GLUSS(p_2, N_2) = LUSS$. Furthermore, if $\frac{p_1}{N_1} \leq \frac{p_2}{N_2}$ then the number of variables at zero with highest reduced cost to be fixed is respectively $\frac{N_1 - p_1}{N_1}|\mathscr{R}| \geq \frac{N_2 - p_2}{N_2}|\mathscr{R}|$. Consequently $GESSV(p_1, N_1)$ is more restricted than $GESSV(p_2, N_2)$, and the relation (18) holds true. □

In the following, we make the assumption that in the case of a problem with first stage integer variables, we compute the reduced costs on the continuous relaxation.

$GLUSS$ measures how much we loose in terms of solution quality when we consider the generalized skeleton solution. But how can one use it in order to analyze and derive the structure of the stochastic solution? How should we choose the number of classes N and p? We answer these questions in the following sections, by presenting a possible procedure using $GLUSS$ and applying it to a wide set of problems from the literature.

4 Experimental plan and instance sets

This section describes the experimental plan and the instance sets considered. Our goal is to study the possibility to use $GLUSS$ and inject information about the skeleton of the stochastic solution from the expected value one. We therefore perform an experimental analysis according to three axes:

- *Computational effort*: what number of variables can we can fix in order to drastically reduce the effort of the stochastic solution computation?
- *Feasibility*: what are the effects of fixing a subset of the variables from the expected value solution with regards to the feasibility of the stochastic model?
- *Optimality*: how to use the $GLUSS$ to find an optimal or near optimal stochastic solution?

The experimental campaign was carried out using two main sets of problem instances. The first set includes three stochastic optimization models related to three real-case applications: a single-sink transportation problem, a power generation scheduling case, and a supply transportation problem. The second set consists of instances from SIPLIB [1], which is a collection of standard test problems to facilitate computational and algorithmic research in stochastic integer programming. The original instances included in this collection were of limited size. However, SIPLIB has recently been updated with a set of large-scale highly combinatorial stochastic problems that are relevant to the field of city logistics. As benchmarks, available solutions for some of the instances included in the collection are also reported in [1]. Finally, it should be noted that all numerical experiments were conducted using CPLEX 12.5.

Section 4.1 presents our methodology, including a proposed approach to set up the number of classes N and the class parameter p of $GLUSS(p, N)$. The following four subsections describe the test problems. In each subsection we describe the problem and we give the mathematical problem. The corresponding numerical data are summarized in Annex 1.

4.1 Using $GLUSS$ in stochastic programming

We computed VSS, $LUSS$ and $GLUSS(p, N)$ for each instance set. When available, the optimal solutions from the literature were used, otherwise, we computed them. We now briefly describe the procedure we adopted, which can be applied and extended to any stochastic programming problem.

Let us recall that parameter N defines the number of sets in which the out of basis variables of $\bar{x}(\bar{\xi})$ are grouped. These sets provide a characterization of the variables with respect to their reduced costs. The higher the value of N is and the closer the reduced cost values of the variables included in each set are. Therefore, we start by considering only three classes, $N = 3$, with the exception of the "single-sink transportation problem" for which only two variables in the deterministic solution are at zero (we set $N = 2$). The use of three classes defines a rough characterization, where the out of basis variables of $\bar{x}(\bar{\xi})$ are included in a higher, lower or medium range reduced cost set. It should be noted that a sensitivity analysis where parameter N is increased was performed using the "power generation scheduling" problem (for $N = 4$ given the four variables that are fixed to zero in the deterministic solution) and the "supply transportation" problem (for $N = 3, 10, 50, 100$), which enabled a more refined study.

For a given value N, generating sets $\mathscr{R}_1, \dots, \mathscr{R}_N$ and the partition of the variables $\mathcal{J}_1, \dots, \mathcal{J}_N$, our objective is to identify which out of basis variables of $\bar{x}(\bar{\xi})$ should be fixed in the stochastic model to produce an optimal, or near-optimal, solution. To do so, the parameter p is first fixed to its upper limit (i.e., $p = N$) to compute $GLUSS(N, N)$. We know, from Proposition 1 that, for a fixed N, $GLUSS(p, N)$ can only increase when p decreases. Therefore, parameter p is then iteratively decreased by a value of one as long as the following condition is verified: $GLUSS(p, N) = GLUSS(p - 1, N)$. This process is stopped when parameter p reaches a certain value $\overline{p}$ such that $GLUSS(\overline{p}, N) > GLUSS(\overline{p} + 1, N)$. At this point, we define $\overline{\mathcal{J}} = \bigcup_{p=\overline{p}+1,\dots,N} \mathcal{J}_p$ as the set of out of basis variables of $\bar{x}(\bar{\xi})$ that should be fixed to their respective limit in the stochastic model. The process to obtain $\overline{\mathcal{J}}$ is summarized in Algorithm 1. The procedure begins by initializing parameter p and set $\overline{\mathcal{J}}$ (lines: 1 and 2). In the main loop (lines: 3 to 9), set $\overline{\mathcal{J}}$ is updated until either the following condition is observed: $GLUSS(p, N) > GLUSS(p+1, N)$, or, parameter p reaches the value zero.

It is important to realize that the value to which parameter N is fixed greatly influences the results obtained by Algorithm 1 (i.e., set $\overline{\mathcal{J}}$), as well as the overall numerical effort involved in applying the procedure. As previously mentioned, a high value of N leads to a more refined characterization of the out of basis variables. In turn, this will help to include in $\overline{\mathcal{J}}$ a larger number of these variables. However, a refined characterization also entails a higher computational effort to run the algorithm. For one, measuring $GLUSS(p, N)$ for high values of p will be more time consuming, considering that the restriction defined by $x_j = \bar{x}_j(\bar{\xi}),\ j \in \mathcal{J}_p, \ldots, \mathcal{J}_N$ will tend to be smaller. Furthermore, the $GLUSS$ measure will be evaluated a potentially higher number of times, given the larger number of considered classes $\mathscr{R}_1, \ldots, \mathscr{R}_N$. Therefore, a careful analysis should be applied to find the appropriate value of N for the specific problem being solved. Towards this end, we should mention that the proposed approach can be recursively applied on any class $\mathscr{R}_1, \ldots, \mathscr{R}_N$. Therefore, a specific class can be further divided into subclasses to perform a more systematic localized exploration.

Algorithm 1 Defining set $\overline{\mathcal{J}}$

Require: N, $\mathcal{J}_1, \ldots, \mathcal{J}_N$
1: $p = N$
2: $\overline{\mathcal{J}} = \mathcal{J}_p$
3: **while** $p > 1$ **do**
4: $\quad p = p - 1$
5: $\quad$ **if** $GLUSS(p, N) > GLUSS(p+1, N)$ **then**
6: $\qquad$ **return** $\overline{\mathcal{J}}$
7: $\quad$ **end if**
8: $\quad \overline{\mathcal{J}} = \overline{\mathcal{J}} \cup \mathcal{J}_p$
9: **end while**
10: **return** $\overline{\mathcal{J}}$

4.2 A single-sink transportation problem

This problem is inspired by a real case of *clinker* replenishment, provided by the largest Italian cement producer located in Sicily [9]. The logistics system is organized as follows: *clinker* is produced by four plants located in Palermo (PA), Agrigento (AG), Cosenza (CS) and Vibo Valentia (VV) and the warehouse to be replenished is in Catania. The production capacities of the four plants, as well as the demand for clinker at Catania, are considered stochastic.

All the vehicles are leased from an external transportation company, which we assume to have an unlimited fleet. The vehicles must be booked in advance, before the demand and production capacities are revealed. Only full-load shipments are allowed. When the demand and the production capacity become known, there is an option to cancel some of the bookings against a cancellation fee α. If the quantity delivered from the four suppliers using the booked vehicles is not enough to satisfy the demand in Catania, the residual quantity is purchased from an external company at a higher price b. The problem is to determine, for each supplier, the number of vehicles to book in order to minimize the total costs, given by the sum of the transportation costs (including the cancellation fee for vehicles booked but not used) and the costs of the product purchased from the external company. The notation adopted is:

$$\begin{aligned} \mathscr{I} = \{i : i = 1, \dots, I\} &\quad : \quad \text{set of suppliers (AG, CS, PA, VV) ;} \\ \mathscr{S} = \{s : s = 1, \dots, S\} &\quad : \quad \text{set of scenarios.} \end{aligned}$$

t_i : unit transportation costs of supplier $i \in \mathscr{I}$;
c_i : unit production costs of supplier $i \in \mathscr{I}$;
b : buying cost from an external source (we assume that $b > \max_i(t_i + c_i)$) ;
q : vehicle capacity ;
g : maximum capacity that can be booked ;
l_0 : initial inventory level at the customer ;
$l_{\max}$: storage capacity at the customer ;
p^s : probability of scenario $k \in \mathscr{S}$;
a_i^s : production capacity of supplier $i \in \mathscr{I}$ in scenario $s \in \mathscr{S}$;
d^s : customer demand at scenario $s \in \mathscr{S}$;
α : cancellation fee ;

with the decision variables

$x_i \in \mathbb{N}$: number of vehicles booked from supplier $i \in \mathscr{I}$;
$z_i^s \in \mathbb{N}$: number of vehicles actually used from $i \in \mathscr{I}$ in $s \in \mathscr{S}$;
y^s : product to purchase from an external source in scenario $s \in \mathscr{S}$;

In the two-stage (one-period) case, we get the following mixed-integer stochastic programming model with recourse:

$$\min \; q\sum_{i=1}^{I} t_i x_i + \sum_{s=1}^{S} p^s \Big[b\, y^s - (1-\alpha) q \sum_{i=1}^{I} t_i \,(x_i - z_i^s) \Big] \tag{19}$$

$$
\begin{aligned}
\text{s.t.} \qquad q\sum_{i=1}^{I} x_i &\leq g, && (20)\\
l_0 + \sum_{i=1}^{I} q z_i^s + y^s - d^s &\geq 0\,, \qquad s \in \mathscr{S}, && (21)\\
l_0 + \sum_{i=1}^{I} q z_i^s + y^s - d^s &\leq l_{\max}\,, \qquad s \in \mathscr{S}, && (22)\\
z_i^s &\leq x_i\,, \qquad i \in \mathscr{I},\ s \in \mathscr{S}, && (23)\\
q z_i^s &\leq a_i^s\,, \qquad i \in \mathscr{I},\ s \in \mathscr{S}, && (24)\\
x_i &\in \mathbb{N}\,, \qquad i \in \mathscr{I}, && (25)\\
y^s &\geq 0\,, \qquad s \in \mathscr{S}, && (26)\\
z_i^s &\in \mathbb{N}\,, \qquad i \in \mathscr{I},\ s \in \mathscr{S}. && (27)
\end{aligned}
$$

The first sum in the objective function (19) is the booking costs of the vehicles, while the second sum represents the expected cost associated to the recourse actions, consisting of buying extra clinker (y^s) and canceling unwanted vehicles. Constraint (20) guarantees that the number of booked vehicles from the suppliers to the customer is not greater than g/q. Constraints (21) and (22) ensure that the second-stage storage level is between zero and $l_{\max}$. Constraints (23) guarantee that the number of vehicles serving supplier i is at most equal to the number of vehicles booked in advance, and constraints (24) control that the quantity of clinker delivered from supplier i does not exceed its production capacity a_i^s. Finally, (25)–(27) define the decision variables of the problem (both for the first and second stages).

The goal is to find, for each supplier, the number of vehicles to book at the beginning of the first period.

4.3 Power generation scheduling

The second real-case problem is based on an economic scheduling model formulated in [18] and [5] as a deterministic mixed integer program. Power generation scheduling involves the selection of generating units to be put into operation and the allocation of the power demand among the units over a set of time periods. In the problem considered, there are two types of generating units available (i.e., four units of type 1 and four units of type 2). Each type is defined according to specific technical characteristics and operational costs. Therefore, a generating unit will run at a level that is between a minimum and a maximum threshold, these threshold values being type specific. When a unit is used, there is a base hourly cost that is charged for running it at the minimum level. In the case where a unit runs above the minimum threshold, an extra hourly cost is applied for each additional megawatt. There is also a starting up cost that is charged each time a new generating unit is used. Once again, all specific cost values vary according to the unit types.

At any considered time period, there must be a sufficient number of operating generators to meet a possible increase in the overall demand of up to 15%. In the event of an increase, the running levels of the used units are simply adjusted to meet the new demand requirements. In the present problem, two time periods are considered. While the demands of the first time period are assumed known, the demands in the second time period are stochastic. Therefore, the problem is formulated as a two-stage stochastic model. In the first stage, a set of generating units are chosen and their operating levels are fixed for the two time periods defined in the problem (an estimate is used here for the demands in the second period). In the second stage, the actual values of the demands in the second period are observed and the number units and their operating levels are adjusted accordingly. Production decisions are thus made after the demands have been revealed. Instead of writing the model in terms of scenarios, we consider a node formulation defined on the structure of the scenario tree (see Table 19 in Annex). Therefore, nodes $n = 1, 2$ represent the first stage of the model, while nodes $n = 3, \ldots, 22$ define the 20 considered scenarios that can be observed in the second stage. For each node n in the scenario tree, value $pa(n)$ defines its predecessor.

We now define the model that is considered. To do so, let us first define the general notation that is used:

$$\begin{aligned} \mathscr{I} = \{i : i = 1, \ldots, I\} &\quad : \quad \text{types of generating units;} \\ \mathscr{N} = \{n : n = 1, \ldots, N\} &\quad : \quad \text{ordered set of nodes of the scenario tree;} \end{aligned}$$

m_i : minimum output level for generator of type $i \in \mathscr{I}$;
M_i : maximum output level for generator of type $i \in \mathscr{I}$;
D^n : demand in node $n \in \mathscr{N}$;
p^n : probability of node $n \in \mathscr{N}$;
C_i : cost per hour per megawatt (mw) of unit $i \in \mathscr{I}$ for operating above minimum level;
E_i : cost per hour per megawatt (mw) of unit $i \in \mathscr{I}$ for operating at minimum level;
F_i : start-up cost of unit $i \in \mathscr{I}$;
$u_{i,max}$: upper bound on the total number of generators of type $i \in \mathscr{I}$;
u_i^0 : starting value of open units of type $i \in \mathscr{I}$;

The decision variables are:

u_i^n : number of generating units of type $i \in \mathscr{I}$ working in node $n \in \mathscr{N}$;
s_i^n : number of generators of type $i \in \mathscr{I}$ started up in node $n \in \mathscr{N} \setminus \{1\}$;
x_i^n : total output rate from generators of type $i \in \mathscr{I}$ in node $n \in \mathscr{N}$;

The formulation of the generator scheduling problem as an integer program including start-up costs is now defined as follows:

$$\sum_{n \in \mathscr{N}} p^n \left[\sum_{i \in \mathscr{I}} C_i \left(x_i^n - m_i u_i^n \right) + \sum_{i \in \mathscr{I}} E_i u_i^n + \sum_{i \in \mathscr{I}} F_i s_i^n \right] \tag{28}$$

$$\begin{aligned}
\text{s.t.} \qquad \sum_{i \in \mathscr{I}} x_i^n \;&\geq\; D^n, \qquad n \in \mathscr{N}, && (29)\\
x_i^n \;&\geq\; m_i u_i^n \;, \qquad i \in \mathscr{I},\; n \in \mathscr{N}, && (30)\\
x_i^n \;&\leq\; M_i u_i^n \;, \qquad i \in \mathscr{I},\; n \in \mathscr{N}, && (31)\\
\sum_{i \in \mathscr{I}} M_i u_i^n \;&\geq\; \frac{115}{110} D^n \;, \qquad n \in \mathscr{N}, && (32)\\
s_i^n \;&\geq\; u_i^n - u_i^{pa(n)} \;,\; i \in \mathscr{I} \;,\; n \in \mathscr{N} \setminus \{1\}, && (33)\\
u_i^1 \;&=\; u_i^0 \;, \qquad i \in \mathscr{I}, && (34)\\
u_i^n \;&\leq\; u_{i,max} \;, \qquad i \in \mathscr{I},\; n \in \mathscr{N}, && (35)\\
x_i^n \;&\geq\; 0 \;, \qquad i \in \mathscr{I},\; n \in \mathscr{N}, && (36)\\
s_i^n \;&\in\; \mathbb{N} \;, \qquad i \in \mathscr{I},\; n \in \mathscr{N}, && (37)\\
u_i^n \;&\in\; \mathbb{N} \;, \qquad i \in \mathscr{I},\; n \in \mathscr{N}. && (38)
\end{aligned}$$

The objective function (28) consists in the minimization of the total costs, which include the starting up costs of units and their operational costs (both at the minimum level and above it) for each period. Constraints (29) guarantee that the demand in each period is met, whereas (30) and (31) make sure that the output lies within the limits of the operating generators at all times. Constraints (32) guarantee that, for each period, the additional load requirement of 15% is met without the need to resort to additional generators, while constraints (33) ensure that the number of generators that are started up in node n be equal to the increase in the number of operating units with respect to the node $pa(n)$. Finally, constraints (34)-(35) define the starting values and upper bounds for the number of started up units and (36)-(38) impose the necessary non-negativity and integrality requirements on the decision variables of the problem.

4.4 Supply transportation problem

This problem is inspired by a real case of *gypsum* replenishment in Italy, provided by the primary Italian cement producer, see [11] for more details. The logistic system is organized as follows: 24 suppliers, each of them having several plants located all around Italy, are used to satisfy the demand for gypsum of 15 cement factories belonging to the same company. The demands for gypsum at the 15 cement factories are considered stochastic. As in the first problem considered, shipments are performed by capacitated vehicles, which have to be booked in advance, before the demand is revealed. When the demands become known, there is the option to discount vehicles that were booked but not actually used. However, if the quantity shipped from the suppliers using the booked vehicles is not enough to satisfy the observed demands, vehicle services to transport the extra demand of gypsum directly to the factories can be purchased from an external company at a premium price. The problem is to determine for each of the supplier plants, the number of vehicles to book to replenish in gypsum the factories in order to minimize the total cost. The total cost is defined as the sum of the booking costs for the vehicles used to perform the distribution operations between the plants and the factories (including the discount for vehicles booked but not used), and the costs of the extra vehicles added to satisfy the observed demand. It should be noted that, in all cases, the cost of a vehicle is obtained by multiplying its capacity by a unit cost that either reflects the transportation cost between a plant and a factory (for the vehicles booked in advance), or, the premium rate charged by the external company for a direct transportation service to a factory. Regarding the discount for the vehicles booked but not used, it is expressed as a fixed percentage of the cost associated to the number of unused vehicles.

The notation adopted is the following:

$\mathscr{K} = \{k : k = 1, \ldots, K\}$: set of suppliers;
$\mathscr{O}_k = \{i : i = 1, \ldots, O_k\}$: set of plant locations of supplier $k \in \mathscr{K}$;
$\mathscr{D} = \{j : j = 1, \ldots, D\}$: set of cement factories (destinations);
$\mathscr{S} = \{s : s = 1, \ldots, S\}$: set of scenarios;

t_{ij} : unit transportation cost from plant $i \in \mathscr{O}_k, k \in \mathscr{K}$ to factory $j \in \mathscr{D}$;
b_j : the premium rate charged by the external company for a vehicle assigned to factory $j \in \mathscr{D}$;
q : the capacity of a vehicle;
g_j : maximum capacity which can be booked for the factory $j \in \mathscr{D}$;
v_k : maximum requirement capacity of supplier $k \in \mathscr{K}$;
a_k : minimum requirement capacity of supplier $k \in \mathscr{K}$;
$l_{\max}$: storage capacity at the factories;
α : discount;
p^s : probability of scenario $s \in \mathscr{S}$;
d_j^s : demand of factory j in scenario $s \in \mathscr{S}$.

The decision variables are

$x_{ij} \in \mathbb{N}$: number of vehicles booked between plant $i \in \mathscr{O}_k,\ k \in \mathscr{K}$ and factory $j \in \mathscr{D}$;
$z_{ij}^s \in \mathbb{N}$: number of vehicles actually used between plant $i \in \mathscr{O}_k,\ k \in \mathscr{K}$ and factory $j \in \mathscr{D}$, for scenario $s \in \mathscr{S}$;
$y_j^s \in \mathbb{N}$: number of extra vehicles used from the external company for factory $j \in \mathscr{D}$, for scenario $s \in \mathscr{S}$.

The two-stage integer stochastic programming model with recourse can now be defined as follows:

$$\min q\sum_{k=1}^{K}\sum_{i=1}^{O_k}\sum_{j=1}^{D} t_{ij}x_{ij}+\sum_{s=1}^{S} p^s\left[\sum_{j=1}^{D} qb_j\, y_j^s-\alpha q\sum_{k=1}^{K}\sum_{i=1}^{O_k}\sum_{j=1}^{D} t_{ij}\left(x_{ij}-z_{ij}^s\right)\right] \tag{39}$$

$$\begin{aligned}
s.t. \qquad q\sum_{k=1}^{K}\sum_{i=1}^{O_k} x_{ij} &\leq g_j\ , \quad j \in \mathscr{D}, && (40)\\
0 \leq l_j^0 + q\Big(\sum_{k=1}^{K}\sum_{i=1}^{O_k} z_{ij}^s + y_j^s\Big) - d_j^s &\leq l_{\max}\ , \quad j \in \mathscr{D}, \quad s \in \mathscr{S}, && (41)\\
z_{ij}^s &\leq x_{ij}\ , i \in \mathscr{O}_k, k \in \mathscr{K}, j \in \mathscr{D}, s \in \mathscr{S}, && (42)\\
a_k \leq q\sum_{i=1}^{O_k}\sum_{j=1}^{D} z_{ij}^s &\leq v_k\ , \quad k \in \mathscr{K}, \quad s \in \mathscr{S}, && (43)\\
x_{ij} &\in \mathbb{N}\ , i \in \mathscr{O}_k, k \in \mathscr{K}, j \in \mathscr{D}, && (44)\\
y_j^s &\in \mathbb{N}\ , j \in \mathscr{D}, s \in \mathscr{S}, && (45)\\
z_{ij}^s &\in \mathbb{N}\ , i \in \mathscr{O}_k, k \in \mathscr{K}, j \in \mathscr{D}, s \in \mathscr{S}. && (46)
\end{aligned}$$

The first sum in the objective function (39) denotes the booking costs of the vehicles between the plants and the factories, while the second sum represents the expected recourse costs, which include the cost of the extra vehicles provided by the external company and the discount for the unused booked vehicles. Constraints (40) guarantee that, for each factory $j \in \mathscr{D}$, the number of booked vehicles from the suppliers to the factory does not exceed g_j/q. Constraints (41) ensure that the storage levels of factories $j \in \mathscr{D}$ are between zero and $l_{\max}$. Constraints (42) guarantee that the number of vehicles used by the suppliers are at most equal to the number vehicles booked in advance. Constraints (43) ensure that, for all suppliers $k \in \mathscr{K}$, the number of vehicles used allow the volume of product transported to be between the minimum (i.e., a_k) and maximum (i.e., v_k) established requirements. Finally, (44)–(46) define the decision variables of the problem.

Table 1: SIPLIB instance size

Size	Integer Variables
S	[0,100]
M	[100,1000]
L	> 1000

4.5 SIPLIB

SIPLIB is an available collection of test problems that are used to facilitate computational and algorithmic research in stochastic integer programming. The test problem data is provided in the standard SMPS format unless otherwise mentioned. When available, information on the underlying problem formulations and known solutions are also included, see [1]. The problems that we use are the ones characterized by a two-stage formulation and the presence of integer, or binary, variables in the first stage:

$DCAP$ **test set** is a collection of stochastic integer programs arising in dynamic capacity acquisition and allocation under uncertainty. All problem instances have complete recourse, mixed-integer first-stage variables, pure binary second-stage variables, and discrete distributions.

$SSLP$ **test set** consists of two-stage stochastic mixed-integer programs arising in server location under uncertainty. The problems have pure binary first-stage variables, mixed-binary second-stage variables, and discrete distributions.

$SEMI$ **test set** consists of instances of a two-stage multi-period stochastic integer problem arising in the planning of semiconductor tool purchases. The instances have mixed-integer first-stage variables and continuous second-stage variables.

$mpTSP_s$ **test set.** Instances of the multi-path Traveling Salesman Problem with stochastic travel times ($mpTSP_s$), a variant of the deterministic TSP, where each pair of nodes is connected by several paths and each path entails a stochastic travel time. The problem, arising in the domain of City Logistics, aims to find an expected minimum Hamiltonian tour connecting all nodes [10, 14].

Instances are grouped by size in terms of integer/binary variables. The three groups, namely Small (S), Medium (M), and Large (L), and their range in terms of integer/binary variables are given in Table 1. Table 2 shows the main characteristics of the instances where, for each problem set, Columns 2-3 give the number of instances and their type in terms of number of integer/binary variables, Columns 4-5 display the number of variables in the first and second stage, respectively, Columns 6-7 show the number of integer/binary variables in the first and the second stage, respectively, and Column 8 gives the number of scenarios.

Table 2: SIPLIB instance set description

Problem	Inst #	Type	1-stage v.	2-stage v.	Int. 1-stage v.	Int. 2-stage v.	$\lvert S\rvert$
$DCAP$	12	S	12	[25,35]	6	[25,35]	[200,500]
$SSLP$	10	M	[5,15]	[100,700]	[5,15]	[100,700]	[5,2000]
$SEMI$	3	M	614	9800	612	0	[2,4]
$mpTSP_s$	5	L	[50,100]	[7500, 30000]	[50,100]	[7500, 30000]	100

5 Numerical results

We now present and analyze the results obtained by applying the $GLUSS(p,N)$ measure to the problems described above. We followed the procedure described in 4.1, computing each time VSS, $LUSS$ and $GLUSS(p,N)$, $p = 1,\dots,N$. Detailed solutions of the different instances for the first three test problems may be found at: http://www.matapp.unimib.it/∼maggioni/GLUSS.html. Also, the results obtained on the SIPLIB problems have been integrated to the SIPLIB library [1].

5.1 The single-sink transportation problem

The single-sink transportation problem ($SSTP$) (see Section 4.2) aims to find the number of vehicles to book for each supplier at the beginning of January 2007. We run the model for 10 different instances with a demand randomly generated in the interval $[d^{min}, d^{max}]$, where $d^{min} = 20000$ and $d^{max} = 30000$ are respectively the minimum and maximum demand observed in the historical data.

The deviation (in %) from the optimal solution of the stochastic model, (19)-(27), are reported in Table 3. In Table 4, the optimal solution (optimal number of booked vehicles for each supplier and total optimal cost) for the deterministic and the stochastic models, as well as for the various quality measures are reported for instance 9 (similar observations and arguments apply to the other instances).

The deterministic model always books the exact numbers of vehicles needed ($\bar{x}_i = \bar{z}_i^k$, $i \in \mathscr{I}$, $k \in \mathscr{K}$); it sorts the suppliers according to the transportation costs and books a full production capacity from the cheapest one (AG), followed by the next-cheapest (PA). The deterministic model thus books much fewer vehicles than the stochastic one, resulting in a solution costing only two-thirds of the stochastic counterpart. However, EEV is much higher (€ 481 484.25 instead of the predicted cost of € 287 874) resulting

Table 3: Results for the SSTP

Instance		% from RP	
	VSS	$GLUSS(p, 2)$	
		$p = 1$	$p = 2$
1	12.28	5.41	0
2	16.18	6.95	0
3	12.93	6.03	0
4	13.405	6.05	0
5	18.11	4.37	0
6	14.27	7.32	0.17
7	17.46	9.13	0.03
8	12.97	4.95	0
9	17.27	8.11	0
10	13.91	3.95	0
Mean	14.88	6.23	0.02

in

$$VSS = 481\,484.25 - 410\,573 = 70910.36, \tag{47}$$

which shows that we can save about 17% of the cost by using the stochastic model, compared to the deterministic one.

Table 4: Optimal solutions for the quality measures of SSTP instance 9

Instance	Problem Type	AG	CS	PA	VV	Objective value (€)
9	deterministic	206	0	514	0	287 874=EV
	stochastic	377	0	533	200	410 573=RP
	VSS	206	0	514	0	481 484.25=EEV
	$LUSS$	390	0	633	0	443 881.93= $GESSV(1,2) = ESSV$
	$GLUSS(2, N)$	377	0	533	200	410 573= $GESSV(2,2)$

Why is the deterministic solution bad? Is this due to a shortsighted guess on the randomness (leading to too *few booked vehicles* from the four suppliers), or, can it be explained by the fact the *wrong suppliers* were chosen? Therefore, we compute the $LUSS$ following the skeleton solution from the deterministic model, not allowing vehicles to be booked from either CS or VV. The *Expected Skeleton Solution Value* $ESSV$ is then € 443 881.93, still higher than RP with a consequent *Loss Using the Skeleton Solution* of

$$LUSS = 443\,881.93 - 410\,573 = 33\,308.04, \tag{48}$$

which measures the loss when vehicles are booked exclusively from suppliers AG and PA as suggested by the deterministic model. We can thus conclude that the deterministic solution is bad because it books the wrong number of vehicles from the wrong suppliers.

It should be noted that this approach simply requires solving a MIP of smaller dimension when compared to the original problem.

Since the skeleton solution from the deterministic model sets to zero only the vehicles from CS and VV (formulation (13)), we compute their reduced costs in the continuous relaxation, $r_{CS} = 495$ and $r_{VV} = 277.5$. Let be $\mathscr{R}_1 = \{r_{VV}\}$ and $\mathscr{R}_2 = \{r_{CS}\}$. Fixing at zero only the variables at zero in the expected value solution with the highest reduced cost, we obtain $\mathscr{R}_2 = \{r_{CS}\}$ the *Generalized Expected Skeleton Solution Value* $GESSV(2, N) = RP$ with a consequent *Generalized Loss Using Skeleton Solution:*

$$GLUSS(2, N) = GESSV(2, N) - RP = 0. \tag{49}$$

Not allowing vehicles to be booked both from CS and VV (i.e., fixing at zero the variables at zero in the expected value solution, with associated reduced costs $\mathscr{R}_1 = \{r_{VV}\}$, and $\mathscr{R}_2 = \{r_{CS}\}$), we again compute the *Expected Skeleton Solution Value* $GESSV(1, N) = ESSV = 443\,881.93$ and

$$GLUSS(1, N) = LUSS = 33\,308.04. \tag{50}$$

It should be noticed that, since the number of variables at zero in the deterministic solution is 2, the maximum number of classes is $N = 2$.

From the measures computed, we can conclude that the deterministic solution does not perform well in a stochastic environment. This is explained by the insufficient number of vehicles that are booked in the fist stage (720 instead of 1110) from the suppliers AG and PA. However, by following the skeleton solution with highest reduced costs (i.e., not booking vehicles from CS) we reach the stochastic solution.

5.2 The power generation problem

The power generation problem (PGP) (Section 4.3) selects power units of type 1 or 2 to operate and allocates the power demand among the selected units. We run the model for 10 different instances with demand randomly generated in the interval $[d^{min}, d^{max}]$, where $d^{min} = 33$ and $d^{max} = 687$ are respectively the minimum and maximum demand observed in the historical data. Results are reported in Tables 5 and 6. The former reports the deviations (in %) with respect to RP for VSS, $GLUSS(p, 3)$, $p = 1, \ldots, 3$, and $GLUSS(p, 4)$. The latter illustrates the discussion that follows with the results obtained for the first instance, displaying the first stage solutions of generating units u_i^2, the number of started up generators s_i^2, total output rate x_i^2 ($i \in \mathscr{I}$) and the total cost.

We evaluated the *Expected Value* solution under the mean scenario $\bar{D}$ in the stochastic environment (model (28) - (38)). As illustrated in the case of instance 1 (Table 6), the

Table 5: Results for the PGP (% deviation from RP)

Instance	VSS	$GLUSS(p,3)$			$GLUSS(p,4)$			
		1	2	3	1	2	3	4
1	10.17	10.17	0	0	10.17	0	0	0
2	10.03	10.03	0	0	10.03	0	0	0
3	7.05	0	0	0	0	0	0	0
4	10.66	10.66	0	0	10.66	0	0	0
5	10.07	10.07	0	0	10.07	0	0	0
6	6.78	0	0	0	0	0	0	0
7	6.14	0	0	0	0	0	0	0
8	5.93	0	0	0	0	0	0	0
9	6.84	6.84	0	0	6.84	0	0	0
10	7.44	7.44	0	0	7.44	0	0	0
Mean	8.11	5.44	0	0	5.44	0	0	0

deterministic model closes down as many units as possible to simply cover the considered demand, ending up with only four units of type 1. Because the deterministic solution only keeps 4 units running, instead of the 8 (4 units of type 1 and 4 of type 2) included in the stochastic solution, the associated total cost reduces to 104 285 € compared to 117 927.5 € for the stochastic counterpart. However, the 4 units working in the deterministic solution are not enough to satisfy the high demand scenarios, yielding

$$VSS = 129\,927.5 - 117\,927.5 = 12\,000, \tag{51}$$

causing a loss of 10.17% given the need to restart some units at the second stage. We now investigate why the deterministic solution is bad by means of the following $LUSS$ and $GLUSS$ tests.

Table 6: Optimal solutions from measures VSS, $LUSS$ and $GLUSS$ for PGP instance 1

Problem type	u_1^2	u_2^2	s_1^2	s_2^2	x_1^2	x_2^2	Objective value (€)
Deterministic	4	0	0	0	300	0	104 285=EV
Stochastic	4	4	0	0	180	120	117 927.5=RP
VSS	4	0	0	0	300	0	129 927.5 =EEV
$LUSS$	4	0	0	0	300	0	127 877.5=$ESSV$
$GLUSS(p,N)$, $p = 2,\dots,N$	4	4	0	0	180	120	117 927.5=$GESSV(p,N) = RP$
$GLUSS(p,N)$, $p = 1$	4	0	0	0	300	0	129 927.5=$GESSV(p,N) = ESSV$

To apply the $LUSS$, we follow the skeleton solution from the deterministic model and close units of type 2, not required to satisfy the deterministic demand. The stochastic model reacts by opening units of type 2 at the second stage at higher cost. As a consequence, the associated *Expected Skeleton Solution Value* $ESSV = EEV$ and $LUSS = VSS$, confirming that deterministic solution has a bad structure (required units for the stochastic environment are closed).

To apply the $GLUSS(p, N)$, we compute the reduced costs of the decision variables at zero in the skeleton solution from the deterministic model, which closes units of type 2, $u_2^2 = 0$, yielding $x_2^2 = 0$, and do not start up any generator, $s_i^2 = 0$; thus $r_{u_2^2} = 500$ and $r_{s_1^2} = 14000$, $r_{s_2^2} = 16000$ and $r_{x_2^2} = 50$. We also define $r^{max} = r_{s_2^2} = 16000$ and $r^{min} = r_{x_2^2} = 50$.

Notice that, since the number of variables at zero in the deterministic solution is 4, the maximum number of classes to consider is $N = 4$. We computed two measures, dividing the difference $r^{max} - r^{min}$ into $N = 3$ and $N = 4$ classes of constant width, respectively. With the two values of N we have that $GLUSS(p, N) = 0$, with $p \in \{2, \ldots, N\}$, while the percentage gap between $GLUSS$ and RP becomes 10.17 % when $p = 1$. This shows how the wrong choice from the deterministic solution is in the selection of variable u_2^2.

The results obtained on instances 3, 6, 7 and 8 show a different behavior since $GLUSS(1, N) = LUSS$ is able to perfectly reproduce the value of RP, where the VSS is around 6.5% (see Table 5). This means that, in these instances, the EV problem is able to identify the appropriate structure in terms of zero and non-zero variables, but fails in providing the correct first-stage non zero values.

In conclusion, the deterministic solution is bad because it tends to follow in every period the market profile, thus closing units that could be needed in the following time periods. However, by following the skeleton of the deterministic solution with highest reduced costs (i.e., do not starting up any generator, $s_i^2 = 0$) we reach the stochastic solution.

5.3 The supply transportation problem

VSS, $LUSS$ and $GLUSS(p, N)$, $p = 1, \ldots, N$ were performed for the supply transportation problem (STP), see Section 4.4, which identifies the number of vehicles to book for each plant of each supplier, for the replenishment of gypsum at minimum total cost. Data represents the first week of March 2014. We run the model for 10 different instances with demand randomly generated in the interval $\left[d_j^{min}, d_j^{max}\right]$, $j \in \mathscr{J}$, where d_j^{min} and d_j^{max} are the minimum and maximum demand observed in the historical data, respectively (see Columns 3 and 4 of Table 7).

The cost values associated to the solutions of the deterministic model, the expected value problem (4), and the stochastic formulation (39)-(46) are reported in Table 8 for the 10 instances The deterministic model will always book the exact number of vehicles

Table 7: Destinations with minimum & maximum observed demand and deterministic & stochastic solution values averaged over 10 instances

	Demand		Solution	
Destination	Minimum	Maximum	EEV	Stochastic
$j \in \mathscr{D}$	d_j^{min}	d_j^{max}	$\bar{x}_{ij}$	x_{ij}
1	27.45	298.43	6	9
2	202.01	1479.89	29	26
3	171.78	680.16	14	21
4	0	216.96	4	7
5	0	101.26	2	3
6	0	196.93	4	6
7	0	216.20	4	7
8	0	200.43	4	6
9	0	545.19	10	15
10	0	234.37	4	7
11	0	318.89	6	9
12	0	430.36	7	11
13	0	199.42	4	6
14	0	223.50	4	7
15	0	723.46	12	20

needed for the next period and so, $\bar{x}_{ij} = \bar{z}^s_{ij}$, $i \in \mathscr{O}_k$, $k \in \mathscr{K}$, $j \in \mathscr{J}$, $s \in \mathscr{S}$; it sorts the suppliers and their plants according to the transportation costs and books a full production capacity from the cheapest one, followed by the next-cheapest, and so on. As long as there is sufficient transportation capacity, the model will never purchase extra gypsum from external sources, i.e. $y_j = 0$, $\forall j \in \mathscr{D}$. The total cost then reduces to the booking cost at the first stage.

The last two columns of Table 7 show the total number of booked vehicles at each cement factory averaged over the 10 instances, $\sum_{k\in\mathscr{K}}\sum_{i\in\mathscr{O}_k} x_{ij}$, $\forall j \in \mathscr{D}$, for the expected value solution and the optimal solution of the stochastic problem, respectively. The deterministic model books much fewer vehicles than the stochastic one, resulting in a solution costing only 83% of the stochastic counterpart (Table 8). The EEV is infeasible, however, resulting in $VSS = \infty$, which shows that the expected value solution is not appropriate in a stochastic setting.

Table 8: Optimal solution values for the STP

Instance	Problem Type	1^{st} stage cost (€)	Objective value (€)
1	Deterministic	79 709.30	79 709.30=EV
	Stochastic	123 673.00	95 738.53=RP
2	Deterministic	76 768.80	76 768.79=EV
	Stochastic	119 626.00	93 468.32=RP
3	Deterministic	77 386.50	77 386.50=EV
	Stochastic	121 147.00	93 297.18=RP
4	Deterministic	74 332.00	74 332.00=EV
	Stochastic	122 054.00	90 734.68=RP
5	Deterministic	76 101.30	76 101.30=EV
	Stochastic	119 638.00	93 014.14=RP
6	Deterministic	75 066.10	75 066.10=EV
	Stochastic	118 657.00	91 661.99=RP
7	Deterministic	76 992.00	76 992.00=EV
	Stochastic	119 004.00	94 066.06=RP
8	Deterministic	78 859.30	78 859.30=EV
	Stochastic	123 980.00	94 306.65=RP
9	Deterministic	75 111.80	75 111.80=EV
	Stochastic	119 033.00	91 966.36=RP
10	Deterministic	80 344.30	80 344.30=EV
	Stochastic	123 781.00	96 296.00=RP
Mean	Deterministic	77 067.14	77 067.14=EV
	Stochastic	121 059.35	93 454.99=RP

Why is the deterministic solution bad? Is it because of an overly optimistic guess on the randomness, leading to too *few booked vehicles* from the plants $i \in \mathscr{O}_k$ of suppliers $\mathscr{K}$, or is it because of wrong choices being made regarding the *suppliers and plants*? To answer these questions, we again perform the $LUSS$ and $GLUSS$ measures.

To compute the $LUSS$, we follow the skeleton solution from the deterministic model, not allowing vehicles to be booked from the plants $i \in \mathscr{O}_k$ (for all suppliers $k \in \mathscr{K}$), such that $x_{ij} = \bar{x}_{ij}(\bar{\xi}) = 0$ in the expected value solution. The *Expected Skeleton Solution Value* $ESSV$ is still infeasible with an associated *Loss Using the Skeleton Solution* $LUSS = \infty$. Therefore, the chosen suppliers and associated plants, derived from the solution to the deterministic model, are unsuited for the stochastic case. We can thus conclude that the deterministic solution is inappropriate because the wrong number of vehicles are booked from the the wrong suppliers and plants.

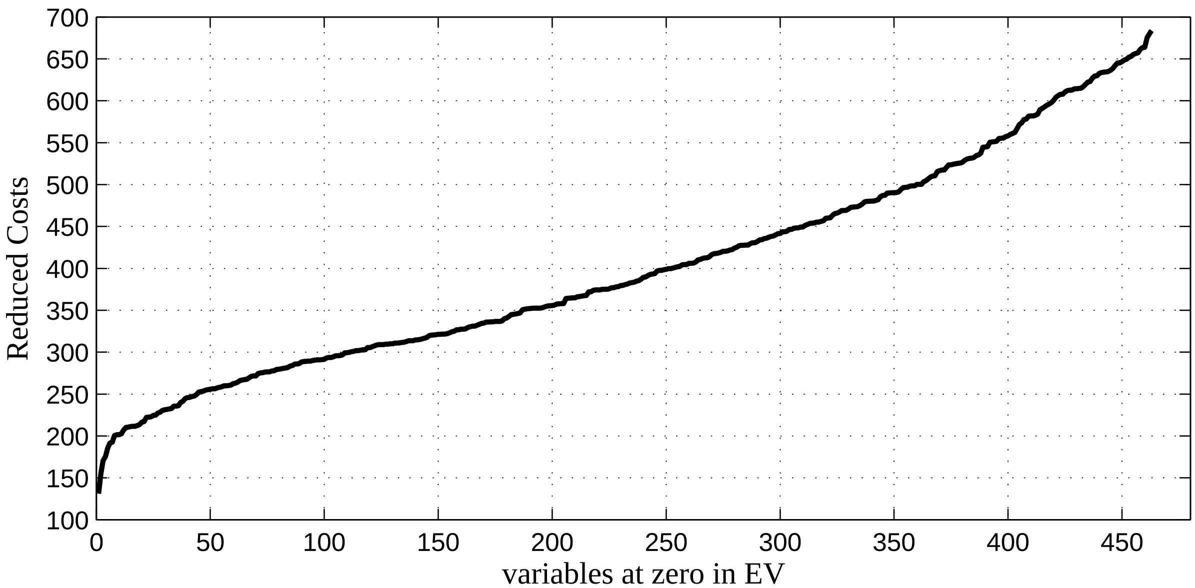

Figure 1: Reduced costs of the variables at zero in the EV solution of STP instance 1

In applying $GLUSS(p, N)$, we analyze the reduced costs of the variables at zero in the deterministic solution, illustrated in Figure 1 for the first instance. The range of reduced costs, which is from $r^{min} = 131$ to $r^{max} = 683$, is sufficiently broad to allow testing the sensitivity of the results with a large number of classes N. We therefore divide the difference $r^{max} - r^{min} = 552$ into $N = 3, 10, 50, 100$, classes $\mathscr{R}_1, \ldots, \mathscr{R}_N$ of constant width, respectively. Results are reported in Tables 10, 11 and 12, respectively.

Contrary to the VSS and $LUSS$, the $GLUSS(p, N)$ is able to find optimal results when a limited subset of variables are fixed. In the case of $GLUSS(p, 3)$, for $p = 1, 2, 3$, the appropriate variables from the deterministic solution are identified as the ones included in the last two classes (i.e., $\left[r^{min} + \frac{r^{max} - r^{min}}{3}, r_{max}\right]$), considering that $GLUSS(2, 3) = GLUSS(3, 3) = 0$. On the other hand, fixing at zero all the variables

Table 9: CPU time (seconds) for the computation of EV, RP. $GLUSS(p,3)$, $p = 1,\dots,3$ and $GLUSS(p,10)$, $p = 1,\dots,10$, for the STP

Measure	CPU time (ss)
EV	0.0780005
RP	3.07322
$GLUSS(3,3)$	1.49761
$GLUSS(2,3)$	0.748805
$GLUSS(1,3)$	0.140401
Total $GLUSS(p,3)$	2.246415
$GLUSS(10,10)$	1.794011
$GLUSS(9,10)$	1.809612
$GLUSS(8,10)$	1.48201
$GLUSS(7,10)$	1.357209
$GLUSS(6,10)$	1.294808
$GLUSS(5,10)$	0.920406
$GLUSS(4,10)$	0.608404
$GLUSS(3,10)$	0.390002
$GLUSS(2,10)$	0.140401
$GLUSS(1,10)$	0.156001
Total $GLUSS(p,10)$	9.952864

at zero in the deterministic solution yields $GLUSS(1,3) = \infty$. It should also be noticed that $GLUSS(p,3)$, $p = 2,3$ is able to replicate the optimal values of the stochastic problem while reducing the computational effort by 50% when $p = 3$ and by 75% when $p = 2$; see Table 9.

One obtains more refined information on the wrong variables from the deterministic solution by increasing the number of classes to $N = 10$, and identifying the good variables to fix as the ones belonging to the classes in the interval $\left[r^{min} + \frac{3(r_{max}-r_{min})}{10}, r^{max}\right]$. These results are displayed in Table 10. Furthermore, by also fixing the variables belonging to the interval $\left[\frac{2(r_{max}-r_{min})}{10}, \frac{3(r_{max}-r_{min})}{10}\right]$, one obtains a nearly optimal solution. Adding the class $p = 2$, results in $GLUSS(2,10) = \infty$ and consequently $GLUSS(1,10) = \infty$. As previously observed, $GLUSS(4,10)$ is able to replicate the optimal values of the stochastic problem while reducing the computational effort by a significant margin (i.e., 80%), see Table 9.

Increasing the number of classes to $N = 50$, see Table 11, further refines the information deduced from the deterministic-model solution regarding the good variables to fix, as $GLUSS(p,50) = 0$, with $p = 15,\dots,50$. In terms of the computational effort, the observed gains increase to 84% with $p = 15$. By setting $N = 100$, two extra variables from

Table 10: Results of $GLUSS(p, 3)$ and $GLUSS(p, 10)$ for the STP as % from RP

Instance	VSS	$GLUSS(p, 3)$			$GLUSS(p, 10)$									
		1	2	3	1	2	3	4	5	6	7	8	9	10
1	∞	∞	0	0	∞	∞	0.006	0	0	0	0	0	0	0
2	∞	∞	0	0	∞	∞	0.006	0	0	0	0	0	0	0
3	∞	∞	0	0	∞	∞	0.008	0	0	0	0	0	0	0
4	∞	∞	0	0	∞	∞	0.084	0	0	0	0	0	0	0
5	∞	∞	0	0	∞	∞	0.001	0	0	0	0	0	0	0
6	∞	∞	0	0	∞	∞	0	0	0	0	0	0	0	0
7	∞	∞	0	0	∞	∞	0.01	0	0	0	0	0	0	0
8	∞	∞	0	0	∞	∞	0.006	0	0	0	0	0	0	0
9	∞	∞	0	0	∞	∞	0.002	0	0	0	0	0	0	0
10	∞	∞	0	0	∞	∞	0.002	0	0	0	0	0	0	0
Mean	∞	∞	0	0	∞	∞	0.0129	0	0	0	0	0	0	0

the deterministic solution can be detected, $GLUSS(p, 100) = 0$, with $p = 28, \ldots, 100$, see Table 12). In this case, the gain in computational effort is 81% with $p = 28$.

5.4 SIPLIB

In this subsection we discuss the results of $GLUSS(p, N)$ with respect to $LUSS$ and VSS for the set of SIPLIB library instances. Results are obtained using the best known solutions of the RP, i.e., the proven optima for all the instances. For the $GLUSS(p, N)$, we used the number of classes set to $N = 3$. Notice that the definition of $LUSS$ is equivalent to $GLUSS(1, 3)$. Table 13 summarizes the results obtained for the $DCAP$ instances, where Column 1 gives the instance name, Columns 2-5 show the gaps (in %) relative to the optimal values of the stochastic formulation (the RP) for the VSS and the $GLUSS(p, N)$, $p = 1, 2, 3$, while Columns 4-9 display the corresponding computational times in CPU seconds.

The results illustrate how the first-stage solution obtained by solving the mean value problem fails to provide a good solution in the stochastic case, with a VSS mean error fo 31%. $LUSS$ (and $GLUSS(1, 3)$) reduces the error to 23%, which remains unacceptable in many practical situations. This behavior is observed until we fix p to a value that reduces the number of fixed variables in our modified stochastic formulation. We obtain a deviation from the proven optima of 4% when $p = 3$. In this case, we also obtain a large reduction of the computational effort (about 7 times on average). These reductions reach one order of magnitude on the largest instances.

Table 11: Results of $GLUSS(p, 50)$ for the STP as % from RP

Instance	VSS	$GLUSS(p, 50)$			
1	∞	$p \leq 10 : \infty$	$11 \leq p \leq 14 : 0.0066$		$p \geq 15 : 0$
2	∞	$p \leq 8 : \infty$	$9 \leq p \leq 10 : 0.1644$	$11 \leq p \leq 14 : 0.0063$	$p \geq 15 : 0$
3	∞	$p \leq 10 : \infty$	$11 \leq p \leq 14 : 0.00806$		$p \geq 15 : 0$
4	∞	$p \leq 10 : \infty$	$11 \leq p \leq 14 : 0.084$		$p \geq 15 : 0$
5	∞	$p \leq 10 : \infty$	$11 \leq p \leq 14 : 0.0012$		$p \geq 15 : 0$
6	∞	$p \leq 10 : \infty$			$p \geq 11 : 0$
7	∞	$p \leq 10 : \infty$	$11 \leq p \leq 14 : 0.0103$		$p \geq 15 : 0$
8	∞	$p \leq 10 : \infty$	$11 \leq p \leq 14 : 0.0069$		$p \geq 15 : 0$
9	∞	$p \leq 10 : \infty$	$11 \leq p \leq 14 : 0.0029$		$p \geq 15 : 0$
10	∞	$p \leq 10 : \infty$	$11 \leq p \leq 14 : 0.0029$		$p \geq 15 : 0$

Table 12: Results of $GLUSS(p, 100)$ for the STP as % from RP

Instance	VSS	$GLUSS(p, 100)$			
1	∞	$p \leq 19 : \infty$	$p = 20 : 0.0355$	$21 \leq p \leq 27 : 0.0066$	$p \geq 28 : 0$
2	∞	$p \leq 15 : \infty$	$p = 16 : 0.1733$	$17 \leq p \leq 19 : 0.1644$	$p = 20 : 0.0391$
				$21 \leq p \leq 27 : 0.0063$	$p \geq 28 : 0$
3	∞	$p \leq 19 : \infty$	$p = 20 : 0.021$	$21 \leq p \leq 27 : 0.00806$	$p \geq 28 : 0$
4	∞	$p \leq 19 : \infty$	$p = 20 : 0.1881$	$21 \leq p \leq 27 : 0.0849$	$p \geq 28 : 0$
5	∞	$p \leq 19 : \infty$	$p = 20 : 0.017$	$21 \leq p \leq 27 : 0.0012$	$p \geq 28 : 0$
6	∞	$p \leq 19 : \infty$	$p = 20 : 0.0278$		$p \geq 21 : 0$
7	∞	$p \leq 19 : \infty$	$p = 20 : 0.0424$	$21 \leq p \leq 27 : 0.0103$	$p \geq 28 : 0$
8	∞	$p \leq 19 : \infty$	$p = 20 : 0.0492$	$21 \leq p \leq 27 : 0.0069$	$p \geq 28 : 0$
9	∞	$p \leq 19 : \infty$	$p = 20 : 0.0204$	$21 \leq p \leq 27 : 0.0029$	$p \geq 28 : 0$
10	∞	$p \leq 19 : \infty$	$p = 20 : 0.0273$	$21 \leq p \leq 27 : 0.0029$	$p \geq 28 : 0$

Table 13: Results of SIPLIB $DCAP$ instances

Instance	% from RP				CPU time (ss)			
	VSS	$GLUSS(p,3)$			RP	$GLUSS(p,3)$		
		1	2	3		1	2	3
dcap233.200	12.6	3.49	3.49	0	18.15	1.89	5.75	5.56
dcap233.500	44.2	10.93	10.93	0	15.89	13.23	28.73	29.85
dcap243.200	32.9	30.65	30.65	1.28	51.68	1.94	1.9	3.03
dcap243.300	1.6	1.38	1.38	0.01	23.48	6	6.04	46.38
dcap243.500	17.4	16.59	16.59	0.86	83.84	16.07	16.88	24.68
dcap332.200	57.3	50.95	50.95	7.16	133.34	6.28	3.7	5.4
dcap332.300	28.9	28.09	28.09	3.84	141.32	4.34	5.75	14.39
dcap332.500	24.7	23.62	23.62	8.82	199.67	8.6	8.55	48.2
dcap342.200	35.4	33.42	33.42	9.41	131.94	2.11	2.26	8.15
dcap342.300	48.6	28.26	28.26	6.96	493.15	5.51	6.92	15.68
dcap342.500	61.5	27.84	27.84	6.5	349.98	26.98	26.29	25.1
Mean	33.17	23.20	23.20	4.08	149.31	8.45	10.25	20.58

The $SSLP$ results are reported in Table 14 (same column definitions as the previous table). The results of $GLUSS(p, 3)$, computed with $p = 1$, are reported to show that $LUSS = GLUSS(1, 3)$ reproduces exactly the value fo RP, while the VSS is very high (50% on average). This means that, the EV problem preserves the structure in terms of zero and non zero variables, but fails in providing the correct first-stage non zero values. Thus, in this case, the gain in using the $GLUSS(1, 3)$ resides in the reduction of the computational time by a factor of 2. This is far from marginal considering that, when the size of the instances increases, solving the full stochastic formulation reaches 10 000 seconds, while $GLUSS(p, N)$ and $LUSS$ find the optimal solution within a computational time that reduces up to 5 times.

A completely different behavior regarding $LUSS$ is observed in the results obtained for the $SEMI$ instances, displayed in Table 15 (same organization as the previous table). In this case, the performance obtained with the $LUSS$ is equal to the VSS, which provide a relatively good gap (close to 5% on average). The $GLUSS(3, 3)$ is able to find optimal results when a limited subset of variables are fixed. When the number variables set to 0 increases ($p = 2$), the gap remains small (0.15%). In this case, the gain in terms of computational effort is somewhat limited when $p = 3$, while becoming significant, almost 2 orders of magnitude, when $p = 2$.

Up to now, we have examined the effect of the $GLUSS(p, N)$ on small and medium-sized instances. However, what about larger-sized instances? Is $GLUSS(p, N)$ able to replicate optimal or near optimal values while reducing the computational effort? The

Table 14: Results for SIPLIB $SSLP$ instances

Instance	% from RP		CPU time (ss)	
	VSS	$GLUSS(p,3)$	RP	$GLUSS(p,3)$
		1		1
sslp5.25.50	43.36	0	0.8	0.6
sslp5.25.100	42.83	0	1.5	1.5
sslp10.50.50	30.43	0	920.8	498.6
sslp10.50.100	31.64	0	3608.9	2711.8
sslp10.50.500	32.24	0	3620	2702.2
sslp10.50.1000	32.16	0	10936	2705.2
sslp10.50.2000	32.93	0	40683	4618.4
sslp15.45.5	78.68	0	3.8	9.8
sslp15.45.10	74.24	0	8.5	6.3
sslp15.45.15	73.48	0	262.2	83.1
Mean	47.199	0	6004.6	1333.7

Table 15: Results for SIPLIB $SEMI$ instances

Instance	% from RP				CPU (ss)			
	VSS	$GLUSS(p,3)$			RP	$GLUSS(p,3)$		
		1	2	3		1	2	3
semi2	3.59	3.59	0.17	0	2164	24.2	81.1	1701.1
semi3	4.38	4.38	0.23	0	8914	22.2	149	5568
semi4	5.68	5.68	0.06	0	27519	40.3	757.7	15923
Mean	4.55	4.55	0.15	0	12865.67	28.9	329.26	7730.62

Table 16: Results for SIPLIB $mpTSP_s$ instances

Instance	% from RP				CPU (ss)			
	VSS	$GLUSS(p,3)$			RP	$GLUSS(p,3)$		
		1	2	3		1	2	3
D0.50	4.22	4.22	2.54	0	473.3	2.8	58.25	265.05
D1.50	4.88	4.88	2.78	0	137.4	0.9	86.1	127
D2.50	2.05	2.05	0.64	0	655.5	0.7	79.8	411.1
D3.50	3.75	3.75	1.71	0	2069.1	0.8	129.5	567.8
D1.100	4.22	4.22	2.54	0	12376	2.6	58.3	256.1
Mean	3.82	3.82	2.04	0	3142.26	1.56	82.39	325.41

answer is yes to both questions, as can be seen in Table 16 for the SIPLIB $mpTSP_s$ instances. Once again, $GLUSS(p, N)$, with $p = 3$, is able to replicate the optimal values while reducing the computational effort by an order of magnitude.

To conclude, it is clear how the $GLUSS(p, N)$ can be effectively used to find high quality solutions to stochastic problems by starting from the EV solutions. Furthermore, when compared to the effort needed to find the optimal solution to the full stochastic formulation, $GLUSS(p, N)$ considerably reduces the computational times.

6 Highlights and general trends

The detailed results presented in Section 5 show how the $GLUSS$ can be used to derive a structure of the stochastic solution starting from data extracted from the continuous relaxation of the expected value solution. In this section, we summarize the lessons learned from our experiments applying $GLUSS$ to different problems, considering the issues of computational effort, feasibility and optimality. Finally, trends and perspectives are also highlighted.

One of the main issues that emerges when using the $GLUSS$ is how to choose the number N of classes dividing the reduced costs of out-of-basis variables. While on one hand, in order to reduce the problem size, it would be preferable to fix the largest possible number of variables, on the other hand, fixing too large a number may result in errors in terms of feasibility and optimality. The general trend emerging from the empirical observations is that fixing to 0 about 33% of the non-basic variables with the highest reduced costs is a good compromise. Indeed, applying this policy, we reached the optimal stochastic solutions without feasibility issues and reducing the computational time up to two orders of magnitude for the largest instance (Table 16).

From the point of view of problem optimality, it seems that, as already noted for deterministic combinatorial models, the reduced costs obtained from a continuous relaxation of an integral problem hint to the variables to make inactive in order to guide the search for optimal solutions to stochastic programs [13]. Moreover, the results show that, even when the VSS is high and, then, the objective function of the expected value deterministic model is far from the one of the stochastic problem, the expected value deterministic solution is providing correct information about the optimal stochastic solution. On the other hand, problems with just a few variables with positive reduced costs in their deterministic solution structure (e.g., the $DCAP$ instances in SIPLIB, Table 13), highlight the need to extend the $GLUSS$ approach by defining a measure for ranking also the basic variables associated to the continuous relaxation of the expected value deterministic problem.

Regarding the distribution of the reduced costs in the expected value deterministic solution, one idea is to compute them, and plot them or pass them through a statistical package, to see if one can observe a trend referable to a known probability distribution. Unfortunately, the answer appears to be "no", even though the distribution seems to have a certain regularity for low values of the number of classes N. For larger numbers of sets into which to categorize the variables, this regularity is less evident. We illustrate this phenomenon with the results obtained for instance 1 of the supply transportation problem. Figure 2 displays the histograms of the reduced cost distribution. The graphs show how, up to N=10, the reduced cost distribution has almost a Gumbel shape. Increasing the number of classes to 50 and then to 100, its behavior becomes very irregular and a reduced costs probability distribution is difficult to be identified. Similar observations were made for other instances and problem classes (e.g., the $mpTSP_s$).

We observed feasibility issues when fixing subsets of variables from the deterministic solution of the supply transportation problem, following the computation of $GLUSS(1,3)$, $GLUSS(p,10)$, with $p = 1,2$, $GLUSS(p,50)$ with $p = 1,\dots,10$ and $GLUSS(p,100)$ with $p = 1,\dots,19$. We therefore performed a sensitivity analysis on values of a number of parameters, the stochastic demand d_j^s and the minimum capacity requirement capacity a_k of supplier $k \in \mathscr{K}$, aiming to obtain the largest set of variables from the deterministic solution that cause infeasibility in the stochastic one.

The results of this analysis show that the infeasibility comes out in classes $GLUSS(p,100)$, with $p = 1,\dots,19$, for $a_k < 16.13\%\ v_k$, $k = 4,6,10,11,15,16$ and $a_1 < 2000$ (see Table 23) since too large a number of variables have been fixed not allowing to satisfy the constraint (43) on the minimum capacity requirement. For $a_k = 16.13\%\ v_k$, $k = 4,6,10,11,15,16$ and $a_1 = 2000$, the stochastic problem itself becomes infeasible and, consequently, also all $GLUSS(p,100)$, $p = 1,\dots,100$. The reason of the infeasibility is the following: if the value of the minimum capacity requirement a_k is increased in constraint (43), the model decides to transport at least for the required quantity. Consequently, for a scenario with low demand, constraint (41) is no longer satisfied, since the maximum storage capacity

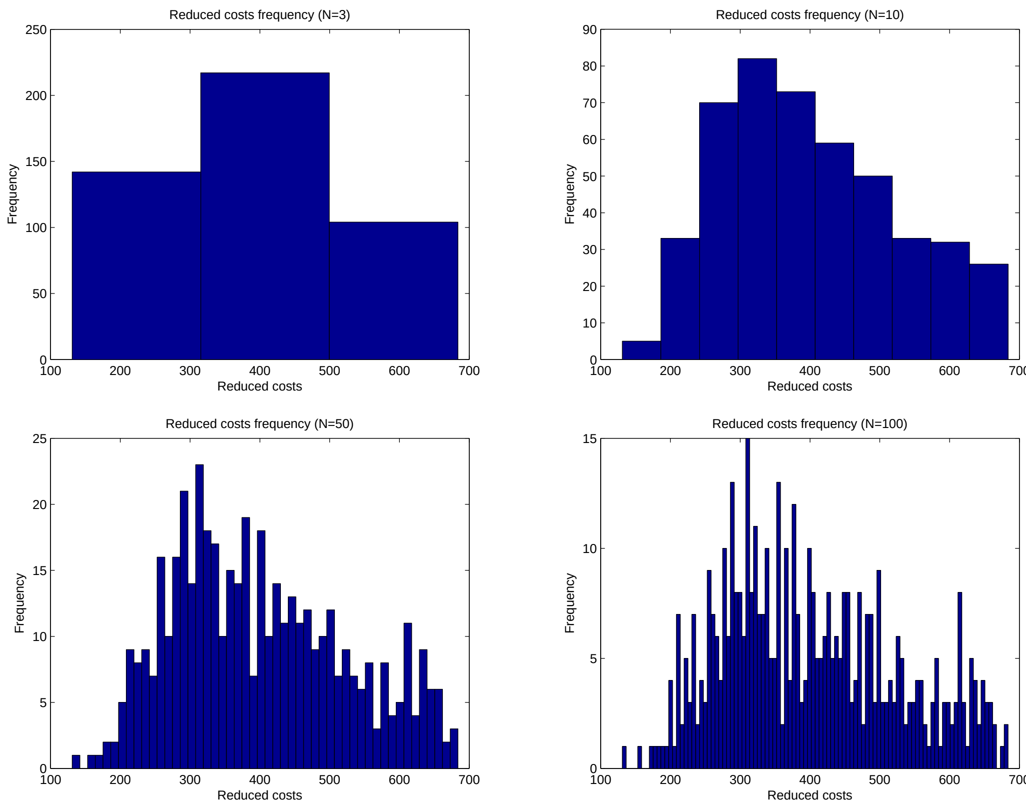

Figure 2: Absolute frequency of reduced costs of out of basis variables at zero in the EV solution for the STP instance 1, for $N = 3, 10, 50, 100$

at the customer is overtaken, generating the infeasibility. On the other hand, high demand scenarios will not bring infeasibility to the model since constraint (41) is satisfied by acquiring extra product from external sources at a higher price.

Histograms of the reduced-cost distributions are plotted in Figure 3, for $N = 3, 10, 50, 100$. First, one can notice how, when considering higher values of N, the classes containing the largest number of variables become the ones in the middle and in the left tail, i.e., the classes characterized by the lowest reduced costs. Moreover, the results show that there is empirical evidence that the $GLUSS$ is stable also from the point of view of feasibility, if one does not try to fix p close to 1, i.e., one fixes to 0 the largest part of out-of-basis variables.

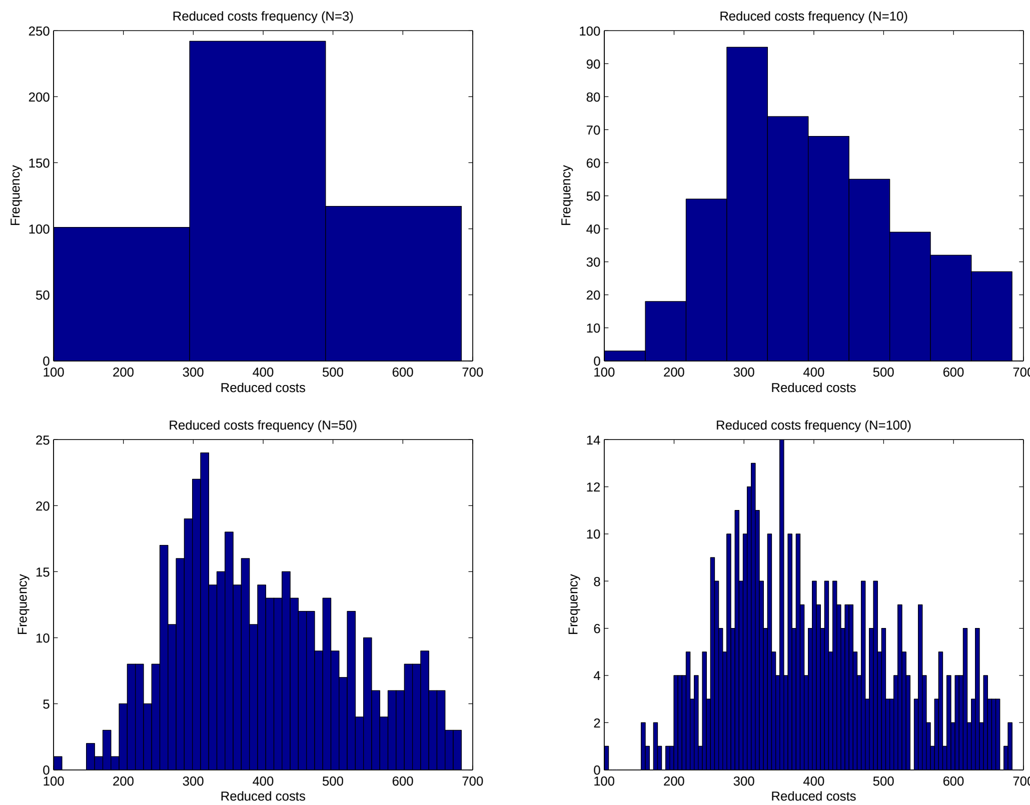

Figure 3: Absolute frequency of reduced costs of out of basis variables at zero in the EV solution for the STP instance 1, for $N = 3, 10, 50, 100$, with $a_1 = 2000$ and $a_k = 16.13\%\ v_k$, $k = 4, 6, 10, 11, 15, 16$.

Another interesting empirical evidence comes from the computational results. Recall that in the literature on deterministic combinatorial optimization, there exists similar approaches to the $GLUSS$ [2, 13]. Indeed, in order to reduce the computational time, one may fix to zero the largest part of the out-of-basis variables in the continuous relaxation of the problem. Then, to identify the appropriate core set of out-of-basis variables to be included in the restricted problem, the search is performed starting from the ones with the smallest reduced cost. This is exactly the opposite of the $GLUSS$. This may be explained by the fact that, in order to obtain a substantial reduction of the computational effort in the deterministic combinatorial case, one has to remove a lot of out-of-basis variables. In stochastic programming, on the contrary, just removing a small subset of out-of-basis variables pays a lot in terms of computational effort (see the results in Table 16). This is probably due to the particular structure of two-stage models with recursion and the effect of fixing first-stage variables in the second-stage scenario sub-problems.

From all the case studies considered, we derive a hint on how to proceed when we want to apply the $GLUSS$ to a new problem. Thus, an empirical method is the following:

- Divide the reduced costs in $N = 3$ intervals and fix in the stochastic first stage solution only the variables belonging to the third class, i.e., with highest reduced costs;
- If there are feasibility issues, consider the removed interval and split it again into three intervals;
- If one desires a greater precision (and the number of zero variables appearing in the continuous relaxation of the deterministic approximation is sufficiently large), split the out-of basis variables into 10 bids and try the values $p = 2, 3$, and 4.

7 Conclusions and future directions

In this paper, we analyzed the *quality* of the expected value solution with respect to the stochastic one, and introduced the *Generalized Loss Using Skeleton Solution* $GLUSS$, a measure that goes beyond the standard *Value of the Stochastic Solution* VSS and the *Loss Using Skeleton Solution*, $LUSS$ [12].

$GLUSS$ takes into account the information on reduced costs of out-of-basis variables in the deterministic expected value optimal solution. Reduced costs of first stage out-of-basis variables from EV are sorted, grouped into homogeneous classes, and then fixed in the associated stochastic problem from highest to lowest, significantly reducing the computational effort.

We performed a wide range of experiments on instances drawn both from the Stochastic Programming literature and real cases. The results show that the $GLUSS$ can help identify the good and bad variables to keep from the deterministic to the stochastic solution. In all the cases considered, fixing the variables with *high* reduced costs from the EV when solving the RP allowed us to reach exactly the stochastic solution. We could thus identify the main causes of goodness of the expected value solution in the variables with highest reduced cost in the deterministic solution, measured by a zero *Generalized*

Loss Using the Skeleton Solution $0 = GLUSS(p, N) \leq LUSS \leq VSS$. These results also show how a smart usage of the information coming from the linear programming theory can be effectively incorporated in a Stochastic Programming resolution approach in order to build accurate solutions.

The proposed $GLUSS$ measure and procedure can then be effectively used both for problems actually solvable but that must be run very often, and for intractable real-world problems, to reduce the computational time of solving the stochastic problem, without loosing in terms of solution quality.

The introduction of the $GLUSS$ opens a number of interesting future research directions, including how to extend and incorporate this idea into various algorithmic frameworks, such as progressive hedging, diving procedures [13], etc. The computational analysis performed in this paper points to a second avenue. We have seen when classifying the non-basis variables in term of their reduced cost that whose with the *highest* values can be hard-fixed to zero without affecting the final quality of the stochastic solution. On the other side, the variables with the *lowest* reduced costs should be present in the stochastic model. But what about the variables which are in between these two extreme classes? Can we define a way to identify those hedging variables and to incorporate this? What is the appropriate number N of classes needed and their usage (which ones to be fixed and which ones no)? A related, but different research avenue concerns the case, studied within the branch-and-bound literature for deterministic formulations, of identifying a measure of the willingness to fix a non-zero variable and how to fix it. We expect to report on some of these issues in the near future.

Acknowledgments

Partial funding for this project was provided by the Italian University and Research Ministry under the UrbeLOG project-Smart Cities and Communities. The first author acknowledges financial support from the grants "Fondi di Ricerca di Ateneo di Bergamo 2014", managed by Francesca Maggioni. Partial funding for this project has also been provided by the Natural Sciences and Engineering Council of Canada (NSERC) through its Discovery Grants program. We also gratefully acknowledge the support of the Fonds de recherche du Québec through their infrastructure grants.

While working on this project, the second author was Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal. Mircea Simionica and Daniele Magnaldi are also acknowledged for their collaboration.

References

[1] S. Ahmed, R. Garcia, N. Kong, L. Ntaimo, G. Parija, F. Qiu, and Sen. S. Siplib: A stochastic integer programming test problem library, 2015. http://www.isye.gatech.edu/ sahmed/siplib.

[2] E. Angelelli, R. Mansini, and M.G. Speranza. Kernel search: A general heuristic for the multi-dimensional knapsack problem. *Computers & Operations Research*, 37(11):2017 – 2026, 2010. Metaheuristics for Logistics and Vehicle Routing.

[3] J. R. Birge and F. Louveaux. *Introduction to stochastic programming*. Springer, Princeton, NJ, USA, 2011.

[4] J.R. Birge. The value of the stochastic solution in stochastic linear programs with fixed recourse. *Mathematical Programming*, 24(1):314–325, 1982.

[5] L.L. Garver. Power generation scheduling by integer programming-development of theory. *Power Apparatus and Systems, Part III. Transactions of the American Institute of Electrical Engineers*, 81(3):730–734, April 1962.

[6] P. Kall and S.W. Wallace. *Stochastic Programming*. John Wiley & Sons, 1994.

[7] A.G. Lium, T.G. Crainic, and S.W. Wallace. A study of demand stochasticity in service network design. *Transportation Science*, 43(2):144–157, 2009.

[8] F. Maggioni, E. Allevi, and M. Bertocchi. Bounds in multistage linear stochastic programming. *Journal of Optimization Theory and Applications*, 163(1):200–229, 2014.

[9] F. Maggioni, M. Kaut, and L. Bertazzi. Stochastic optimization models for a single-sink transportation problem. *Computational Management Science*, 6:251–267, 2009.

[10] F. Maggioni, G. Perboli, and R. Tadei. The multi-path traveling salesman problem with stochastic travel costs: Building realistic instances for city logistics applications. *Transportation Research Procedia*, 3:528–536, 2014.

[11] F. Maggioni, F. Potra, and M. Bertocchi. Stochastic versus robust optimization for a supply transportation problem. (submitted), 2014.

[12] F. Maggioni and S.W. Wallace. Analyzing the quality of the expected value solution in stochastic programming. *Annals of Operations Research*, 200:37–54, 2012.

[13] G. Perboli, R. Tadei, and D. Vigo. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science*, 45:364–380, 2011.

[14] R. Tadei, G. Perboli, and F. Perfetti. The multi-path traveling salesman problem with stochastic travel cost. *EURO Journal on Transportation and Logistics*, forthcoming:doi:10.1007/s13676–014–0056–2, 2014.

[15] B.K. Thapalia, T.G. Crainic, M. Kaut, and S.W. Wallace. Single-commodity stochastic network design with multiple sources and sinks. *INFOR*, 49(3):195–214, 2011.

[16] B.K. Thapalia, T.G. Crainic, M. Kaut, and S.W. Wallace. Single-commodity stochastic network design with random edge capacities. *European Journal of Operational Research*, 9:139–160, 2012.

[17] B.K. Thapalia, S.W. Wallace, M. Kaut, and T.G. Crainic. Single source single-commodity stochastic network design. *Computational Management Science*, 9(1):139–160, 2012.

[18] H.P. Williams. *Model buiding in mathematical programming.* Wiley & Sons, 2013.

1 Annex

1.1 A single-sink transportation problem

Problem data are presented in Tables 17–18. Table 17 presents the production and transportation costs for each supplier, together with its distance from the customer in Catania, while Table 18 reports the monthly production capacity of each supplier in the considered period (zero entries represent production site closures due to equipment failure or maintenance).

Table 17: Production costs c_i and transportation costs t_i from Catania.

Supplier	c_i (€/t)	t_i (€/t)
Porto Empedocle (AG)	18.79	11.40
Castrovillari (CS)	9.55	33.00
Isola d. Femmine (PA)	11.00	14.10
Vibo Valentia (VV)	11.54	18.50

We used in our computational experiment, the vehicle capacity $q = 30$ tonnes (t), the storage capacity $l_{max} = 35$ kilotonnes (kt) and the *daily* unloading capacity of 1800 t, giving us the *monthly* unloading capacity $g = 21 \times 1800\,\text{t} = 37.8\,\text{kt}$, or 1260 full vehicles. The cost of clinker from an external source was set to $b =$ €45/t and the cancellation fee to $\alpha = 0.5$. For the initial inventory level l_0 at the customer, we have taken the value at the beginning of January 2007, that is $l_0 = 2000\,\text{t}$.

1.2 Power generation scheduling

Table 19 reports energy demands at the nodes $n \in \mathscr{N}$ of the scenario tree, while the characteristics of the two types of generators are shown in Table20. Value $\bar{D}$ is the mean demand considered in the deterministic model. We assume that the number of running units as we enter the modelling period is u_i^0, $i \in \mathscr{I}$. These units have a capacity of 800 mw, which is well above the expected need of $\bar{D} = 300$ mw during the first time period. Consequently, no generators are started up in period one ($s_i^1 = 0$, $i \in \mathscr{I}$) independently of the considered start up cost. The aim of the model is to select and allocate the power demands among an optimal number of operating units of types 1 and 2.

Table 18: Monthly production capacity a_i of suppliers $i \in \mathcal{I}$, January 2003 to May 2007, in kilotonnes (kt).

i	Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
	'03	9.1	4.0	11.1	14.6	21.7	14.2	17.4	8.4	24.9	17.4	12.3	13.0
	'04	0.0	4.1	9.0	10.5	9.3	12.2	11.6	13.6	9.4	11.0	9.7	0.0
AG	'05	0.0	9.1	8.3	21.1	15.0	15.1	12.1	13.2	11.3	13.0	7.1	1.2
	'06	1.7	9.5	4.5	14.0	12.5	15.2	11.3	15.9	6.2	11.9	7.2	9.0
	'07	13.0	13.0	19.0	4.0	10.0							
	'03	10.9	14.0	13.9	19.1	14.1	13.0	4.5	0.0	4.0	13.7	9.1	4.5
	'04	8.3	6.3	3.0	0.0	16.2	14.2	12.3	14.4	19.8	19.3	20.0	15.2
CS	'05	15.1	10.8	21.9	19.7	15.3	10.8	6.3	0.0	9.1	23.2	11.7	0.9
	'06	18.7	0.0	8.9	16.0	17.6	13.9	4.8	5.0	14.1	24.3	14.5	8.1
	'07	17.0	8.0	0.0	0.0	10.0							
	'03	15.5	18.1	23.3	12.4	0.5	5.7	12.5	13.5	12.3	10.2	8.3	12.0
	'04	27.1	10.0	12.8	13.8	13.7	14.0	10.6	1.4	10.3	12.6	11.5	16.9
PA	'05	16.0	3.8	10.6	16.6	23.0	27.7	16.7	13.4	16.8	11.1	19.0	22.4
	'06	27.5	21.5	18.6	20.4	0.0	14.0	14.3	11.2	18.4	16.9	9.4	11.1
	'07	11.0	9.0	7.0	6.0	10.0							
	'03	4.9	1.2	12.7	2.7	19.3	11.9	5.4	3.0	14.6	3.4	15.2	2.5
	'04	4.0	9.4	18.3	10.5	13.9	8.6	6.2	4.3	7.2	12.4	9.5	0.0
VV	'05	3.5	21.1	20.8	13.0	23.5	19.1	8.2	8.6	4.6	9.2	16.2	16.0
	'06	8.5	22.3	21.7	15.1	7.4	10.3	0.0	2.5	4.3	5.2	18.3	6.3
	'07	0.0	0.0	0.0	0.0	10.0							

Table 19: Predecessor $pa(n)$, energy demand D^n and probability p^n at node $n \in \mathscr{N}$ of the two-period scenario tree.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
$pa(n)$	-	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
D^n	300	300	605	630	580	650	600	520	100	180	130	100	120	102	50	41	100	102	125	69	600	596
p^n	1	1	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$
$\bar{D}$	300	300	300																			

Table 20: Costs and production characteristics for generators of type $i \in \mathscr{I}$.

	C_i (€)	E_i (€)	F_i (€)	m_i (mw)	M_i (mw)	u_i^0	$u_{i,max}$
$i = 1$	100	2500	14000	20	80	4	4
$i = 2$	150	5000	16000	30	120	4	4

1.3 Supply transportation problem

Deterministic and stochastic parameter values are reported below. Table 21 lists the set of suppliers $\mathscr{K}$ and the sets of their plants $\mathscr{O}_k$, $k \in \mathscr{K}$. The list of destinations (cement factories) is shown in Table 22 with the premium rates charged by the external company and the unloading capacities (expressed in tons of gypsum). Table 23 provides the minimum and maximum requirements for suppliers $k \in \mathscr{K}$ (again expressed in tons of gypsum). It is assumed that an initial inventory level of $l_j^0 = 0$ is available for all the destinations $j \in \mathscr{D}$. The capacity for all vehicles is fixed to $q = 31$ tons. The discount α is set to the value 0.7. The values of the transportation costs t_{ij} over all origins and destinations are in the following range: $[t_{ij}^{min}, t_{ij}^{max}] = [10.80, 73.52]$. Finally, the demand scenarios were obtained using historical data. Scenarios were built using the weekly demand values for the months of March, April, May and June of 2011, 2012 and 2013. Thus, a set of 48 weekly demand scenarios were obtained and assumed to be equiprobable.

Table 21: Set of suppliers $\mathscr{K}$ with their sets of plants $O_k,\ k \in \mathscr{K}$.

Supplier $k \in \mathscr{K}$	Plant $i \in \mathscr{O}_k$
1	$1, \ldots, 6$
2	7
3	8
4	4
5	9
6	$6, 10$
7	1
8	$1, 2$
9	11
10	12
11	13
12	14
13	15
14	12
15	8
16	16
17	17
18	9
19	$5, 15$
20	5
21	18
22	19
23	7
24	12

Table 22: List of destinations (cement factories) with emergency costs b_j and unloading capacities g_j, $j \in \mathscr{D}$.

Destination $j \in \mathscr{D}$	emergency cost b_j	Unloading capacity g_j
1	72.61	422.95
2	70.58	2054.55
3	68.01	1330.67
4	64.94	453.64
5	73.52	613.41
6	58.57	695.24
7	69.83	443.14
8	66.32	815.36
9	62.63	933.33
10	68.22	319.79
11	48.92	443.11
12	50.04	760.11
13	73.07	381.20
14	59.93	498.33
15	55.63	232411.75

Table 23: Minimum a_k and maximum v_k requirement capacity of supplier $k \in \mathscr{K}$.

Supplier $k \in \mathscr{K}$	a_k	v_k
1	1057.69	-
4	0	96.15
6	0	576.92
10	0	194.23
11	0	480.76
15	0	192.30
16	0	384.61