

Analyzing the quality of the expected value solution in stochastic programming

Francesca Maggioni* – Stein W. Wallace[§]

Abstract. Stochastic programs are usually hard to solve when applied to real-world 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 quality of the solution. The Value of the Stochastic Solution – *VSS* – is normally used to measure the importance of using a stochastic model. But what if *VSS* is large, or expected to be large, but we cannot solve the relevant stochastic program? Shall we just give up? In this paper we investigate very simple methods for studying structural similarities and differences between the stochastic solution and its deterministic counterpart. The aim of the methods is to find out, even when *VSS* is large, if the deterministic solution carries useful information for the stochastic case. It turns out that a large *VSS* does not necessarily imply that the deterministic solution is useless for the stochastic setting. Measures of the structure and upgradeability of deterministic solution such as the *loss using the skeleton solution* and the *loss of upgrading the deterministic solution* will be introduced and basic inequalities in relation to the standard *VSS* are presented and tested on different case studies.

Key words: stochastic programming, expected value problem, value of stochastic solution, quality of deterministic solution, skeleton solution, upgradeability of deterministic solution.

1 Introduction

The Value of the Stochastic Solution – *VSS* – has become a standard term in stochastic programming [2]. It measures the expected gain from solving a stochastic model rather than its deterministic counterpart (where all random variables are replaced by their means). It is mostly used to argue that stochastic programming models are necessary despite the efforts involved. In fact, if the *VSS* is low, it will almost by definition point to a weakness in the modelling itself: the modeller thought the uncertainty was important, when in fact, it was not. So, in these cases, uncertainty is dropped, normally accompanied by an understanding that the conclusion was “obvious”.

*Department of Mathematics, Statistic, Computer Science and Applications, Bergamo University, Via dei Caniana 2, Bergamo 24127, Italy. e-mail: francesca.maggioni@unibg.it

[§]Department of Management Science, Lancaster University Management School, Lancaster LA1 4YX, United Kingdom. e-mail: Stein.W.Wallace@lancaster.ac.uk

In this paper we are interested in cases where VSS is large. So uncertainty is important for the optimal solution, and the deterministic solution is “bad”. This is the normal case. However, in our view, stopping here is a bit simplistic. Stochastic programs, in particular stochastic integer programs, are close to impossible to solve for realistically sized problem. So even though VSS is high, and hence a stochastic program is appropriate, all we may have access to is the deterministic solution. So we may ask: isn’t there more to be said about the deterministic solution than that it is bad? We think there certainly is, and in this paper we ask: why is it bad, what is it that is bad about it? Has the deterministic solution picked the wrong variables (machines, processes, vehicle types...) or has it just assigned them bad values? Could it be that a bad deterministic solution (with a large VSS) actually carry a lot of information?

VSS is often used to describe problem classes, even though, in fact, VSS is instance dependent. Change one single number in the problem formulation, and VSS may change drastically. (See for instance the sensitivity analysis of VSS in [10] for the stochastic second order cone model for mobile-ad-hoc network that we will discuss later.) However, if we use VSS in this very strict sense, it becomes close to useless. So we tend to assume, probably correctly so, that if VSS is high (or low) for a given instance (or selection of instances) then it will also be high (or low) for other instances that have similar characteristics, such as larger instances of the same problem where the parameters balance in a comparable way. This is how we view VSS in this paper. This is crucial, since VSS can be found only for instances where the stochastic program can be solved, while our real interest will often be those cases where it cannot. So we proceed under the assumption that VSS reveals information also for larger and similar instances, fully understanding that the methodology we end up outlining, must be seen as heuristic.

So, in this paper we seek a deeper understanding of the expected value solution in order to investigate its relationship to its stochastic counterpart. For example, does the stochastic optimal solution *inherit* properties from the deterministic one, or are they totally different? A qualitative understanding of the behavior of the deterministic solution relative to the stochastic one could be very useful because it could reveal some general properties of the underlying problem and help us to predict how the stochastic model will perform in two important cases. Firstly, when the stochastic model is actually solvable, but since it is solved repeatedly (daily) we would like to actually solve the deterministic one, if we understood its qualities and how to interpret it, and secondly when it isn’t even solvable. Even in the case of a *bad* deterministic solution (large VSS), we would like to investigate and identify the reasons: is it because of a wrong choice of positive variables (like which facilities to open, what ditches to dig) or is it a value problem, the choice of variables is fine, but the values (like capacities) are off? Can we obtain a good (if not optimal) solution to the stochastic problem by updating the deterministic one?

We believe that a deeper understanding of the relationship between the optimal solutions to stochastic and deterministic models can also be useful in order to understand what to do when a situation is random but the actual numbers/distributions are not known. This applies to algorithmic developments as well as practical use of models in industry and government.

We shall be able to say what is potentially wrong with a solution coming from a deterministic model even if we cannot solve the stochastic one since we don't have data. And once we know what is wrong with the deterministic solution we may be able to compensate to arrive at a better (albeit not optimal) solution.

To motivate our development we introduce beside *VSS*, other measures of badness/goodness of deterministic solutions according to the type of model considered (linear, integer linear, mixed integer linear or non-linear). We limit our analysis to the two-stage stochastic case, even if the investigation could be extended also to multi-stage models. For this purpose we introduce the *loss using skeleton solution LUSS* and the *loss of upgrading the deterministic solution LUDS* which gives deeper information than *VSS* on the structure of the problem. *LUSS* and *LUDS* could be useful to take a fast "good" decision instead of using expensive direct techniques. Basic inequalities for these quantities in relation to *VSS* are also presented.

The paper is organized as follow: basic facts and notations are introduced in Section 2. Subsection 2.1 explains the set-up of our experiments, whereas details and mathematical formulation of the problems we consider for our investigation are explained in Section 3 with a discussion on the numerical results. Section 4 concludes the paper.

2 Basic facts and notations

Let us recall the standard notation that we are going to use in this paper. The following mathematical model represents a general formulation of a stochastic program in which a decision maker has to take a decision x in order to minimize (expected) costs or outcomes:

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

where x is a first-stage decision variable restricted to the set $X \subset \mathbb{R}^n$, E_{ξ} denotes the expectation with respect to a random vector ξ , defined on some probability space (Ω, \mathcal{A}, p) with support $\Omega \in \mathbb{R}^K$ and given probability distribution p on the σ -algebra \mathcal{A} . The function h_2 is the value function of another optimization problem defined as follow:

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

and it is used to reflect the costs associated with adapting to information revealed through a realization ξ of the random vector ξ . The term $E_{\xi} [h_2(x, \xi)]$ in (1) is referred to as the recourse function.

The solution x^* obtained by solving problem (1), is called the *here and now solution* and

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

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:

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

where $\bar{\xi} = E(\xi)$. 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 $\bar{x}(\bar{\xi})$:

$$EEV = E_{\xi} (z(\bar{x}(\bar{\xi}), \xi)) . \quad (5)$$

The value of the stochastic solution is then defined as

$$VSS = EEV - RP , \quad (6)$$

and it measures the expected increase in value from solving the stochastic version of a model rather than the simpler deterministic one. Relations and bounds on EV , EEV and RP can be found for instance in [2] and [3].

2.1 Does the stochastic solution inherit properties from the deterministic one?

In general it is well known that the expected value solution can behave very badly in a stochastic environment. However, it is not always 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 try to answer this question by means of a set of tests which allows us to investigate, even in the case of a large VSS , how the expected value solution relates to its stochastic counterpart. Are there common properties even if the expected value solution is bad in its own right? If there are, these may be useful for understanding large instances of the stochastic model even if we cannot solve them, and even if the expected value solution itself is useless (see for example [14]). We consider the following tests, adopted from [14]:

Test A: The classical evaluation of the expected value solution $\bar{x}(\bar{\xi})$. We calculate its expected performance by computing $EEV = E_{\xi} (z(\bar{x}(\bar{\xi}), \xi))$ and compare it with RP using $VSS = EEV - RP$.

Test B: 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 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. Notice that if the original problem is a:

- linear program, then test B 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, then we still solve a MIP but of smaller dimension.

Let \mathcal{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. Then let \hat{x} be the solution of:

$$\begin{aligned} \min_{x \in X} \quad & E_{\xi} z(x, \xi) \\ \text{s.t.} \quad & x_j = \bar{x}_j(\bar{\xi}), \quad j \in \mathcal{J}. \end{aligned} \quad (7)$$

We then compute the *expected skeleton solution value*

$$ESSV = E_{\xi} (z(\hat{x}, \xi)) , \quad (8)$$

and we compare it with RP by means of *loss using skeleton solution*

$$LUSS = ESSV - RP . \quad (9)$$

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

$$RP \leq ESSV \leq EEV , \quad (10)$$

and consequently,

$$EEV - EV \geq VSS \geq LUSS \geq 0 . \quad (11)$$

Notice that the case $LUSS = 0$ (i.e. $ESSV = RP$) 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 $x_j^* \neq \bar{x}_j(\bar{\xi})$. We shall observe $LUSS = VSS$ if the stochastic program, when not allowed to use the variables in \mathcal{J} , chooses not to change the value of any of the remaining variables (i.e. $\hat{x} = \bar{x}(\bar{\xi})$).

Test C: Consider the expected value solution $\bar{x}(\bar{\xi})$ as a starting point (an input) to the stochastic model (1) and compare, in terms of objective functions, to (1) without such input. So we test if the expected value solution is upgradeable to become good (if not optimal) in the stochastic setting. This is equivalent to adding in problem (1) the constraint $x \geq \bar{x}(\bar{\xi})$ and hence solve the following problem with solution \tilde{x} :

$$\begin{aligned} \min_{x \in X} \quad & E_{\xi} z(x, \xi) \\ \text{s.t.} \quad & x \geq \bar{x}(\bar{\xi}) . \end{aligned} \quad (12)$$

We then compute the *expected input value*

$$EIV = E_{\xi} (z(\tilde{x}, \xi)) \quad (13)$$

and we compare it with RP , by means of the *loss of upgrading the deterministic solution*:

$$LUDS = EIV - RP . \quad (14)$$

We have:

$$RP \leq EIV \leq EEV , \quad (15)$$

and consequently,

$$EEV - EV \geq VSS \geq LUDS \geq 0 . \quad (16)$$

Notice that $LUDS = 0$ (i.e. $EIV = RP$) corresponds to *perfect upgradability* of the deterministic solution, a case in which the conditions $x \geq \bar{x}(\bar{\xi})$ are satisfied by the stochastic solution x^* even without being enforced by constraints (under the assumption that the stochastic first-stage decision is unique), that is $\tilde{x} = x^*$; on the other hand, $0 < LUDS < VSS$, if there exists a component $i \in \{1, \dots, n\}$ such that $x_i^* < \bar{x}_i$, then $\tilde{x}_i = \bar{x}_i$ (case of *partial upgradability*). The case $LUDS = VSS$ corresponds to the *not upgradability* in which the condition $x \geq \bar{x}(\bar{\xi})$ is no longer satisfied by any of the components of solution x^* and then $\tilde{x} = \bar{x}(\bar{\xi})$ (i.e. $EIV = EEV$).

Test D: This is a generalization of Test B. According to the interpretation of the variables involved and the actual model, partial information is imported from the expected value solution. Notice that for a mixed-integer program MIP with integer variables only at the first stage, the test reduces the problem to be linear. Furthermore if the integer variables are restricted to be binary, the test coincides with test B. On the other hand, if all first stage variables are integer, the test reduces to the computation of VSS (Test A).

To show the usefulness of these simple tests, we now apply these experiments on a set of stochastic programs of different types. They will be described in Section 3:

- Mixed integer stochastic program for a “single-sink transportation problem”;
- Mixed integer stochastic program for a “furniture company problem”;
- Second order cone stochastic program for a “mobile ad-hoc network problem”;
- Mixed integer stochastic program for a “power generation scheduling problem”.

We shall see that while no test is useful for all problems, and no problem benefits from all tests, in total, we shall get a good evaluation of the expected value solution.

3 Problems description

3.1 Stochastic optimization models for 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 [11]. 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 reservations 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 the following:

Sets:

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

Parameters:

$$\begin{aligned}t_i &: \text{ unit transportation costs of supplier } i \in \mathcal{I} ; \\ c_i &: \text{ unit production costs of supplier } i \in \mathcal{I} ; \\ b &: \text{ buying cost from an external source (we suppose } b > \max_i(t_i + c_i) \text{) ;} \\ q &: \text{ vehicle capacity ;} \\ g &: \text{ unloading capacity at the customer ;} \\ l_0 &: \text{ initial inventory level at the customer ;} \\ l_{\max} &: \text{ storage capacity at the customer ;} \\ p^k &: \text{ probability of scenario } k \in \mathcal{K} ; \\ a_i^k &: \text{ production capacity of supplier } i \in \mathcal{I} \text{ in scenario } k \in \mathcal{K} ; \\ d^k &: \text{ customer demand at scenario } k \in \mathcal{K} ; \\ \alpha &: \text{ cancellation fee ;}\end{aligned}$$

Variables:

$$\begin{aligned}x_i \in \mathbb{N} &: \text{ number of vehicles booked from supplier } i \in \mathcal{I} ; \\ z_i^k \in \mathbb{N} &: \text{ number of vehicles actually used from } i \in \mathcal{I} \text{ in } k \in \mathcal{K} ; \\ y^k &: \text{ product to purchase from an external source in scenario } k \in \mathcal{K} ;\end{aligned}$$

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

$$\min q \sum_{i=1}^I t_i x_i + \sum_{k=1}^K p^k \left[b y^k - (1 - \alpha) q \sum_{i=1}^I t_i (x_i - z_i^k) \right] \quad (17)$$

$$\text{s.t.} \quad q \sum_{i=1}^I x_i \leq g \quad (18)$$

$$l_0 + \sum_{i=1}^I qz_i^k + y^k - d^k \geq 0, \quad k \in \mathcal{K} \quad (19)$$

$$l_0 + \sum_{i=1}^I qz_i^k + y^k - d^k \leq l_{\max}, \quad k \in \mathcal{K} \quad (20)$$

$$z_i^k \leq x_i, \quad i \in \mathcal{I}, k \in \mathcal{K} \quad (21)$$

$$qz_i^k \leq a_i^k, \quad i \in \mathcal{I}, k \in \mathcal{K} \quad (22)$$

$$x_i \in \mathbb{N}, \quad i \in \mathcal{I} \quad (23)$$

$$y^k \geq 0, \quad k \in \mathcal{K} \quad (24)$$

$$z_i^k \in \mathbb{N}, \quad i \in \mathcal{I}, k \in \mathcal{K} \quad (25)$$

The first sum in the objective function (17) is the booking costs of the vehicles, while the second sum represents the recourse actions, consisting of buying extra clinker (y^k) and canceling unwanted vehicles. Constraint (18) guarantees that the total quantity delivered from the suppliers to the customer is not greater than the customer's unloading capacity g , inducing thus an upper bound on the total number of vehicles. Constraints (19) and (20) ensure that the second-stage storage level is between zero and l_{\max} . Constraint (21) guarantees that the number of vehicles servicing supplier i is at most equal to the number booked in advance and (22) controls that the quantity of clinker delivered from supplier i does not exceed its production capacity a_i^k . Finally, (23)–(25) define the first- and second-stage decision variables of the problem.

3.2 Stochastic optimization model for a furniture company

The second example we consider is the production problem described in [6]. The Dakota Furniture Company manufactures desks, tables and chairs under uncertain demand. Different approaches can be considered in order to capture the relationship between time at which the decisions have to be taken and the time at which the demand is revealed. In the following we will use model (P.2) presented in [6] where the acquisition of raw materials and the production of furniture have to be determined before the demand is revealed. Each type of furniture requires lumber and two types of skilled labor: carpentry and finishing. Dakota wants to determine how much of each item to produce and the resources required to meet the production in order to maximize the profit. The notation adopted is as follows.

Sets:

$$\begin{aligned} \mathcal{P} &= \{p : p = 1, \dots, P\} && : \text{ set of products (desk, tables or chairs);} \\ \mathcal{W} &= \{w : w = 1, \dots, W\} && : \text{ set of board feet of lumber/hours of carpentry, finishing;} \\ \mathcal{K} &= \{k : k = 1, \dots, K\} && : \text{ set of possible scenarios.} \end{aligned}$$

Parameters:

- D_p^k : demand for product $p \in \mathcal{P}$ in scenario $k \in \mathcal{K}$;
- c_w : unit cost for work $w \in \mathcal{W}$;
- e_p : selling price for product $p \in \mathcal{P}$;
- p^k : probability of scenario $k \in \mathcal{K}$;
- $m_{w,p}$: the production requirements for product $p \in \mathcal{P}$.

The decision variables are:

- y_p : number of product $p \in \mathcal{P}$ to produce;
- x_w : number of board feet/hours of work $w \in \mathcal{W}$ to use;
- s_p^k : number of product $p \in \mathcal{P}$ to sell in scenario $k \in \mathcal{K}$;

Dakota's problem can be formulated as the following mixed integer stochastic program:

$$\max - \sum_{w \in \mathcal{W}} c_w x_w + \sum_{k \in \mathcal{K}} p^k \sum_{p \in \mathcal{P}} e_p s_p^k \quad (26)$$

$$\text{s.t.} \quad -x_w + \sum_{p \in \mathcal{P}} m_{w,p} y_p \leq 0, \quad w \in \mathcal{W} \quad (27)$$

$$s_p^k - D_p^k \leq 0, \quad p \in \mathcal{P}, k \in \mathcal{K} \quad (28)$$

$$s_p^k - y_p \leq 0, \quad p \in \mathcal{P}, k \in \mathcal{K} \quad (29)$$

$$x_w \geq 0, \quad w \in \mathcal{W} \quad (30)$$

$$y_p \in \mathbb{N}, \quad p \in \mathcal{P} \quad (31)$$

$$s_p^k \in \mathbb{N}, \quad p \in \mathcal{P}, k \in \mathcal{K} \quad (32)$$

The objective function (26) represents the maximization of income from selling the items minus the production costs. Constraint (27) guarantees that the resources acquired satisfy the production schedule, constraint (28) ensures that the production meets the demand, constraint (29) means that the number of products sold does not exceed the production. Finally constraints (30)-(32) define the first- and second-stage decision variables of the problem. We refer to [6] for data used in the simulations and numerical results.

3.3 Stochastic location-aided routing in mobile ad hoc networks

We consider a location-aided routing problem in a wireless ad-hoc network [9]. A wireless ad-hoc network consists of a group of mobile nodes that communicate with each other in the absence of a fixed infrastructure by virtue of their proximity. Because of the scarcity of wireless channels and the mobility of the wireless nodes, the design of routing protocols is a crucial issue in mobile ad-hoc networks and a number of routing protocols have been proposed with

the goal of searching for a route when hosts move. One of these algorithm is the Stochastic Location-Aided Routing (SLAR) [8], based on the use of local information (given for example by the Global Positioning System, GPS). SLAR tries to reduce the search space for a desired route. When a *sender node*, say S , needs to find a route to a *destination node*, say D , S broadcasts a route request to all its neighbors. On receiving the route request, D responds by sending a route reply message to S following the reverse path of the route request received by D . If the route request message does not get to D , the protocol allows S to initiate a new route discovery with an expanded request zone.

The main concepts behind the algorithm are the *expected zone* and *requested zone*. The former is the region in the shape of a circle (see Figure 1), where S (for simplicity fixed at the origin $0 \in \mathbb{R}^n$) expects to find D after an elapsed time t_1 , based on the knowledge that node D was located at l at time t_0 and its lowest velocity is v . The latter is the region defined by S which includes the expected zone, for spreading the route request to reach D in case it does not belong to the former zone.

One of the main characteristics of a mobile ad hoc network is the mobility of the nodes. Instead of a deterministic approaches, the SLAR algorithm models the movement speed and direction of the typical user D by random variables, giving for instance more probability to a particular direction. The movements of D are then represented by ellipsoid scenarios E^k , $k \in \mathcal{K}$ (see Figure 1), randomly generated by uniform and normal distributions in a neighborhood of the starting position l of the destination node. This choice corresponds to a typical real situation in which people are moving along preferred directions (for example different motorways) identified by the length of the main semiaxis σ_1^k and angle φ^k of the ellipsoid E^k with the possibility to exit from the motorway for short distances (length of the second semiaxis σ_2^k).

SLAR uses the following three-stage procedure:

1. calculate the initial expected zone (circle C) where the destination node is expected to be at time t_1 ; the disk C is required to contain the smallest disk C_0 centred in l and radius $v(t_1 - t_0)$ corresponding to the minimum speed v at which D is supposed to move (assuming a radial direction). The route request is then sent from the source node to cover this circle. Notice that the main decisions at this stage are the center $\tilde{u} \in \mathbb{R}^n$ and radius $r = \sqrt{\tilde{u}^T \tilde{u} - \gamma}$ of the circle:

$$C = \{u \in \mathbb{R}^n : u^T u - 2\tilde{u}^T u + \gamma \leq 0\} . \quad (33)$$

2. The route request is sent out to look for D by flooding inside the expected zone C . If D is in C , no further action is needed (see the ellipsoid E^1 in Figure 2); the route request reaches the destination and the reply message is sent back to the source. Then a route is established between the source and the destination node.
3. In case the destination node D is not found in stage 2, D should be in an ellipsoid E^k , $k \in \mathcal{K}$, not covered by C (see ellipsoids E^2 and E^3 in Figure 2). The disk C is then enlarged in order to cover the ellipsoid E^k and to get a new circle $C^{*,k}$ (requested zone)

$$C^{*,k} = \{u \in \mathbb{R}^n : u^T u - 2\tilde{u}^T u + \gamma - \zeta^k \leq 0\} , \quad (34)$$

with the same center $\tilde{u} \in \mathbb{R}^n$ of C and radius $\sqrt{\tilde{u}^T \tilde{u} - \gamma + \zeta^k}$ enlarged by the quantity $\zeta^k \in \mathbb{R}^+ \cup \{0\}$.

A key step is to determine a cost-effective initial expected zone so as to balance the message flooding cost with latency to reach the destination node D . The cost of choosing the expected region C is proportional to the distance $d_1 = \sqrt{\tilde{u}^T \tilde{u}}$ of the centre \tilde{u} from the source node S and to the radius r . In [9] second order cone constraints are considered to describe the inclusion of the disk C_0 into C and of the ellipsoid E^k in the second stage circle $C^{*,k}$, $k \in \mathcal{K}$. The stochastic second order cone formulation (SSOCP) allows to solve the problem with a much larger number of scenarios (20250) than what is possible with a semidefinite formulation [1]. We refer to [9] for details on the stochastic second order cone model and on scenarios generation procedure.

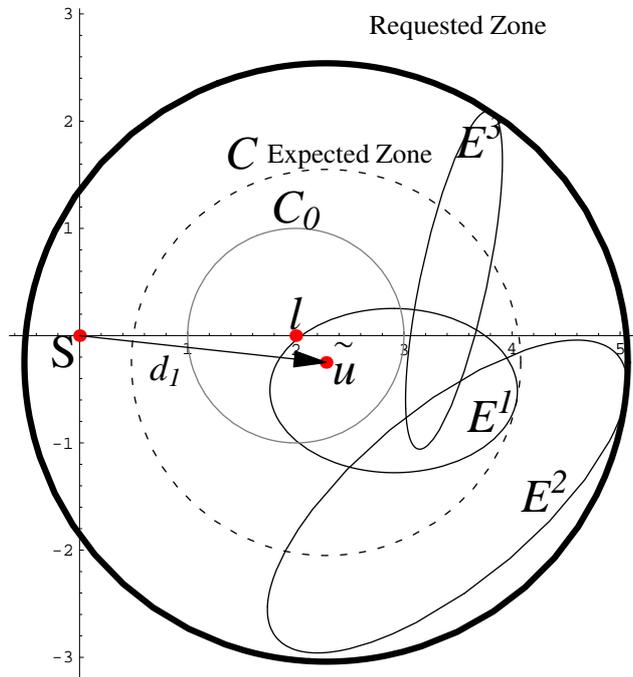


Figure 1: Expected zone (dashed circle) with SLAR algorithm in the case of the ellipsoid scenarios E^1 , E^2 and E^3 .

3.4 Stochastic optimization model for power generation scheduling

Our last problem is based on an economic scheduling model formulated in [13] and [5] as a deterministic mixed integer program. Power generation scheduling involves the selection of units to be put into operation and the allocation of the power demand among operating units. We consider here a 2-stage stochastic version of the model presented in [13]; it is written in terms of nodes of the scenario tree, built on the uncertain energy demand at the second time period. So production decisions are made after demand has been revealed. The following formulation is considered.

Sets:

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

Parameters:

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

The decision variables are:

$$\begin{aligned} u_i^n & : \text{ number of generating units of type } i \in \mathcal{I} \text{ working in node } n \in \mathcal{N}; \\ s_i^n & : \text{ number of generators of type } i \in \mathcal{I} \text{ started up in node } n \in \mathcal{N}; \\ x_i^n & : \text{ total output rate from generators of type } i \in \mathcal{I} \text{ in node } n \in \mathcal{N}. \end{aligned}$$

A formulation of the generator scheduling problem as an integer program including start-up costs can be as follow:

$$\sum_{n \in \mathcal{N}} p^n \left[\sum_{i \in \mathcal{I}} C_i (x_i^n - m_i u_i^n) + \sum_{i \in \mathcal{I}} E_i u_i^n + \sum_{i \in \mathcal{I}} F_i s_i^n \right] \quad (35)$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}} x_i^n \geq D^n, \quad n \in \mathcal{N} \quad (36)$$

$$x_i^n \geq m_i u_i^n, \quad i \in \mathcal{I}, n \in \mathcal{N} \quad (37)$$

$$x_i^n \leq M_i u_i^n, \quad i \in \mathcal{I}, n \in \mathcal{N} \quad (38)$$

$$\sum_{i \in \mathcal{I}} M_i u_i^n \geq \frac{115}{110} D^n, \quad n \in \mathcal{N} \quad (39)$$

$$s_i^n \geq u_i^n - u_i^{pa(n)}, \quad i \in \mathcal{I}, n \in \mathcal{N} \setminus \{1\} \quad (40)$$

$$u_i^1 = u_i^0, \quad i \in \mathcal{I} \quad (41)$$

$$u_i^n \leq u_{i,max}, \quad i \in \mathcal{I}, n \in \mathcal{N} \quad (42)$$

$$x_i^n \geq 0, \quad i \in \mathcal{I}, n \in \mathcal{N} \quad (43)$$

$$s_i^n \in \mathbb{N}, \quad i \in \mathcal{I}, n \in \mathcal{N} \quad (44)$$

$$u_i^n \in \mathbb{N}, \quad i \in \mathcal{I}, n \in \mathcal{N} \quad (45)$$

The objective function (35) consists in the minimization of the total costs of starting, producing power at minimum output and producing power above the minimum output for each time period. Constraint (36) guarantees that demand must be met in each period, whereas (37) and (38) make sure that output lies within the limits of the operating generators. Constraint (39) means that the extra guaranteed load requirement must be able to be met without starting up any more generators and (40) that the number of generators started in node n must equal the increase in number with respect to the node $pa(n)$ of the previous period. Finally constraints (41)-(42) define starting values and upper bound of open units and (43)-(45) the decision variables of the problem.

3.5 Comparison tests for the “single-sink transportation problem”

Tests A, B and C are performed for the single-sink transportation problem described in Section 3.1. The model aims to find, for each supplier, the number of vehicles to book at the beginning of January 2007.

Test A We compare the solution to the stochastic model (17)-(25) with the expected value problem (4). Solutions to the deterministic model are reported in Table 1: the model will always book the exact numbers of vehicles needed for the next period (so $\bar{x}_i = \bar{z}_i^k$, $i \in \mathcal{I}$, $k \in \mathcal{K}$); it sorts the suppliers according to the transportation costs and books a full production capacity from the cheapest one (AG), following by the next-cheapest (PA).

The deterministic model 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 (€495 788 instead of the predicted cost of €294 898) resulting in

$$VSS = 495\,788 - 438\,304 = 57\,384, \quad (46)$$

which shows that we can save about 12% of the cost by using the stochastic model, compared to the deterministic one.

Table 1: Optimal solutions from tests A, B and C for the “single-sink transportation problem”. The table shows optimal number of booked vehicles for each supplier and total optimal costs.

	AG	CS	PA	VV	Objective value (€)
deterministic	206	0	530	0	294 898=EV
stochastic	400	0	563	117	438 301=RP
Test A	206	0	530	0	495 788=EEV
Test B	400	0	637	0	462 214=ESSV
Test C	400	0	563	117	438 301=EIV

Why is the deterministic solution bad? Because of a too optimistic guess on the randomness (leading to too *few booked vehicles* from the four suppliers) or because of the *wrong suppliers*? We perform the following tests:

Test B We follow the skeleton solution from the deterministic model, not allowing to book vehicles from CS and VV. The *Expected skeleton solution value ESSV* is then €462 214, still higher than *RP* with a consequent *loss using the skeleton solution* of

$$LUSS = 462\,214 - 438\,304 = 23\,910, \quad (47)$$

which measures the loss by booking vehicles coming only from suppliers AG and PA as suggested by the deterministic model. We can conclude that the deterministic solution is bad because it books the wrong number of vehicles from the wrong suppliers.

Notice that this approach requires us to solve a MIP but with smaller dimension than the original problem.

Test C The number of vehicles booked from AG and PA in the deterministic solution $\bar{x}_i(\bar{d}, \bar{a}_i)$, $i \in \mathcal{I}$ is taken as input in the stochastic model and we check if the solution can be upgraded in a second run. The test amounts to adding to the stochastic model (17)-(25) the constraint $x_i \geq \bar{x}_i(\bar{d}, \bar{a}_i)$, $i \in \mathcal{I}$ and solve it. Notice that for all the four suppliers the constraint is automatically satisfied, as the booked number vehicles in the stochastic solution is higher than in the deterministic one (see Table 1) with $LU DS = 0$. Hence, the deterministic solution is *perfectly upgradeable*.

In conclusion the deterministic solution does not perform well in a stochastic environment because of the too low number of vehicles booked at the first stage (736 instead of 1080) just considering AG and PA as possible suppliers. However the company can consider the deterministic solution as a lower bound for the stochastic case. This might be useful information.

3.6 Comparison tests for “furniture company problem”

Tests from the previous section are now performed for the Dakota furniture problem described in Section 3.2. The model aims to find how many items to produce and resources to acquire to meet the demand.

Table 2: Optimal solutions from tests A, B, C and D for “Dakota furniture company problem”. The table shows optimal number x_w , $w \in \mathcal{W}$ of resources to acquire to produce item y_p , $p \in \mathcal{P}$ (desks, tables and chairs) and total profit.

	x_1	x_2	x_3	y_1	y_2	y_3	Objective value (€)
deterministic	1 950	850	487.5	150	125	0	4 165=EV
stochastic	1 060	420	265	50	110	0	1 142=RP
Test A	1 950	850	487.5	150	125	0	865=EEV
Test B	1 060	420	265	50	110	0	1 142=ESSV
Test C	1 950	850	487.5	150	125	0	865=EEV=EIV
Test D	1 950	850	487.5	110	150	20	885

Test A The stochastic model (26)-(32) is compared with the expected value problem (see Table 2). Sensitivity analysis on the deterministic results indicates that the solution “produce as many desks and tables as can be sold ($\bar{y}_1 = \bar{s}_1$ $\bar{y}_2 = \bar{s}_2$), but do not produce any chairs ($\bar{y}_3 = 0$)” remains valid for any set of (nonnegative) demands, thus in particular for the mean value \bar{D}_p . Hence,

$$VSS = 1142 - 865 = 277 \quad (48)$$

showing that we lose about 25% of the total profit by implementing the deterministic solution.

Why is the deterministic solution bad? Is it because of the acquisition of *too many resources*? Or because of the *wrong number* of items are produced? Or because the *wrong types of items* are produced (desks and tables instead of chairs)? The following tests help us to find an explanation.

Test B As in the deterministic solution we do not allow the production of chairs ($y_3 = 0$), a condition already satisfied by the stochastic solution (see Table 2). This leads to $LUS = 0$ which means that the deterministic solution has a perfect structure (case of *perfect skeleton solution*) producing the right items (desks and tables), but plans to acquire too many resources and to produce too many desks for the demand in the market ($\hat{x}_w = x_w^*$, $\hat{y}_p = y_p^*$, $w \in \mathcal{W}$, $p \in \mathcal{P}$). From an algorithmic perspective, we still solve a mixed integer stochastic linear program but with smaller dimension than the original.

Test C We check the upgradeability for resources acquired and items produced by the deterministic solution in the stochastic environment. None of the conditions $x_w \geq \bar{x}_w(\bar{D}_p)$, $y_p \geq \bar{y}_p(\bar{D}_p)$, $w \in \mathcal{W}$, $p \in \mathcal{P}$ are satisfied by the stochastic solution, and consequently $\tilde{x}_w = \bar{x}_w(\bar{D}_p)$, $\tilde{y}_p = \bar{y}_p(\bar{D}_p)$, $w \in \mathcal{W}$, $p \in \mathcal{P}$ will be the solution to the constrained stochastic program of Test C. This is a case of *no upgradeability* as the deterministic solution is useless as a starting point for a stochastic program describing potential updates of the deterministic solution. We have $LUDS = VSS = 277$.

Test D We now fix just the resource quantities from the deterministic solution $x_w = \bar{x}_w(\bar{D}_p)$, $w \in \mathcal{W}$, $p \in \mathcal{P}$, allowing the stochastic model to decide on the number of desks, tables and chairs to produce. So we are solving an integer program. Because of the high amounts resources acquired (see solutions reported in Table 2), the model produces too many items for the demand in the market. The profit is still as much as 22.5% worse than the stochastic one.

We can conclude that the deterministic solution produces the right items (desks and tables) but is bad because it overestimates both the amounts of resources to acquire and the number of items to produce for the needs of the market.

3.7 Tests for “Mobile ad-hoc network problem”

We refer to the problem described in Section 3.3 and to [9] for details on data in the simulation and scenarios generation technique. We performed a sensitivity analysis to see how the VSS depended on the second stage cost q_1 (see [10]), and ended up with $\alpha = \beta = 1$ and $q_1 = 1.5$, paying more for a corrective decision than the original one. With the numbers in the underlying paper, we found $VSS = 0$, which we did not find very useful for our analysis.

Test A Standard evaluation of the deterministic solution associated to the mean scenario E_{mean} (see Figure 2) with center $(\bar{u}_1, \bar{u}_2) = (2.5056, -0.2461)$, angle $\bar{\varphi} = 1.2866$, and semi-axes $\bar{\sigma}_1 = 1.7448$ and $\bar{\sigma}_2 = 0.8586$, respectively, given as the means of centres, angles and semi-axes of the ellipses E^k , $k = 1, \dots, 5$ (see [9]).

Because in a deterministic problem the future is completely known, a recourse action is not required and the consequent total cost is lower (5.38 instead of 8.52 of the stochastic case). The resulting expected region (see Figure 2 and Table 3) appears to be too small to be useful in practice (the radius r is 1.8 instead of 2.07 of the stochastic case) and the centre is located furthestmost from the sender node S (the distance $d_1 = 2.16$ instead of 2.02).

Table 3: Optimal solutions from the deterministic and stochastic models and tests A and D. The table shows optimal first stage variables, the radius r of the circle C and costs.

	d_1	d_2	\tilde{u}_1	\tilde{u}_2	γ	τ	r	1 st st. costs	Objective value
deterministic	2.16	3.23	2.15	-0.14	1.42	1.75	1.80	5.38	5.38
stochastic	2.02	4.27	2.01	-0.23	-0.17	2.49	2.07	6.29	8.52
Test A	2.16	3.23	2.15	0.14	1.42	1.75	1.80	5.38	9.31
fixing det. centre	2.15	4.98	2.15	-0.14	-0.34	2.70	2.23	7.13	8.77
fixing det. d_1	2.16	4.70	2.15	-0.14	-0.04	2.63	2.17	6.86	8.59
fixing det. radius	2.08	-3.23	2.07	-0.23	1.12	1.99	1.80	5.31	9.03

The value of the stochastic solution is given by

$$VSS = 9.31 - 8.52 = 0.79 , \quad (49)$$

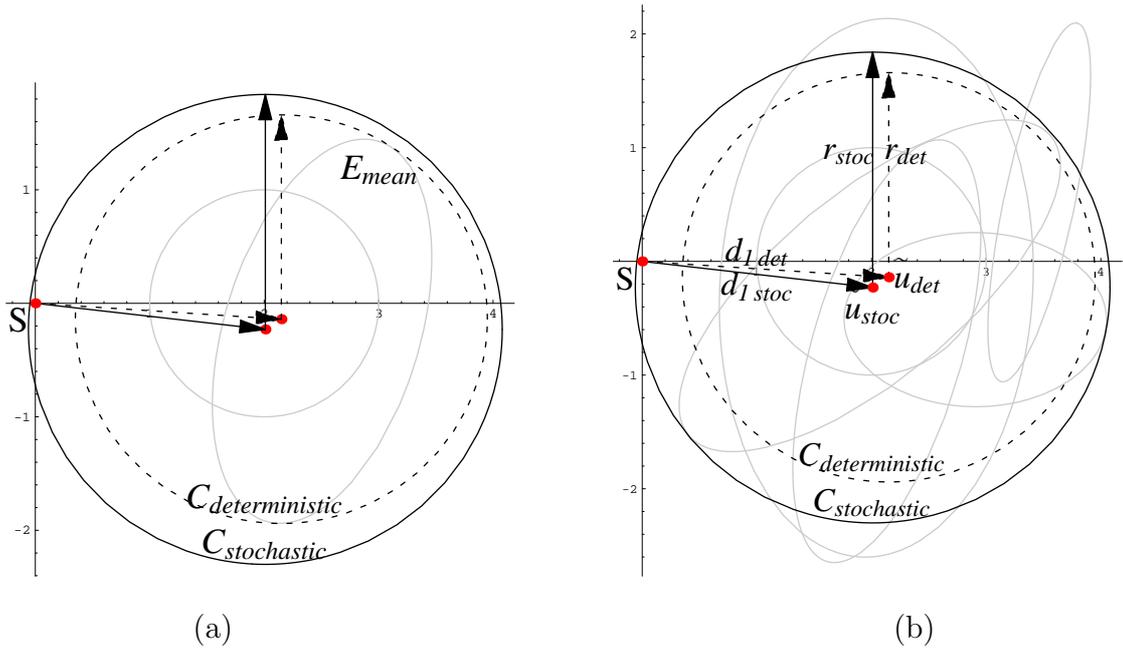


Figure 2: Comparison between the stochastic solution $C_{stochastic}$ and the *mean value solution* (dashed circle $C_{deterministic}$) for the mean scenario E_{mean} (plotted in (a)), and mean of the ellipses scenarios E^1, \dots, E^5 (plotted in (b)).

which shows that we save about 9.27% of the cost by using the stochastic model instead of the deterministic one.

What is wrong in the mean value circle $C_{deterministic}$? The location of the centre (too far from the sender node S) or the small radius r ? We develop the following tests where we fix separately the first stage decision variables (centre \tilde{u} , radius r and distance d_1 of the centre from the source node S). See results in Table 3 and Figure 3.

Test D First we take the centre from the deterministic circle $C_{deterministic}$ allowing the stochastic model to decide on the radius. This choice implies a small loss (2.93%) because of the possibility to cover the random movement of the destination node D through a circle with larger radius (dotted-dashed line circle in Figure 3(a)) than in the stochastic case (solid line).

A weaker option is obtained by forcing the distance of the centre from S at the deterministic value $d_1 = 2.16$ with a loss of just 0.82%.

Reasonably, the worst case is when we fix the radius from the deterministic solution: the expected zone (dotted-dashed line circle in Figure 3(b)) is simply too small. The percentage losses with respect the stochastic recourse problem increases to 5.98%.

By the tests we can conclude that the deterministic model delivers a good choice of the centre but not of the radius, as it is too small to contain the larger ellipsoid scenarios. Hence, there is something to be learned from the deterministic solution.

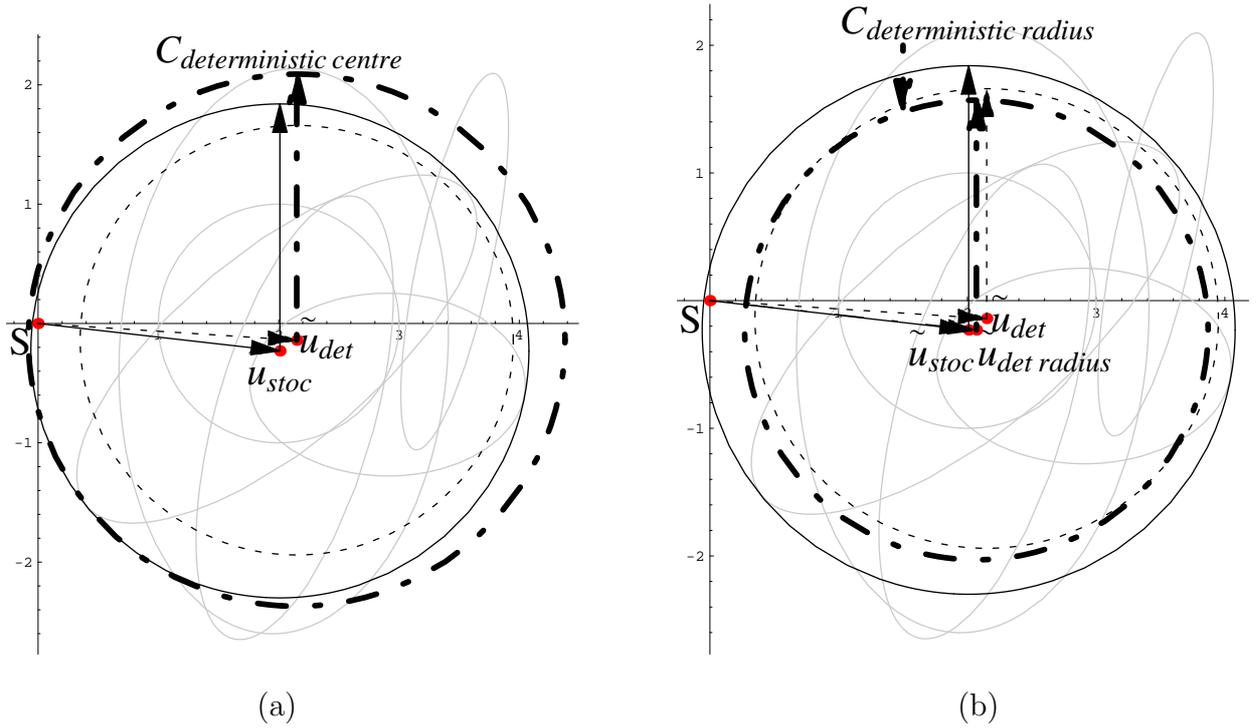


Figure 3: Comparison between the stochastic solution (solid line circle) and the expected region (dotted-dashed circle) with (a) centre and (b) radius from the deterministic solution (dashed circle).

3.8 Comparison tests for power generation problem

Table 4 reports energy demand on the nodes $n \in \mathcal{N}$ of the scenario tree, while characteristics of the two types of generators are shown in Table 5. We assume that the number of running units as we enter the modelling period is u_i^0 , $i \in \mathcal{I}$. These units have a capacity of 800 mw, well above the expected need of $\bar{D} = 300$ mw during the next time period. A natural consequence is that no generators will be started up in period one ($s_i^1 = 0$, $i \in \mathcal{I}$) independently of the start up cost. The aim of the model is to select and allocate the power demand among an optimal number of operating units of types 1 and 2.

Table 4: Energy demand D^n and probability p^n at node $n \in \mathcal{N}$ of the two-period (one proper stage) scenario tree. \bar{D} represents the mean demand considered in the deterministic model.

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

Test A Here we evaluate of the expected value solution under the mean scenario $\bar{D} = 300$

Table 5: Costs and production characteristics for generators of type $i \in \mathcal{I}$.

	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

Table 6: Optimal solutions from tests A, B, C and D for “energy power generation problem”. The table shows first stage solutions of generating units u_i^1 , the number of started up generators s_i^1 , total output rate x_i^1 ($i \in \mathcal{I}$) and total cost.

	u_1^1	u_2^1	s_1^1	s_2^1	x_1^1	x_2^1	Objective value (€)
deterministic	4	0	0	0	300	0	104 000=EV
stochastic	4	3	0	0	210	90	115 477.5=RP
Test A	4	0	0	0	300	0	127 877.5 =EEV
Test B	4	0	0	0	300	0	127 877.5=ESSV
Test C	4	3	0	0	210	90	115 477.5=RP
Test D	4	0	0	0	300	0	127 877.5 =EEV

mw in the stochastic environment (35)-(45). Solutions are reported in Table 6: the deterministic model closes down as many units as possible for the demand, ending up with only four units of type 1. We observe this result in many test – the deterministic solution closes down as many units as it can. Because the deterministic solution keep only 4 units running instead of 7 (4+3) (as in the stochastic one), the resulting total cost in the model itself reduces to 104 000 € against 115 477 € of the stochastic counterpart. However the 4 units working in the deterministic solution are not enough to satisfy the high demand scenarios in the second stage, bringing us to a

$$VSS = 127\,877.5 - 115\,477.5 = 12\,400 \quad (50)$$

implying a loss of 10% caused by the need to restart some units at the second stage.

So let us see why the deterministic solution is bad. We answer by means of the following tests:

Test B We follow the skeleton solution from the deterministic model closing units of type 2, not required to satisfy the deterministic demand of 300 mw. The model reacts by opening units of type 2 at the second stage at higher cost. The associated expected skeleton solution value $ESSV = EEV$ and $LUSS = VSS$ means that the deterministic solution has a bad structure because it closes units required in the stochastic environment.

As before from an algorithmic perspective, we still solve a mixed integer stochastic linear program but with a smaller dimension than the original one.

Test C We check the upgradeability of the number of operating units ($u_i^1 \geq \bar{u}_i^1$) allowing the stochastic model to decide on the output rates and possibly new units. Notice that for

both types of units the constraint is automatically satisfied; the number of units opened in the deterministic case can then be considered as a lower bound for the stochastic one with $LUDS = 0$ (a case of perfect upgradeability).

In conclusion the deterministic solution is bad because it tends to follow in every period the market profile, by closing units that could be needed in the following time period. However the deterministic solution gives us a lower bound on the number of units to open in the stochastic context.

4 Conclusions

In this paper we have analyzed the *quality* of the expected value solution in terms of its structure and upgradeability to the stochastic solution. A qualitative understanding of the deterministic solution can be very useful both in case of untractable real-world problems or for problems actually solvable but that should be run very often. Measures of partial information from the expected value solution, such as the quality of its structure (*loss using skeleton solution, LUSS*) and upgradeability to the stochastic solution (*loss of upgrading the deterministic solution, LUDS*) have been defined and related to the standard *value of the stochastic solution VSS*. *LUSS* and *LUDS*, here computed on different small case studies can help us to understand the behavior of the deterministic solution and the reasons of its badness/goodness. In conclusion, by means of the tests proposed, we can identify the main causes of badness/goodness of the expected value solution as follows:

- the 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* $0 < LUSS \leq VSS$.
- the 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.
- the non-upgradeability of the deterministic solution to the stochastic measured by a positive *loss of upgrading the deterministic solution* $LUDS > 0$. Situations where the deterministic solution is bad, but it is upgradable, is of great importance in many cases.

Acknowledgements

The first author would like to thank the Department of Management Science at Lancaster University where this work has been done, for the kind hospitality. The financial support from research grants 2008-2009-2010 of the Department of Mathematics, Statistics, Computer

Science and Applications, University of Bergamo, responsible Prof. Marida Bertocchi is also acknowledged.

References

- [1] Ariyawansa, K.A., and Zhu, Y., A preliminary set of applications leading to stochastic semidefinite programs and chance-constrained semidefinite programs, *Mathematical and Computer Modeling*, (under review).
- [2] Birge J.R., The value of the stochastic solution in stochastic linear programs with fixed recourse, *Math. Prog.*, Vol. 24, pp. 314-325 (1982).
- [3] Birge, J.R., and Louveaux, F., *Introduction to stochastic programming*, Springer-Verlag, New York (1997).
- [4] Dantzig, G.B., Linear programming under uncertainty, *Management Science*, Vol. 1, pp. 197–206 (1995).
- [5] Garver, L.L., Power generation scheduling by integer programming-Development of Theory, *IEEE Trans. Power Apparatus and Systems*, No. 81, pp. 730–735 (1963).
- [6] Higle, J.L. and Wallace, S.W., Sensitivity analysis and uncertainty in linear programming, *Interfaces INFORMS*, Vol. 33, No. 4, pp. 53-60 (2003).
- [7] Kall, P. and Wallace, S.W., *Stochastic Programming*. Wiley, Chichester (1994).
- [8] Ko, Y-B., and Vaidya, N.H., Location-Aided Routing (LAR) in mobile ad hoc networks, *Wireless Network*, Vol. 6, No. 4, pp. 307-321 (2000).
- [9] Maggioni, F., Potra, F., Bertocchi, M. and Allevi, E., Stochastic second-order cone programming in mobile ad hoc networks, *Journal of Optimization, theory and applications*, Vol. 143, pp. 309-328 (2009).
- [10] Maggioni, F., Wallace, S.W., Bertocchi, M. and Allevi, E., Sensitivity analysis in stochastic second-order cone programming for mobile ad hoc networks, preprint.
- [11] Maggioni, F., Kaut, M. and Bertazzi, L., Stochastic Optimization models for a single-sink transportation problem, *Comput. Manag. Sci*, Vol. 6, pp. 251-267 (2009).
- [12] Wallace S.W., Decision making under uncertainty: is sensitivity analysis of any use? *Operation Research*, Vol 48, No. 1, pp. 20-25 (2000).
- [13] Williams, H.P., *Model buiding in mathematical programming*, Wiley & Sons, ISBN 047190606 (1985).
- [14] Thapalia, B.K., Wallace, S.W., Kaut, M. and Crainic, T.G., Single Source Single-Commodity Stochastic Network Design, *Comput. Manag. Sci*, to appear.