Worst-case-expectation approach to optimization under uncertainty

Alexander Shapiro and Wajdi Tekaya
School of Industrial and Systems Engineering,
Georgia Institute of Technology,
Atlanta, Georgia 30332-0205, USA

Murilo Pereira Soares and Joari Paulo da Costa
ONS - Operador Nacional do Sistema Elétrico
Rua da Quitanda, 196, Centro
Rio de Janeiro, RJ, 20091-005, Brasil

Abstract. In this paper we discuss multistage programming with the data process subject to uncertainty. We consider a situation were the data process can be naturally separated into two components, one can be modeled as a random process, with a specified probability distribution, and the other one can be treated from a robust (worst case) point of view. We formulate this in a time consistent way and derive the corresponding dynamic programming equations. In order to solve the obtained multistage problem we develop a variant of the Stochastic Dual Dynamic Programming (SDDP) method. We give a general description of the algorithm and present computational studies related to planning of the Brazilian interconnected power system.

Key Words: multistage stochastic programming, robust optimization, robust distribution stochastic programming, time consistency, dynamic equations, Stochastic Dual Dynamic Programming, sample average approximation, risk neutral and risk averse approaches, case studies

1 Introduction

There are basically two popular approaches to optimization under uncertainty. One is the approach of robust optimization where one optimizes a worst possible case of a considered problem. The other approach, of stochastic programming, models uncertain parameters as random variables with a specified probability distribution and optimization is applied to the expected value of the objective function. We may refer to the recent books [1] and [7] where these approaches are discussed in details. Both approaches have advantages and disadvantages and can be applied in different situations. The robust approach could be too conservative especially in cases where uncertain parameters have a large range of variability. On the other hand, the stochastic programming approach makes sense if averaging of the objective function could be reasonably justified by applying the Law of Large Numbers to repeated operations.

There are also situations where the involved uncertain parameters can be naturally divided into two groups, for one group the robust approach makes sense while for the other the stochastic programming approach is more appropriate. In this paper we discuss how the robust and stochastic programming approaches can be combined together. We consider static (two-stage) and dynamic (multi-stage) problems.

This paper will be organized as follows. In the next section we introduce basic ideas in the framework of a static formulation. The main development will be given in section 3 where a dynamic (multistage) setting will be discussed. In section 4 we outline an approach based on the Stochastic Dual Dynamic Programming (SDDP) method (cf., [3]) to a numerical solution of such multistage programs. Finally in section 5 we discuss numerical experiments with this approach applied to Brazilian operation planning of hydro plants. In particular we compare this with a risk averse approach based on coherent risk measures (cf., [4],[6],[9]).

2 Static case

Consider a problem where an objective function $F(x, \xi)$, depending on vector $\xi \in \mathbb{R}^d$ of uncertain parameters, is optimized (say minimized) subject to feasibility constraints $x \in \mathcal{X} \subset \mathbb{R}^n$. In particular, in two stage modeling $F(x, \xi)$ can be the optimal value of second stage problem. Suppose further that vector ξ can be partitioned $\xi = (\xi^1, \xi^2)$ with ξ^1 varying in specified uncertainty set $\Xi^1 \subset \mathbb{R}^{d_1}$, and $\xi^2 \in \mathbb{R}^{d_2}$ modeled as a random vector with specified probability distribution. This leads to the following Robust - Stochastic Programming formulation of the corresponding optimization problem

$$\operatorname*{Min}_{x \in \mathcal{X}} \left\{ f(x) := \sup_{\xi^1 \in \Xi^1} \mathbb{E}\left[F(x, \xi^1, \xi^2)\right] \right\}. \tag{2.1}$$

For given x and ξ^1, the expectation in (2.1) is performed with respect to the probability distribution of random vector ξ^2. In this formulation one optimizes the worst case average of the objective.

It could be worthwhile to note that the above function $f(x)$ can be also written as

$$f(x) = \sup_{P \in \mathfrak{M}} \mathbb{E}_P\left[F(x, \xi)\right], \tag{2.2}$$

where $\mathfrak{M}$ is a set of probability measures P on $\mathbb{R}^d$ of the form $P = P_1 \times P_2$ with P_2 being the specified probability measure (distribution) of ξ^2 and P_1 being any probability measure supported on the set Ξ^1. The maximum in (2.2) is attained at a probability measure with P_1 being a measure of mass one at a point of Ξ^1. Therefore problem (2.1) can be viewed as a particular case of the robust distribution approach to stochastic programming.

Note that

$$\sup_{\xi^1\in\Xi^1} \mathbb{E}\left[F(x,\xi^1,\xi^2)\right] \le \mathbb{E}\left[\sup_{\xi^1\in\Xi^1} F(x,\xi^1,\xi^2)\right] \tag{2.3}$$

and the inequality can be strict. Therefore formulation (2.1) is not the same as optimizing expectation of the worst case objective function. For applications that we have in mind formulation (2.1) is relevant, we will discuss this later.

We make the following assumptions.

(A1) The function $F : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ is continuous.

(A2) For every point $(x, \xi^1) \in \mathbb{R}^n \times \mathbb{R}^{d_1}$ there exist a neighborhood V and integrable function $h(\xi^2)$ such that $|F(x,\xi^1,\xi^2)| \le h(\xi^2)$ for all $(x,\xi^1) \in V$ and a.e. ξ^2.

(A3) The set Ξ^1 is nonempty and compact.

(A4) The function $F(x,\xi)$ is convex in x.

It follows by (A1) and (A2) that the expectation function $\psi(x,\xi^1) := \mathbb{E}\left[F(x,\xi^1,\xi^2)\right]$ is well defined finite valued and continuous. Together with (A3) this implies that the function $f(x) = \sup_{\xi^1\in\Xi^1} \psi(x,\xi^1)$ is finite valued. Also by (A4) the function $f(x)$ is convex. In order to proceed with cutting plane type algorithms for solving problem (2.1) we need a procedure for computing subgradients of $f(x)$.

Since $f(x)$ is convex finite valued, it is subdifferentiable. By the Levin-Valadier theorem (cf., [2, p.213]) its subdifferential is

$$\partial f(x) = \operatorname{conv}\left\{\cup_{\xi^1\in\bar{\Xi}^1(x)} \partial\psi(x,\xi^1)\right\}, \tag{2.4}$$

where conv(A) denotes convex hull of set A, all subdifferentials are taken with respect to x and

$$\bar{\Xi}^1(x) := \arg\max_{\xi^1\in\Xi^1} \psi(x,\xi^1).$$

Also by Strassen theorem (cf., [10]) $\partial\psi(x,\xi^1) = \mathbb{E}\left[\partial F(x,\xi^1,\xi^2)\right]$. Thus, under assumptions (A1)-(A4),

$$\partial f(x) = \operatorname{conv}\left\{\cup_{\xi^1\in\bar{\Xi}^1(x)} \mathbb{E}\left[\partial F(x,\xi^1,\xi^2)\right]\right\}. \tag{2.5}$$

In particular, if the probability distribution of ξ^2 has finite support $\{\omega^1,...,\omega^K\}$ with respective probabilities $p_1,...,p_K$, then

$$\mathbb{E}\left[F(x,\xi^1,\xi^2)\right] = \sum_{k=1}^K p_k F(x,\xi^1,\omega^k) \ \text{ and } \ \partial\mathbb{E}\left[F(x,\xi^1,\xi^2)\right] = \sum_{k=1}^K p_k \partial F(x,\xi^1,\omega^k). \tag{2.6}$$

In order to apply these formulas in a numerical, cutting plane type, algorithm we need a procedure for computing maximum of $\mathbb{E}\left[F(x,\xi^1,\xi^2)\right]$ over $\xi^1 \in \Xi^1$. In general this could be an intractable problem. If the set Ξ^1 is finite of not too large cardinality, then of course this can be done in a straightforward way. We will discuss this further in the next section.

3 Multistage case

In this section we discuss an extension of the above approach to a multistage setting. For the sake of simplicity we consider *linear* multistage problems. Let $\xi_1, ..., \xi_T$ be the uncertainty process underlying the corresponding multistage problem with ξ_1 being known (deterministic). Suppose that the data vectors $\xi_t \in \mathbb{R}^{d_t}$, $t = 2, ..., T$, are decomposed into two parts. That is $\xi_t = (\xi_t^1, \xi_t^2)$, with $(\xi_2^1, ..., \xi_T^1) \in \Xi^1$ and $\xi_2^2, ..., \xi_T^2$ being a random process with a specified probability distribution. We refer to $\xi_2^1, ..., \xi_T^1$ as the *uncertain* parameters and to $\xi_2^2, ..., \xi_T^2$ as the *random* parameters of the model.

Consider the following worst-case-expectation formulation of the linear multistage program

$$\begin{array}{lll} \operatorname*{Min}\limits_{x_1, x_2(\cdot), ..., x_T(\cdot)} & \sup\limits_{(\xi_2^1, ..., \xi_T^1) \in \Xi^1} & \mathbb{E}\left\{c_1^\top x_1 + c_2^\top x_2(\xi_{[2]}) + ... + c_T^\top x_T(\xi_{[T]})\right\} \\ & \text{s.t.} & A_1 x_1 = b_1,\ x_1 \geq 0, \\ & & B_t x_{t-1}(\xi_{[t-1]}) + A_t x_t(\xi_{[t]}) = b_t,\ x_t(\xi_{[t]}) \geq 0,\ t = 2, ..., T, \end{array} \tag{3.1}$$

where $\xi_1 = (c_1, A_1, b_1)$, $\xi_t = (c_t, A_t, B_t, b_t)$, $t = 2, ..., T$, $\xi_{[t]} = (\xi_1, ..., \xi_t)$ denotes history of the process up to time t and the expectation is taken with respect to the probability distribution of the random process $\xi_2^2, ..., \xi_T^2$. The optimization is performed over functions (called policies or decision rules) $x_t(\xi_{[t]})$, $t = 1, ..., T$, of the history of the data process (this is emphasized in the notation $x_t(\cdot)$) and the feasibility constraints should be satisfied for all $(\xi_2^1, ..., \xi_T^1) \in \Xi^1$ and almost every (a.e.) realization of the random process. For the two stage case of $T = 2$, problem (3.1) becomes of the form (2.1) with the function $F(x, \xi^1, \xi^2)$ given by the optimal value of the corresponding second stage problem.

We have that

$$\begin{array}{l} \mathbb{E}\left\{c_1^\top x_1 + c_2^\top x_2(\xi_{[2]}) + c_3^\top x_3(\xi_{[3]}) + ... + c_T^\top x_T(\xi_{[T]})\right\} = \\ \quad c_1^\top x_1 + \mathbb{E}\Big\{c_2^\top x_2(\xi_{[2]}) + \left\{\mathbb{E}_{|\xi^2_{[2]}} c_3^\top x_3(\xi_{[3]}) + ... + \mathbb{E}_{|\xi^2_{[T-1]}}\left[c_T^\top x_T(\xi_{[T]})\right]\right\}\Big\}, \end{array}$$

where $\mathbb{E}_{|\xi^2_{[t]}}$ denotes the conditional expectation with respect to $\xi^2_{[t]}$. Thus

$$\begin{array}{ll} \sup\limits_{(\xi_2^1, ..., \xi_T^1) \in \Xi^1} & \mathbb{E}\left\{c_1^\top x_1 + c_2^\top x_2(\xi_{[2]}) + c_3^\top x_3(\xi_{[3]}) + ... + c_T^\top x_T(\xi_{[T]})\right\} \leq \\ & c_1^\top x_1 + \sup\limits_{(\xi_2^1, ..., \xi_T^1) \in \Xi^1} \mathbb{E}\Big\{c_2^\top x_2(\xi_{[2]}) + \sup\limits_{(\xi_2^1, ..., \xi_T^1) \in \Xi^1} \mathbb{E}_{|\xi^2_{[2]}}\left\{c_3^\top x_3(\xi_{[3]}) + ...\right. \\ & \quad \left. + \sup\limits_{(\xi_2^1, ..., \xi_T^1) \in \Xi^1} \mathbb{E}_{|\xi^2_{[T-1]}}\left[c_T^\top x_T(\xi_{[T]})\right]\right\}\Big\}. \end{array} \tag{3.2}$$

This leads to the following nested formulation of the worst-case-expectation multistage problem

$$\operatorname*{Min}_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \rho_{2|\xi_1}\left[\min_{\substack{B_2 x_1 + A_2 x_2 = b_2 \\ x_2 \geq 0}} c_2^\top x_2 + \cdots + \rho_{T|\xi_{[T-1]}}\left[\min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T\right]\right], \tag{3.3}$$

where

$$\rho_{t|\xi_{[t-1]}}[\,\cdot\,] = \sup_{(\xi_2^1, ..., \xi_T^1) \in \Xi^1} \mathbb{E}_{|\xi^2_{[t-1]}}[\,\cdot\,], \quad t = 2, ..., T. \tag{3.4}$$

Note that similar to the static case (discussed in the previous section) the maximum over $(\xi_2^1, ..., \xi_T^1) \in \Xi^1$ in the above formulas can be replaced by the corresponding maximum over all probability distributions supported on Ξ^1. Note also that the inequality in (3.2) can be strict, and the nested worst-case-expectation multistage problem (3.4) is not necessarily equivalent to the problem (3.1) (cf., [8]).

For the nested problem it is possible to write the following dynamic programming equations (cf., [5],[8])

$$Q_t(x_{t-1}, \xi^1_{[t]}, \xi^2_{[t]}) = \inf_{B_t x_{t-1} + A_t x_t = b_t,\ x_t \geq 0} \left\{ c_t^{\mathsf{T}} x_t + \mathcal{Q}_{t+1}(x_t, \xi^1_{[t]}, \xi^2_{[t]}) \right\}, \tag{3.5}$$

$t = 2, ..., T$, with $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and

$$\mathcal{Q}_{t+1}(x_t, \xi^1_{[t]}, \xi^2_{[t]}) = \sup_{(\xi_2^{1'}, ..., \xi_T^{1'}) \in \Xi^1} \mathbb{E}_{|\xi^2_{[t]}} \left\{ Q_{t+1}(x_t, \xi^{1'}_{[t+1]}, \xi^2_{[t+1]}) : \xi^{1'}_{[t]} = \xi^1_{[t]} \right\}. \tag{3.6}$$

There are two basic questions about this model – how to choose the uncertainty set Ξ^1 and how to solve the obtained problem. These two questions are related, on one hand choice of Ξ^1 should make practical sense, on the other hand the corresponding problem (3.1) should be amendable to a numerical solution. Suppose that the data process $\xi_2, ..., \xi_T$ is *stagewise independent*, that is random vector ξ_t^2 is independent of $(\xi_2^2, ..., \xi_{t-1}^2)$, $t = 3, ..., T$, and the uncertainty set $\Xi^1 = \Xi_2^1 \times \cdots \times \Xi_T^1$ is given by a direct product of individual uncertainty sets. If the uncertain parameters have reasonably small variations, one can construct the respective uncertainty sets by introducing small variations of the parameters around their nominal values. In a sense this approach can be considered as a study of sensitivity of the computed solutions to small variations of the uncertain parameters.

In the above case of stagewise independence the dynamic equations (3.5)–(3.6) simplify to

$$Q_t(x_{t-1}, \xi^1_t, \xi^2_t) = \inf_{B_t x_{t-1} + A_t x_t = b_t,\ x_t \geq 0} \left\{ c_t^{\mathsf{T}} x_t + \mathcal{Q}_{t+1}(x_t) \right\}, \tag{3.7}$$

$t = 2, ..., T$, with $\mathcal{Q}_{T+1}(\cdot) \equiv 0$ and

$$\mathcal{Q}_{t+1}(x_t) = \sup_{\xi^1_{t+1} \in \Xi^1_{t+1}} \mathbb{E} \left\{ Q_{t+1}(x_t, \xi^1_{t+1}, \xi^2_{t+1}) \right\}, \tag{3.8}$$

where the expectation is taken with respect to ξ^2_{t+1}. Note that under the above assumptions of stagewise independence the cost-to-go functions $\mathcal{Q}_{t+1}(x_t)$ do not depend on the data process. Note also that functions $Q_t(x_{t-1}, \xi^1_t, \xi^2_t)$ are convex in x_{t-1}, and hence functions $\mathcal{Q}_t(x_{t-1})$ are convex. We assume that the cost-to-go functions $\mathcal{Q}_t(x_{t-1})$ are *real valued* and the sets Ξ_t are *compact*.

4 Stochastic Dual Dynamic Programming method

In this section we discuss application of the Stochastic Dual Dynamic Programming (SDDP) method to the nested formulation (3.3). We assume the stagewise independence condition, hence the dynamic equations take the form (3.7)–(3.8). In order to apply the SDDP algorithm we need to compute subgradients of the cost-to-go functions $\mathcal{Q}_{t+1}(x_t)$. Consider function

$$q_t(x_{t-1}, \xi^1_t) = \mathbb{E} \left\{ Q_t(x_{t-1}, \xi^1_t, \xi^2_t) \right\}, \quad t = 2, ..., T,$$

where the expectation is taken with respect to the distribution of ξ^2_t. This function is convex in x_{t-1} and, as it was assumed, is finite valued and hence by the Strassen Theorem

$$\partial q_t(x_{t-1}, \xi^1_t) = \mathbb{E} \left\{ \partial Q_t(x_{t-1}, \xi^1_t, \xi^2_t) \right\}, \tag{4.1}$$

where all subgradients are taken with respect to x_{t-1}. We have that

$$\mathcal{Q}_t(x_{t-1}) = \sup_{\xi_t^1 \in \Xi_t^1} q_t(x_{t-1}, \xi_t^1),$$

and by the Levin-Valadier Theorem

$$\partial \mathcal{Q}_t(x_{t-1}) = \text{conv}\left\{ \cup_{\xi_t^1 \in \bar{\Xi}_t^1(x_{t-1})} \partial q_t(x_{t-1}, \xi_t^1) \right\}, \tag{4.2}$$

where

$$\bar{\Xi}_t^1(x_{t-1}) := \arg\max_{\xi_t^1 \in \Xi_t^1} q_t(x_{t-1}, \xi_t^1).$$

Therefore a subgradient of $\mathcal{Q}_{t+1}(x_t)$ is given by

$$\nabla \mathcal{Q}_t(x_{t-1}) = \mathbb{E}\left\{ \nabla Q_t(x_{t-1}, \bar{\xi}_t^1, \xi_t^2) \right\}, \tag{4.3}$$

for some $\bar{\xi}_t^1 \in \bar{\Xi}_t^1(x_{t-1})$ (as before all subgradients are taken with respect to x_{t-1}).

The first step of numerical solution is to discretize the data process. The random process ξ_t^2, $t = 2, ..., T$, is discretized by generating random sample from probability distribution of each random vector; this is the approach of the Sample Average Approximation (SAA) method. Discretization of the uncertainty sets Ξ_t^1 is a more subtle issue.

4.1 Backward step of the SDDP algorithm

Suppose for the sake of simplicity that only the right hand sides b_t, $t = 2, ..., T$, are uncertain. That is, $\xi_t = b_t$ and $\xi_t^1 = b_t^1$, $\xi_t^2 = b_t^2$, $b_t = (b_t^1, b_t^2)$. Let b_{tj}^2, $j = 1, ..., N$, be the respective random sample, $t = 2, ..., T$. Let $\bar{x}_t$, $t = 1, ..., T-1$, be trial points (we can use more than one trial point at every stage of the backward step, an extension to that will be straightforward). Let $\mathcal{Q}_t(\cdot)$ be the cost-to-go functions of dynamic programming equations associated with the considered multistage problem, and $\mathfrak{Q}_t(\cdot)$ be a current approximation of $\mathcal{Q}_t(\cdot)$ given by the maximum of a collection of *cutting* planes

$$\mathfrak{Q}_t(x_{t-1}) = \max_{k \in \mathcal{I}_t} \left\{ \alpha_{tk} + \beta_{tk}^{\mathsf{T}} x_{t-1} \right\}, \; t = 1, ..., T-1. \tag{4.4}$$

For a given b_T^1 and $b_{Tj} = (b_T^1, b_{Tj}^2)$ at stage $t = T$ we solve N problems

$$\operatorname*{Min}_{x_T \in \mathbb{R}^{n_T}} c_T^{\mathsf{T}} x_T \;\text{ s.t. } B_T \bar{x}_{T-1} + A_T x_T = b_{Tj}, \; x_T \geq 0, \; j = 1, ..., N. \tag{4.5}$$

The optimal value of problem (4.5) is $Q_{Tj}(\bar{x}_{T-1}, b_T^1)$. We should compute value of the cost-to-go function $\mathcal{Q}_T(x_{T-1})$ at trial point $\bar{x}_{T-1}$. We have

$$\mathcal{Q}_T(\bar{x}_{T-1}) = \sup_{b_T^1 \in \Xi_T^1} q_T(\bar{x}_{T-1}, b_T^1), \tag{4.6}$$

where

$$q_T(\bar{x}_{T-1}, b_T^1) = N^{-1} \sum_{j=1}^{N} Q_{Tj}(\bar{x}_{T-1}, b_T^1). \tag{4.7}$$

Now going one stage back $Q_{T-1,j}(\bar{x}_{T-2}, b_{T-1}^1)$ is equal to the optimal value of problem

$$\operatorname*{Min}_{x_{T-1} \in \mathbb{R}^{n_{T-1}}} c_{T-1}^{\mathsf{T}} x_{T-1} + \mathcal{Q}_T(x_{T-1}) \;\text{ s.t. } B_{T-1} \bar{x}_{T-2} + A_{T-1} x_{T-1} = b_{T-1,j}, \; x_{T-1} \geq 0. \tag{4.8}$$

However, function $\mathcal{Q}_T(\cdot)$ is not available. Therefore we replace it by $\mathfrak{Q}_T(\cdot)$ and hence consider problem

$$\operatorname*{Min}_{x_{T-1}\in\mathbb{R}^{n_{T-1}}} c_{T-1}^{\mathsf{T}} x_{T-1} + \mathfrak{Q}_T(x_{T-1}) \text{ s.t. } B_{T-1}\bar{x}_{T-2} + A_{T-1}x_{T-1} = b_{T-1,j},\ x_{T-1} \geq 0. \tag{4.9}$$

Recall that $\mathfrak{Q}_T(\cdot)$ is given by maximum of affine functions (see (4.4)). Therefore we can write problem (4.4) in the form

$$\begin{array}{ll} \operatorname*{Min}_{x_{T-1}\in\mathbb{R}^{n_{T-1}},\theta\in\mathbb{R}} & c_{T-1}^{\mathsf{T}} x_{T-1} + \theta \\ \text{s.t.} & B_{T-1}\bar{x}_{T-2} + A_{T-1}x_{T-1} = b_{T-1,j},\ x_{T-1} \geq 0 \\ & \theta \geq \alpha_{Tk} + \beta_{Tk}^{\mathsf{T}} x_{T-1},\ k \in \mathcal{I}_T. \end{array} \tag{4.10}$$

Then we have to solve the problem

$$\max_{b_{T-1}^1 \in \Xi_{T-1}^1} q_{T-1}(\bar{x}_{T-2}, b_{T-1}^1), \tag{4.11}$$

where

$$q_{T-1}(\bar{x}_{T-2}, b_{T-1}^1) = N^{-1}\sum_{j=1}^{N} Q_{T-1,j}(\bar{x}_{T-2}, b_{T-1}^1),$$

and so on.

Let us consider the following approach. Suppose that we can sample from sets Ξ_t^1. For example if sets Ξ_t^1 are finite, probably with large cardinality, we can sample an element of Ξ_t^1 with equal probability. Consider the cost-to-go function $\mathcal{Q}_T(x_{T-1})$. Sample L points $b_{T\ell}^1$, $\ell = 1,...,L$, from Ξ_T^1. The number L can be small, even $L = 1$. For $\ell = 1,...,L$, compute subgradient (at $\bar{x}_{T-1}$)

$$\gamma_{T\ell} = N^{-1}\sum_{j=1}^{N} \nabla Q_{Tj}(\bar{x}_{T-1}, b_{T\ell}^1). \tag{4.12}$$

Note that by (4.3) the subgradient $\nabla Q_{Tj}(\bar{x}_{T-1}, b_{T\ell}^1)$ can be computed by solving the dual of problem (4.5) for $b_{Tj} = (b_{T\ell}^1, b_{Tj}^2)$. Add the corresponding cutting planes

$$q_T(\bar{x}_{T-1}, b_{T\ell}^1) + \gamma_{T\ell}^{\mathsf{T}}(x_{T-1} - \bar{x}_{T-1}),\ \ell = 1,...,L,$$

to the collection of cutting planes of $\mathfrak{Q}_T(\cdot)$. By (4.6) we have that $\mathcal{Q}_T(\bar{x}_{T-1}) \geq q_T(\bar{x}_{T-1}, b_T^1)$ for any $b_T^1 \in \Xi_T^1$, and hence

$$\mathcal{Q}_T(x_{T-1}) \geq q_T(\bar{x}_{T-1}, b_{T\ell}^1) + \gamma_{T\ell}^{\mathsf{T}}(x_{T-1} - \bar{x}_{T-1}), \tag{4.13}$$

i.e., $q_T(\bar{x}_{T-1}, b_{T\ell}^1) + \gamma_{T\ell}^{\mathsf{T}}(x_{T-1} - \bar{x}_{T-1})$ is indeed a cutting plane for $\mathfrak{Q}_T(\cdot)$.

And so on for stages $t = T-1,...,2$.

4.2 Forward step of the SDDP algorithm

The forward step of the algorithm is done in the standard way. The computed approximations $\mathfrak{Q}_2(\cdot),...,\mathfrak{Q}_T(\cdot)$ (with $\mathfrak{Q}_{T+1}(\cdot) \equiv 0$ by definition) and a feasible first stage solution $\bar{x}_1$ can be used for constructing an implementable policy as follows. For a realization $\xi_t = (c_t, A_t, B_t, b_t)$, $t = 2,...,T$, of the data process, decisions $\bar{x}_t$, $t = 1, ..., T$, are computed recursively going forward with $\bar{x}_1$ being the chosen feasible solution of the first stage problem, and $\bar{x}_t$ being an optimal solution of

$$\operatorname*{Min}_{x_t} c_t^\top x_t + \mathfrak{Q}_{t+1}(x_t) \ \text{ s.t. } A_t x_t = b_t - B_t \bar{x}_{t-1},\ x_t \geq 0, \tag{4.14}$$

for $t = 2, ..., T$. These optimal solutions can be used as trial decisions in the backward step of the algorithm.

The forward step of the standard SDDP algorithm has two purposes: (i) to generate trial points, and (ii) to estimate value of the constructed policy and hence to provide an upper bound for the optimal value of the considered problem. Unfortunately the second function of the forward step cannot be reproduced in the present case.

4.3 Redundant cutting planes elimination

Typically, a significant number of cutting planes added by the SDDP method in the backward step become at some point not necessary for the description of the cost-to-go functions approximations and could be eliminated. In this section we present a subroutine that identifies these redundant cutting planes. This subroutine allows a significant speed up of any SDDP type algorithm in general while preserving the statistical properties of the constructed policy.

First, we start by presenting the problem setting. At each stage, the cost-to-go function of dynamic programming equations are approximated by $\mathfrak{Q}(\cdot)$ given by the maximum of a collection of cutting planes in the following manner:

$$\mathfrak{Q}(x) = \max_{k \in \mathcal{I}} \left\{ \alpha_k + \beta_k^\top x \right\} \tag{4.15}$$

for $x \in \Gamma$ where Γ is a compact set.
An example of a redundant cutting plane is illustrated in Figure 1. In this figure, we assume that all the hyperplanes define half spaces in the non negative orthant and $\Gamma = [0, 4]$.

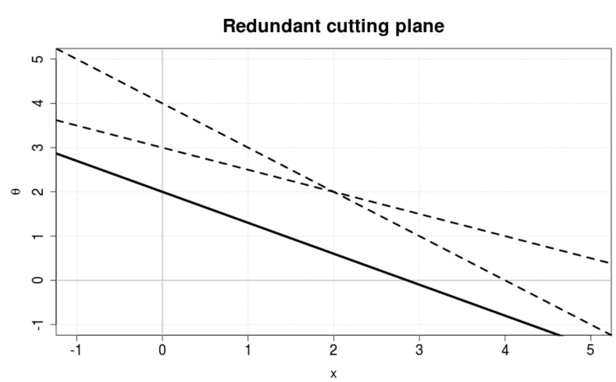


Figure 1: Illustration of a redundant cutting planes

The cutting plane in bold line is redundant since it can never be active in describing $\mathfrak{Q}(.)$ over Γ. Thus, it can be safely discarded. Empirical evidence (cf. section 5.3) shows that the SDDP method tends to generate a significant number of such cutting planes.

Without loss of generality, checking if $\alpha_1 + \beta_1^\mathsf{T} x$ is redundant is equivalent to check the feasibility of the following linear system:

$$\begin{cases} \theta \leq \alpha_1 + \beta_1^\mathsf{T} x \\ \theta \geq \alpha_k + \beta_k^\mathsf{T} x, \forall k \in \mathcal{I} \setminus \{1\} \\ x \in \Gamma \end{cases} \tag{4.16}$$

where (θ, x) are variables and $(\alpha_k, \beta_k)_{k \in \mathcal{I}}$ are known data.
If problem (4.16) is infeasible, then $\alpha_1 + \beta_1^\mathsf{T} x$ is redundant (as illustrated in Figure 1) and could be removed. Otherwise, the constraint must be maintained to preserve the statistical properties of the constructed policy.

Algorithmic description of the redundant cutting planes elimination procedure is described in Algorithm 1.

Algorithm 1 Redundant cutting planes elimination procedure

Require: $\mathfrak{Q}(x) = \max_{k \in \mathcal{I}} \left\{ \alpha_k + \beta_k^\mathsf{T} x \right\}$

1: **for** $j \in \mathcal{I}$ **do**
2: Check feasibility of the polyhedron $\mathbf{P} = \left\{ (\theta, x) : \begin{array}{l} \theta \leq \alpha_j + \beta_j^\mathsf{T} x \\ \theta \geq \alpha_k + \beta_k^\mathsf{T} x, \forall k \in \mathcal{I} \setminus \{j\} \\ x \in \Gamma \end{array} \right\}$
3: **if** $\mathbf{P} = \emptyset$ **then**
4: Discard $(\alpha_j + \beta_j^\mathsf{T} x)$ from $\mathbf{P}$
5: **end if**
6: **end for**

The use of this procedure within the SDDP algorithm and the achieved performance gains are discussed in section 5.3.

4.4 Sampling from the uncertainty set

Following the general methodology of robust optimization (see [1]), suppose that the uncertainty set Ξ_t^1, $t = 2, ..., T$, is an ellipsoid, centered at a point $\bar{\xi}$. That is,

$$\Xi_t^1 := \{\xi : (\xi - \bar{\xi})^\mathsf{T} A (\xi - \bar{\xi}) \leq r\}, \tag{4.17}$$

where A is a positive definite matrix and $r > 0$. Consider the set of points of Ξ_t^1 which are not dominated by other points of Ξ_t^1,

$$D := \left\{ \xi \in \Xi_t^1 : \text{does not exist } \xi' \in \Xi_t^1 \text{ such that } \xi' \neq \xi \text{ and } \xi \leq \xi' \right\}.$$

In applications which we have in mind it makes sense to restrict our sampling to the set $D \subset \Xi_t^1$, i.e., in fact we consider the uncertainty set given by boundary points of the ellipsoid Ξ_t^1 which are "larger" than the reference value $\bar{\xi}$.

The set D can be represented in the following form

$$D = \bigcup_{a \geq 0, \, \|a\|=1} \arg\max \left\{ a^\mathsf{T} \xi : (\xi - \bar{\xi})^\mathsf{T} A (\xi - \bar{\xi}) \leq r \right\}.$$

The maximizer of $a^\mathsf{T} \xi$, subject to $(\xi - \bar{\xi})^\mathsf{T} A (\xi - \bar{\xi}) \leq r$, is obtained by writing the optimality condition

$$a - \lambda A (\xi - \bar{\xi}) = 0,$$

with $\lambda > 0$. Hence such maximizer is given by $\xi^* = \lambda^{-1}A^{-1}a + \bar{\xi}$.

We can generate the required sample $\xi_1, ...,$ as follows. Generate $Z_i \sim N(0, I)$, where I is the identity matrix of an appropriate dimension. Take $\xi_i = cA^{-1}|Z_i| + \bar{\xi}$, where the absolute value $|Z_i|$ of vector Z_i is taken componentwise, and c is the calibration constant such that

$$c^2(A^{-1}|Z_i|)^{\mathsf{T}} A(A^{-1}|Z_i|) = r,$$

i.e., $c = \sqrt{r/\big(|Z_i|^{\mathsf{T}} A^{-1}|Z_i|\big)}$.

4.5 Algorithm description

In this section, we present an algorithmic description for the worst-case-expectation SDDP algorithm with one trial point per iteration.

Algorithm 2 Worst-case-expectation SDDP algorithm

Require: $\{\mathfrak{Q}_t^0\}_{t=2,...,T+1}$(Init. Lower approx.) and i_{max}(max. iterations)

Initialize: $i \leftarrow 0$, $\underline{z} = -\infty$ (Lower bound)
while $i < i_{max}$ **do**
  Sample 1 scenario: $\{c_t, A_t, B_t, (\bar{b}_t^1; b_t^2)\}_{2\le t\le T}$

  (Forward step)
  for $t = 1 \to T$ **do**
    $\bar{x}_t \leftarrow \arg\min_{x_t\in\mathbb{R}^{n_t}} \{c_t^\top x_t + \mathfrak{Q}_{t+1}^i(x_t) : B_t x_{t-1} + A_t x_t = (\bar{b}_t^1; b_t^2), x_t \ge 0\}$
  end for

  (Backward step)
  for $t = T \to 2$ **do**
    for $j = 1 \to N_t$ **do**
      $\left[\widetilde{Q}_{tj}(\bar{x}_{t-1}, \bar{b}_t^1), \widetilde{\pi}_{tj}\right] \leftarrow \min_{x_t\in\mathbb{R}^{n_t}} \{c_t^\top x_t + \mathfrak{Q}_{t+1}^i(x_t) : B_t\bar{x}_{t-1} + A_t x_t = (\bar{b}_t^1; b_{tj}^2), x_t \ge 0\}$
    end for
    $\widetilde{\mathcal{Q}}_t(\bar{x}_{t-1}, \bar{b}_t^1) := \frac{1}{N_t}\sum_{j=1}^{N_t} \widetilde{Q}_{t,j}(\bar{x}_{t-1}, \bar{b}_t^1)$; $\tilde{g}_t := -\frac{1}{N_t}\sum_{j=1}^{N_t} \tilde{\pi}_{tj} B_t$
    $\mathfrak{Q}_t^{i+1} \leftarrow \{x_{t-1} \in \mathfrak{Q}_t^i : -\tilde{g}_t x_{t-1} \ge \widetilde{\mathcal{Q}}_t(\bar{x}_{t-1}, \bar{b}_t^1) - \tilde{g}_t\bar{x}_{t-1}\}$
  end for

  (Lower bound update)
  $\underline{z} \leftarrow \min_{x_1\in\mathbb{R}^{n_1}} \{c_1^\top x_1 + \mathfrak{Q}_2(x_1) : A_1 x_1 = b_1, x_1 \ge 0\}$

  $i \leftarrow i + 1$
end while

4.6 Discussion of the SDDP algorithm

The numerical approach outlined above proceeds in several steps. First the sequence $\xi_2^2, ..., \xi_T^2$ of random parameters is discretized by generating a random sample of size N from each random vector ξ_t^2, $t = 2, ..., T$. Consequently the expectation in the right hand side of (3.8) is approximated by the finite sum (4.7). Once the sample is constructed, we are trying to solve the obtained SAA problem and this sample is not changed in the backward steps of the SDDP algorithm. Convergence properties of the Sample Average Approximations are well documented (e.g., [7, section 5.8]).

Still we have the problem of computing maximum of the corresponding cost-to-go functions with respect to the uncertain parameters. This forces us to do additional sampling from the uncertainty sets in the backward steps of the algorithm. As the number of iterations increases, the collections of generated points will fill the uncertainty sets and in the limit we will reconstruct the corresponding maxima. However, the convergence can be very slow in high dimensional cases. The number of points needed to approximate say a unit ball in $\mathbb{R}^d$ in a uniform way grows exponentially with increase of the dimension d. Therefore this approach could be reasonable when the dimensions d_t^1 are small. Fortunately in the motivating example, discussed in section 5, we have that $d_t^1 = 4$, $t = 2, ..., T$, and moreover, since we are sampling from the boundary of the corresponding ellipsoids, this effectively reduces the dimensions to 3. Numerical experiments reported in section 5 indicate that in the considered case this approach works reasonably well.

5 Motivation and Computational experiments

5.1 Motivation

In stochastic problems like the operation planning of power systems there are many kinds of uncertainties. Some uncertainties are related to nature process, such as the inflows to hydro power plants. Other uncertainties are related to economic uncertainties, such as future thermal plants fuel prices and demand values. In Brazilian power operation planning only the uncertainty of inflows to hydro plants is considered during the solution of the problem, because these are the variables with the larger impact on problem's solution. Moreover, to consider other uncertainties, such as the ones associated with the demand, in the same standard SDDP framework would result in an increase of the state space dimension and eventually hinder the numerical solution of the problem.

The Brazilian hydro power operation planning problem considers a planning horizon of 5 years. The standard approach to solve this problem is to resort to a chain of models that considers long, mid and short term planning horizon.

For the monthly operation planning procedure (mid term planning) the SDDP algorithm is used to estimate the cost-to-go functions for five years ahead. This procedure is repeated every month with updated data and the cost-to-go functions are re-estimated. Some data used during this operation planning are deterministic, or even uncertain but with very small uncertainty, such as the system's installed capacity at a given stage. On the other hand, some data, like the demand, has greater uncertainty, but are considered deterministic based on an up to five years forecast. For instance, Figure 2 shows the historical demand values and the forecast values that were used by the operation planning in several optimization problems. In this picture it is clear the effect of the international economic crisis on the demand values by the end of the year 2008, and its effect on the demand forecast, which was revised on May, 2009.

The forecast errors from January, 2008 to July, 2012 are shown in Figure 3 for Brazilian South-East system. These errors were computed considering the forecast demand for the first year of each optimization problem. As the economic crisis had an impact greater than expected on the demand, there is clearly a positive bias on the errors, specially when the impact of international crisis on Brazilian economy was yet unknown. In cases which there is a greater economic growth than considered on demand forecast, or even higher than expected temperatures, a negative bias is expected on the forecast errors. Although this last behaviours are not frequent in recent history, these are the cases which the methodology proposed on this article deals with, that is, to calculate a policy that is robust to demand values higher than expected. The importance of such robust policies is that, in case the demand happens to be greater than the expected for several months, it can prevent load curtailments that would happen if only an expected policy were used.

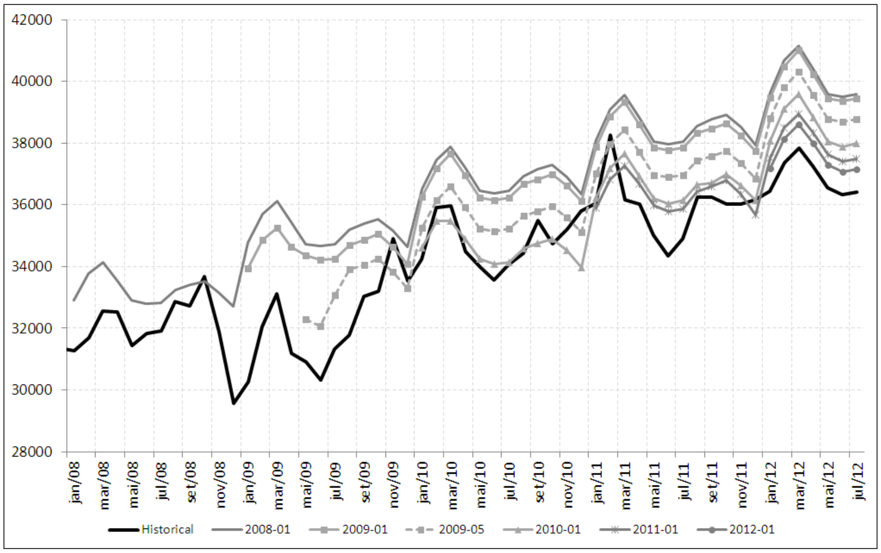


Figure 2: Historical and forecast demand values for Brazilian South-East system

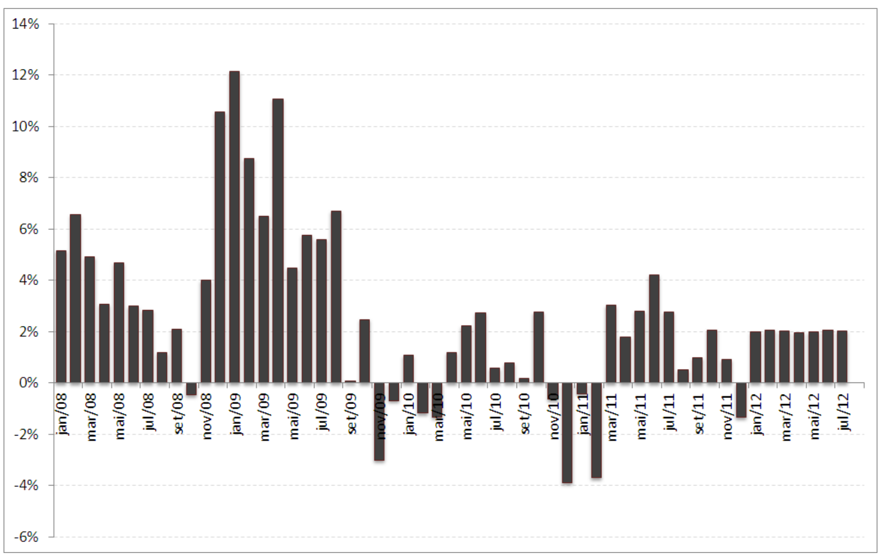


Figure 3: First year forecast errors for Brazilian South-East system

5.2 Case Studies Description

To investigate the effect of considering the worst-case-expectation approach proposed on this article the worst-case-expectation SDDP algorithm was implemented in C++ using Soplex 1.6 linear programming solver and applied to solve the problem. The codes were run on 1 core of a quad-core Intel Xeon E7-4870 @ 2.40GHz and 16GB RAM machine.

The numerical experiments were carried out considering instances of multi-stage linear stochastic problems based on an aggregate representation of the Brazilian Interconnected Power System long-term operation planning problem, as of January 2012. A more detailed description of the Brazilian Interconnected System can be found in [9].

	% of total load curtailment	Cost
1	0 – 5	1206.38
2	5 – 10	2602.56
3	10 – 20	5439.12
4	20 – 100	6180.26

Table 1: Deficit costs and depths

		to				
		SE	S	NE	N	IM
from	SE	–	7700	1000	0	4000
	S	5670	–	0	0	0
	NE	600	0	–	0	3000
	N	0	0	0	–	–
	IM	2854	0	3960	3149	–

Table 2: Interconnection limits between systems

The load of each area must be supplied by local hydro and thermal plants or by power flows among the interconnected areas. A slack thermal generator with high cost that increases with the amount of load curtailment accounts for load shortage at each area (Table 1). Interconnection limits between areas may differ depending of the flow direction, see Table 2. The energy balance equation for each sub-system has to be satisfied for each stage and scenario. There are bounds on stored and generated energy for each sub-system aggregate reservoir and on thermal generations.

The long-term planning horizon for the Brazilian case comprises 60 months, due to the existence of multi-year regulation capacity of some large reservoirs. In order to obtain a reasonable cost-to-go function that represents the continuity of the energy supply after these firsts 60 stages, a common practice is to add 60 more stages to the problem and consider a zero cost-to-go function at the end of the 120^{th} stage. Hence, the objective function of the planning problem is to minimize the expected cost of the operation along the 120 months planning horizon, while supplying the area loads and obeying technical constraints. The total cost is the sum of thermal generating costs plus a penalty term that reflects energy shortage.

The case's general data, such as hydro and thermal plants data and interconnections capacities were taken as static values through time. The demand for each system and the energy inflows in each reservoir were taken as time varying.

The SAA tree generated in case studies has 100 realizations per stage with the total number of scenarios $1 \times 100 \times \cdots \times 100 = 100^{119}$. In the following experiments we run the SDDP algorithm with 1 trial solution per iteration for 3000 iterations. The individual stage costs and policy value are evaluated using 2000 randomly generated scenarios using the same time series model of the optimization process in forward simulations. During this solution evaluation process six demand values were used: the original forecast demand and five increased demand cases (1%, 2%, 3%, 4% and 5% of increase in the forecast value).

The uncertainty sets, discussed in section 4.4, are defined by the following parameters for $t \geq 2$:

$$\begin{cases} r_t & = \left(\|\bar{\xi}_t^1\|_2 \times u\right)^2 \\ A & = \Sigma^{-1} \end{cases}$$

where Σ denotes the demand correlation matrix for the 4 systems estimated using historical demand data (60 observations), u, the uncertainty parameter, denotes the percentual increment on the demand and $\bar{\xi}_t^1$ denotes the demand forecast at stage t.

5.3 Redundant cutting planes elimination

In this section, we discuss the use of the redundant cutting planes elimination subroutine described in Algorithm 1 within the general framework of SDDP type methods.

First, we investigate the question of how frequently should this subroutine be used. We run 3000 iterations of the SDDP method with 1 trial point per iteration on the risk neutral case and the worst-case-expectation case for $u = 3\%$. We run several experiments where we use different constant cycle lengths (i.e. a cycle length of 50 means that we run the subroutine each 50 iterations). The ∞ denotes the case where we don't use the subroutine.

Cycle length	SDDP run time (dd:hh:mm)	subroutine run time (dd:hh:mm)	Total CPU time (dd:hh:mm)
50	00:09:59	00:04:10	00:14:09
100	00:10:18	00:02:06	00:12:24
200	00:11:21	00:01:06	00:12:27
400	00:13:19	00:00:43	00:14:02
∞	01:04:54	-	01:04:54

Table 3: CPU time summary for risk neutral SDDP

Cycle length	SDDP run time (dd:hh:mm)	subroutine run time (dd:hh:mm)	Total CPU time (dd:hh:mm)
50	00:10:56	00:04:23	00:15:19
100	00:11:06	00:02:16	00:13:22
200	00:11:58	00:01:10	00:13:08
400	00:13:55	00:00:44	00:14:39
∞	01:02:48	-	01:02:48

Table 4: CPU time summary for worst-case-expectation SDDP

Over the 3000 iterations, a speed up factor of at least 2 times is recorded with a cycle length of 100 or 200 when compared to experiment without running the subroutine. It is clear that the use of the subroutine significantly improves the method performance. Clearly, a tradeoff between the time spent in removing redundant cutting planes and performing SDDP iterations has to be made. On the one hand, with a cycle length of 50, the lowest SDDP run time is obtained. However, a significant amount of time is spent in running the subroutine. On the other hand, with a cycle length of 400, the lowest time spent on running the subroutine is achieved. Nevertheless, a longer time to run the SDDP is recorded. Furthermore, subroutines runs become more expensive as the number of cutting planes increases. A better strategy consists in changing the cycle length throughout the experiment as function of how costly it is to run the subroutine compared to performing further SDDP iterations.

This subroutine allows to have some measure of the cutting planes efficiency at each stage. Figure 4 plots the percentage of redundant cutting planes compared to the total number of cuts added per stage for the worst-case-expectation approach with $u = 3\%$. Performing 3000 iterations with 1 trial point per iteration generates 3000 cutting planes at each stage. It can be seen that the proportion of redundant cutting planes is higher for earlier stages with more than 60% in the first 70 stages. This can be explained by the continuous refinement of the cost-to-go function approximations for the first stages. In addition, the lower error accumulation in the cost-to-go function approximation for later stages explains the lower proportion of redundant cutting planes.

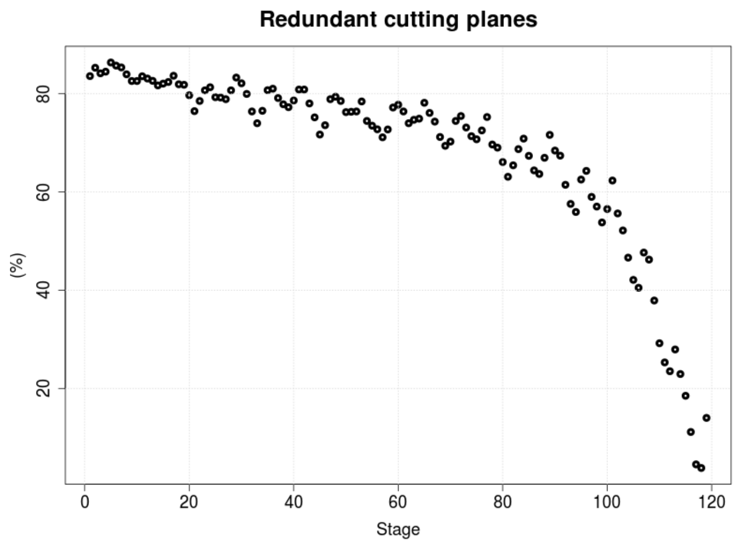


Figure 4: Redundant cutting planes proportion at each stage

5.4 Solution strategies

The first question that may arise before considering the worst-case-expectation approach is related to the number of points that should be sampled from the uncertainty set at each iteration and stage. To answer this question, three possibilities were investigated. The first one, which may seem to be natural, is to sample L demand values from the uncertainty set of stage t and add all calculated cuts to stage $t-1$. This approach requires that the backward computation is repeated L times at each iteration and, as L cuts are added at each iteration instead of one, it is expected a great increase on the CPU time with a small increase on the L value. On the other hand, it is also expected that the uncertainty set is better filled for higher L values. The second approach was to sample L demand values from the uncertainty set but, instead of adding all cuts to stage $t-1$, only the cut related to the worst expected value is added. This approach may reconstruct the maxima as well as the first one, but is going to require less CPU time. In the following experiments the uncertainty parameter $u = 1\%$, $L = 3$ and the described studies are called "Worst-case ($u = 1\%$, $L = 3$) all cuts" and "Worst-case ($u = 1\%$, $L = 3$) worst cut", respectively. The third approach was carried with $L = 1$, that is, sampling one value from the uncertainty set, and is called "Worst-case ($u = 1\%$)". This approach has no additional burden compared to risk neutral approach and may be preferred over the others, but it is unclear if the uncertainty set is going to be well described at the end of the computation. The main objective of the tests performed in this section is to address this issue.

The CPU time for the three strategies and the risk neutral case are shown on Table 5, and it is clear that with the increase of L value the worst-case approach becomes too time consuming. On the other hand, with $L = 1$ there was no additional time required to solve the problem. The next step is to compare the solutions obtained by each sampling approach.

Table 5: Total CPU time – Solution strategies

Case Study	dd:hh:mm:ss
Risk Neutral	01:06:57:40
$r = 1.00\%$ and $L = 1$	01:06:50:30
$r = 1.00\%$ and $L = 3$ worst cut	03:20:32:11
$r = 1.00\%$ and $L = 3$ all cuts	11:12:49:19

The total expected cost for 120 stages and its 95% confidence interval are shown on Table 6. The solution values obtained were very similar, which indicates that the $L = 1$ strategy may give as good results as the other approaches.

Table 6: Total expected values – 120 stages

Case Study	95% CI lower limit	Policy value mean	95% CI upper limit
$r = 1.00\%$ and $L = 1$	27,270,541,887.8	27,864,589,146.1	28,458,636,404.4
$r = 1.00\%$ and $L = 3$ worst cut	27,264,186,251.6	27,849,507,665.6	28,434,829,079.6
$r = 1.00\%$ and $L = 3$ all cuts	27,274,434,102.3	27,868,201,704.1	28,461,969,305.8

To ensure that "Worst-case ($u = 1\%$)" approach gives similar results as the other approaches, all variables related to the problem were compared, such as decision variables, marginal cost (dual variable) and the load curtailment frequencies, which is an index calculated after the problem solution to assess the security of the system's operation. This comparison shows that there was no significant difference on the solution of evaluated approaches. South-East system stored volumes and marginal costs average values, together with its 5% and 95% quantiles are shown on Figure 5 for a simulation with 101% of increase on the forecast demand.

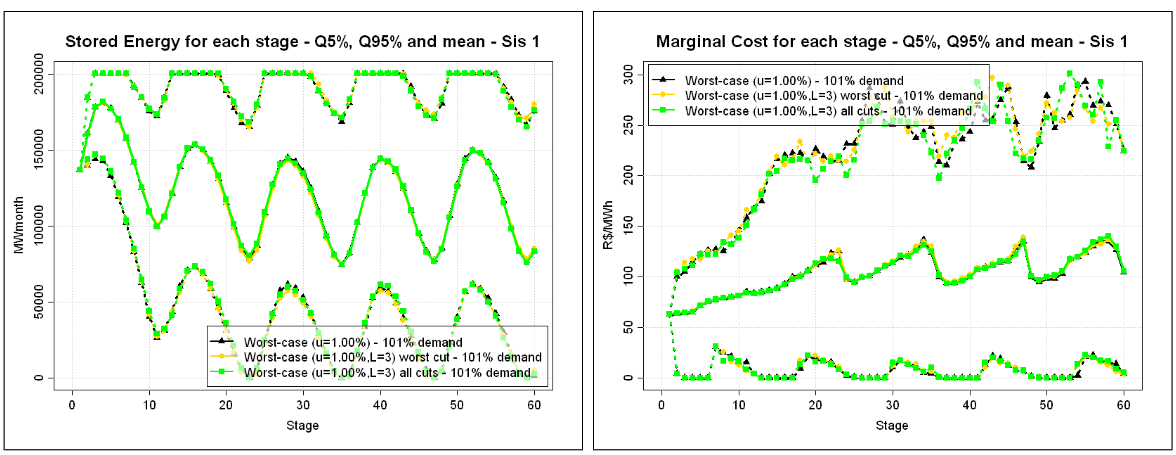


Figure 5: Stored volumes and marginal costs average, 5% and 95% quantiles values – South-East system

The next section comprises the numerical evaluation of the worst-case-expectation methodology itself. As the "Worst-case ($u = 1\%$)" showed good results, the $L = 1$ approach is going to be used on the further studies of this work.

5.5 Computational experiments

Given the approach with $L = 1$ discussed on Section 5.4, three case studies were run: risk neutral SDDP and worst-case-expectation SDDP with uncertainty factor $u = 1\%$ and $u = 3\%$.

In the left graph of Figure 6 we can see the average discounted total cost of 120 stages as a function of the demand increase for both risk neutral and worst-case-expectation approaches. As the demand increases, the average policy value increases for all the approaches. However, the rate of increase for the worst-case-expectation approaches is lower than for the risk neutral approach. The gain achieved by the worst-case-expectation approach related to the risk neutral is shown on the right side of Figure 6.

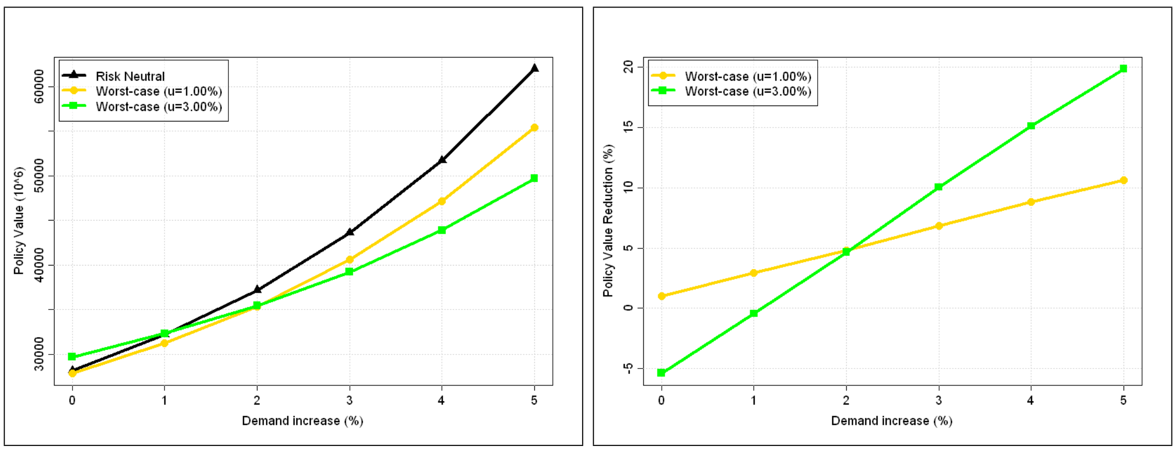


Figure 6: 120 stages policy values for risk neutral and worst-case-expectation ($u = 1\%$ and $u = 3\%$)

On Figure 7 we can see the average stage costs for the forward simulation with 1% of increase on the demand, and on Figure 8 the same average costs together with 95% and 99% quantiles for the forward simulation. We can notice that the average costs of cheaper stages were increased while the average costs of more expensive stages were reduced. This same behaviour is observed on 95% and 99% quantiles, for which the peak values were reduced.

The annual load curtailment frequencies for each depth (Table 1), are shown for the five years of the planning horizon on Figure 9. This simulation was done with an increase of 1% on the demand. The worst-case-expectation approach was able to reduce the deficit frequencies to less than its half values in most years and deficit depths.

The total CPU time used by each case study is compared to the "Risk Neutral" SDDP approach on Table 7. There was no extra CPU time to consider the demand uncertainty in the worst-case approach with $L = 1$, that is, sampling one demand for each stage at each iteration.

Table 7: Total CPU time

Case Study	dd:hh:mm:ss
Risk Neutral	01:06:57:40
$r = 1.00\%$ and $L = 1$	01:06:50:30
$r = 3.00\%$ and $L = 1$	01:07:02:59

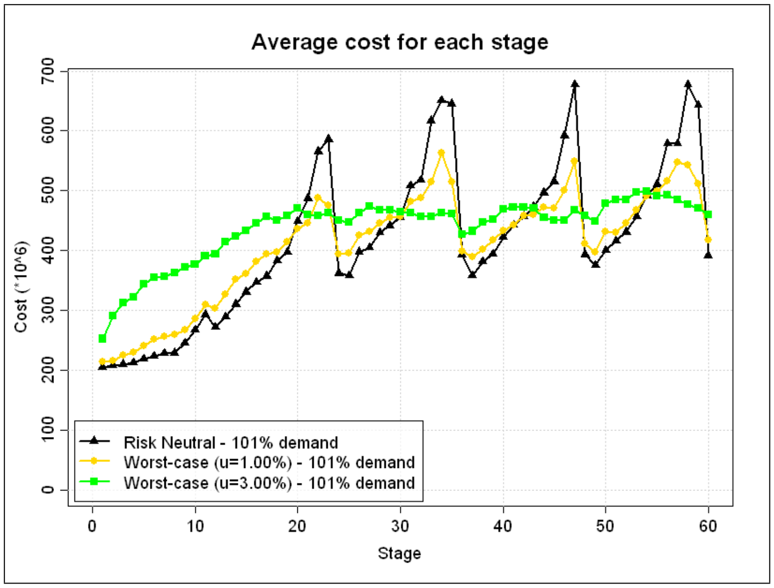


Figure 7: Average individual stage costs for risk neutral and worst-case-expectation ($u = 1\%$ and $u = 3\%$)

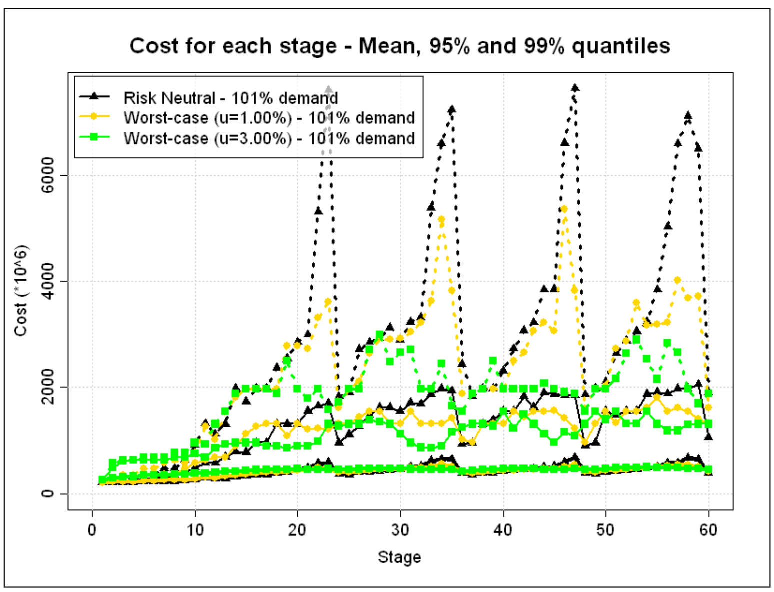


Figure 8: Individual stage costs average, 95% and 99% quantiles for risk neutral and worst-case-expectation ($u = 1\%$ and $u = 3\%$)

5.6 Risk averse vs. Worst-case expectation approaches

In this section, we compare the risk averse approach, discussed in [9], and the worst-case-expectation method suggested in this paper. We run the mean-AV@R risk averse approach with $\lambda = 0.15$ and $\alpha = 0.05$ for 3000 iterations and evaluate the policy using 2000 randomly generated scenarios. This value of λ achieves the lowest 99% quantile among candidates with 0.05 increments (i.e.

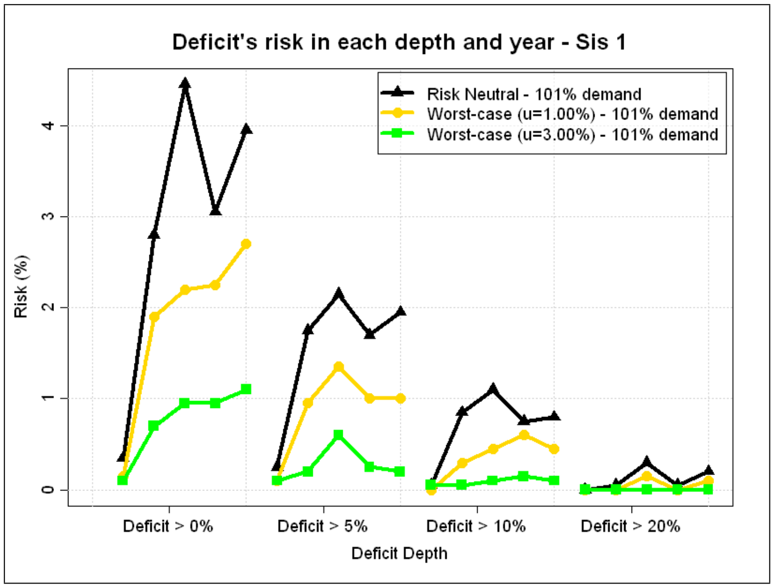


Figure 9: Annual deficit's risks for each depth of load curtailment for South-East system

$0.05, 0.10, 0.15, \ldots)$ for $\alpha \in \{0.05, 0.1\}$. We consider the worst-case-expectation method with $u = 3\%$.

Figure 10 plots the average, 95% and 99% quantiles of the total cost as function of the demand increase for the worst-case-expectation and the risk averse approach. When the forecast demand data is used (i.e. demand increase = 0%), the worst-case-expectation approach has lower average and almost similar 95% and 99% quantiles when compared with the risk averse method. The worst-case-expectation approach has considerably lower average policy value consistently as the demand increases. Furthermore, it outperforms the risk averse method with lower 95% and 99% quantiles when the demand increase is greater than 2%.

Figure 11 shows the average individual stage costs with 0% and 1% demand increase for the worst-case-expectation and the risk averse approach. In most of the first 100 stages, the worst-case-expectation approach has lower average value when compared to the risk averse method. However, in final stages, higher costs occur for the former method. The worst-case-expectation approach allows a smoother average individual costs across stages than the risk averse approach. An increase of 1% in the demand process shifts up approximately in similar manner the average costs for both methods.

Figure 12 plots the 99% quantile of the individual stage costs with 0% and 1% demand increase for the worst-case-expectation and the risk averse approach. The worst-case-expectation approach has higher 99% quantile value most of the time with 0% and 1% demand increase. In some sense, a better performance at this level is expected for the risk averse method, especially under low demand increase.

Figure 13 plots the average and 99% quantile of the individual stage costs with 3% demand increase for the worst-case-expectation and the risk averse approach. The worst-case-expectation approach has lower average individual stage costs than the risk averse method for most of the stages. Furthermore, the latter method exhibits significantly higher 99% quantile peaks. It is expected that the worst-case-expectation is less sensitive than the risk averse method to relatively high demand perturbation.

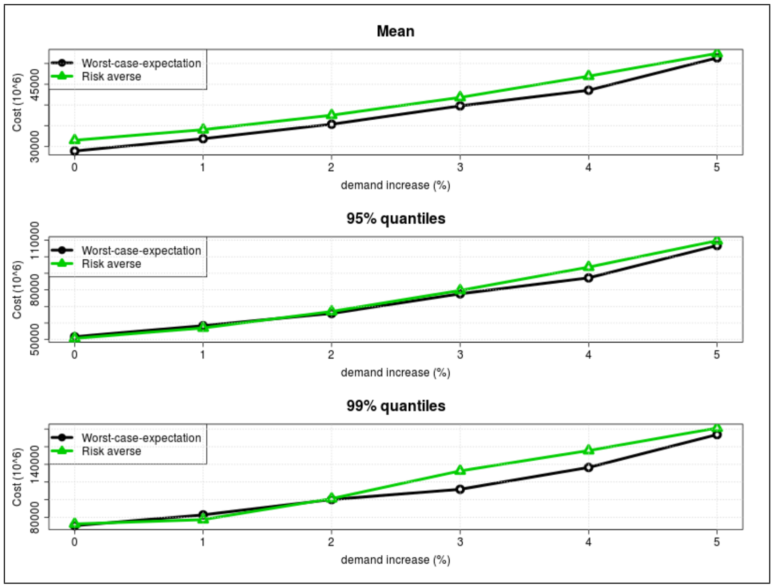
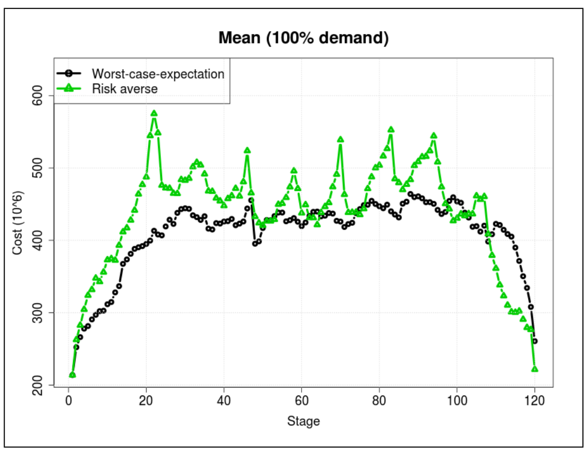


Figure 10: 120 stages discounted cost as function of demand increase



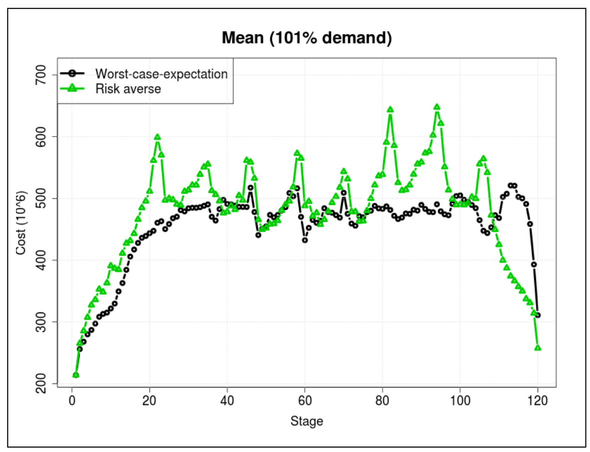


Figure 11: Average individual stage costs with 0% and 1% demand increase

6 Conclusion

In this paper, we investigated a multistage stochastic programming problem where the data process can be naturally separated into two components, one can be modeled as a random process, with a specified probability distribution, and the other one can be treated from a robust point of view. In sections 2 and 3, the basic ideas were discussed in the static and multistage setting and a time consistent formulation with the corresponding dynamic programming equations is presented.

In order to solve the obtained multistage problem an approach based on the Stochastic Dual Dynamic Programming method is suggested in section 4.

Finally, in section 5, we discussed numerical experiments with this approach applied to Brazilian operation planning of hydro plants. The worst-case-expectation approach constructs a policy that is less sensitive to unexpected demand increase with a reasonable loss on average when compared

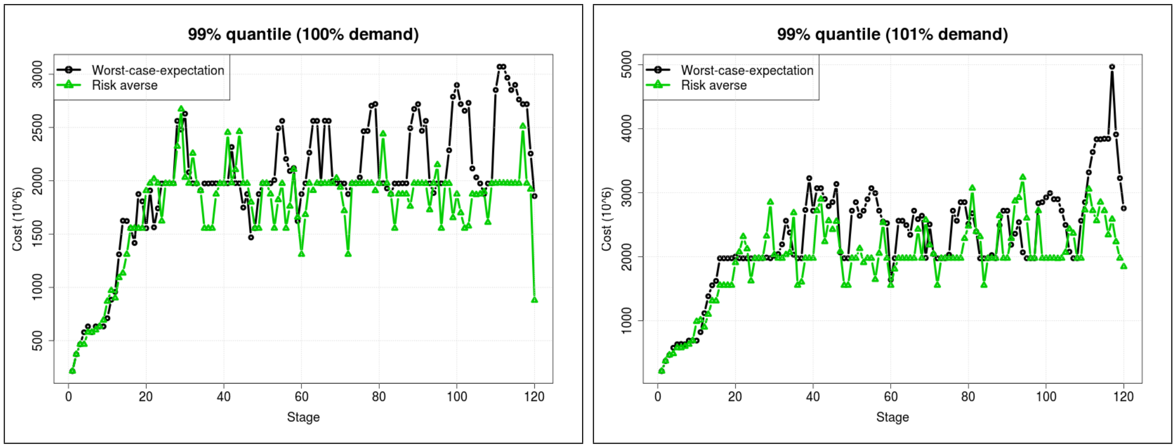


Figure 12: 99% quantile individual stage costs with 0% and 1% demand increase

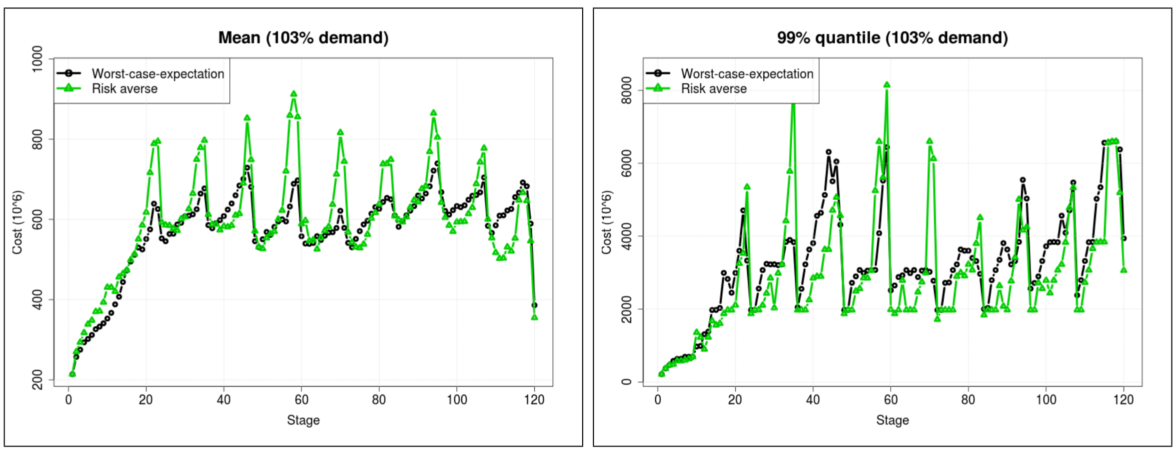


Figure 13: 99% quantile individual stage costs with 0% and 1% demand increase

to the risk neutral method. Also, we compared the suggested method with a risk averse approach based on coherent risk measures. On the one hand, the idea behind the risk averse method is to allow a trade off between loss on average and immunity against unexpected extreme scenarios. On the other hand, the worst-case-expectation approach consists in a trade off between a loss on average and immunity against unanticipated demand increase. The comparison confirms the purpose of both of the methods.

References

[1] A. Ben-Tal, L. El Ghaoui and A. Nemirovski, *Robust Optimization*, Princeton University Press, Princeton, 2009.

[2] A.D. Ioffe and V.M. Tihomirov, *Theory of Extremal Problems*, North-Holland, Amsterdam, 1979.

[3] M.V.F. Pereira and L.M.V.G. Pinto, Multi-stage stochastic optimization applied to energy planning, *Mathematical Programming*, 52 (1991), 359–375.

[4] A.B. Philpott and V.L. de Matos, Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion, *European Journal of Operational Research*, 218 (2012), 470–483.

[5] A. Ruszczyński and A. Shapiro, Conditional risk mappings, *Mathematics of Operations Research*, 31 (2006), 544–561.

[6] A. Shapiro, Analysis of stochastic dual dynamic programming method, *European Journal of Operational Research*, 209 (2011), 63–72.

[7] A. Shapiro, D. Dentcheva and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, Philadelphia, 2009.

[8] A. Shapiro, Minimax and risk averse multistage stochastic programming, *European Journal of Operational Research*, 219 (2012), 719–726.

[9] A. Shapiro, W. Tekaya, J.P. Da Costa and M.P. Soares, Risk neutral and risk averse Stochastic Dual Dynamic Programming method, *European Journal of Operational Research*, 224 (2013), 375–391.

[10] V. Strassen, The existence of probability measures with given marginals, *Annals of Mathematical Statistics*, 38 (1965), 423-439.