Cut-Sharing Across Trees and Efficient Sequential Sampling for SDDP with Uncertainty in the RHS

Pedro Borges*

Abstract

In this paper we show that when a multistage stochastic problem with stage-wise independent realizations has only RHS uncertainties, solving one tree provides a valid lower bound for all trees with the same number of scenarios per stage without any additional computational effort. The only change to the traditional algorithm is the way cuts are calculated. Once the first tree is solved approximately, the inference of the statistical significance of the current number of scenarios per stage is performed solving for each new tree sampled an easy LP to get a lower bound for the new tree that is often tight. The result of the inference are fast estimates of the mean, variance and max variation of lower bounds across many trees. If the variance of the calculated lower bounds is small, we declare the true SDDP problem well approximated because the cutting planes model has a small sensitivity to the trees sampled. Otherwise, we increase the number of scenarios per stage and repeat. We do not make assumptions on the distributions of the random variables. The results are not asymptotic. Our method has applications to the determination of the correct number of scenarios per stage. The stage-wise independence assumption can be dropped as well as the constraint of having only RHS uncertainties. However, the sensitivity of the optimal value with respect to the tree is only for the RHS vectors. Extensions for uncertainties in the objective only are possible via the dual SDDP. We test our method numerically and the computational results are sound.

keywords RHS Sensitivity of Multistage Stochastic Problems, Sequential Sampling

1 Introduction

Multistage stochastic optimization (MSO) has been actively used in energy planning and finance to make decisions in a dynamic and stochastic setting. While the general problem is computationally intractable, the assumption that the stochastic process has stage-wise independent realizations is reasonable in some settings and induces great simplifications to solution methods via the technique known as cut-sharing. Important algorithms leveraging on this technique are the SDDP [PP91] and the CUPPS [CP99].

Cut-sharing is based on realizing that some value functions representing future costs in MSO are the same when the realizations are stage-wise independent. Therefore, there is no need to visit all branches of the tree to improve the current cutting planes representation of the value functions. This is the source of the enormous reduction on the computational cost per iteration made possible by SDDP. CUPPS went further with the cut-sharing principle by sharing cuts among realizations in the same stage when there is only right-hand side (RHS) uncertainty. Cut-sharing for some forms of inter-stage dependence on the realizations is shown in [IM96].

The most used algorithms for multistage stochastic optimization [PP91; CP99; Kel60; LW03] consider given a fixed tree of scenarios. However, specially for high dimensional stochastic and volatile processes, the issue of the statistical significance of the solution found using one tree is important [BM06; SM98]. To approximate and evaluate the quality of candidate solutions of problems with continuous distributions many techniques based on solving separately many sampled trees (sequential sampling) have been introduced [MMF11; MMF17; LSW06]. As far as we know, none of these techniques leverage on cut-sharing across trees to make faster inferences. Instead, they develop statistical theory to make inference about solutions found separately by traditional algorithms [PP91; LW03]. In contrast, we get more lower bounding information from the solution of a single tree.

[*] Instituto de Matemática Pura e Aplicada, Rio de Janeiro, RJ, Brazil (pborges@impa.br)

In general sequential sampling, solving many trees from scratch might be unavoidable as a conservative decision to check if the solution obtained is good enough or not. This is so because somebody might not be willing to take the risks of asymptotic statistics. This comes with sometimes heavy costs because SDDP simulations can be quite time consuming and expensive when clusters are used to solve big problems. In this sense, being able to evaluate the sensitivity of general trees with respect at least to RHS vectors might be helpful because it would concentrate computationally hard sampling on other components of the uncertainty. For this purpose, our technique might be helpful. Moreover, the need to solve bigger problems is always standing and therefore efficiency on the solution of important special cases is always welcome.

In some sense, our computational approach is based on trading the costly exact evaluations of optimal values used in sequential sampling with fast estimates of the lower bounds of these optimal values over many more trees. The lower bounds computed might be very loose if the total number of scenarios per stage is not enough to represent the continuous problem. However, if we conclude that these lower bounds do not change much across many other trees, we can also conclude that solving more trees from scratch is useless as a strategy to improve current confidence intervals because the first optimal value computed has essentially no sensitivity to the tree being used. In this sense, our contribution is more of a tool to state that the true model is reasonably approximated. Nonetheless, the lower bounds can be used as a proxy to build confidence intervals for the true optimal value.

Recently, the topic of sensitivity and duality of multistage stochastic optimization gained some attention [LCCLP20; GSC20; TW20] with the development of the dual SDDP algorithm. See also [HS06; Roc99]. The motivation of these new developments was to improve the computation of the upper bound on the traditional SDDP method as also considered in [PME13]. The method of [LCCLP20] focuses only on RHS uncertainty as we do, while [GSC20] is for general stage-wise independent uncertainty. In principle, [LCCLP20] should have applications in efficient sequential sampling for MSO. However, our method appears to be justifiable on its own because it comes without an additional computational cost.

As our technique applied to the primal SDDP can be used to evaluate fast lower bounds across many trees, it could also be possible to apply the same technique to the dual SDDP to obtain fast estimates of upper bounds across many trees under RHS re-sampling. However, it does not look clear for us how one would proceed should it be possible. On the other hand, as we also show, using the dual SDDP one can apply our techniques to get fast upper bound estimates on the optimal value of a tree under perturbations of the cost vectors as was initially pursued using other strategies in [BCC12] for the sensitivity of energy contracts to some prices.

The approach used in [BCC12] and also in [GSC20] is based on using Daskin's theorem [BS00] to estimate directional derivatives of the optimal value. However, they are forced to get an upper estimate on the directional derivative using the optimal solution computed because Daskin's theorem requires the knowledge of *all* primal optimal solutions. This approach is related to ours as we explain now. Instead of trying to estimate the directional derivative of the optimal value with respect to problem data, we estimate a *global* lower bound on the optimal value with a max-type function (the cutting planes model). As is well known, the directional derivatives of max functions can be computed easily identifying the “active set". Therefore, we can also get an estimate on the directional derivative of the true value function given a specific direction similarly to [BCC12]. Instead of fixing a direction, we are actually interested in evaluating the global lower bounds given by the cutting planes model as we expect these lower bounds to show small variance as a sign of good approximation of the true problem. Moreover, it seems to us that using the individual estimates of the directional derivatives at all directions as in [BCC12] is not appropriate when there are lots of possible directions because (i) analyzing more numbers is harder and (ii) the joint effect of these inclinations is better understood when they are plugged together in a cutting planes model that gives one single estimate for the optimal value as opposed to tables 6 and 7 of [BCC12].

In summary, our method is sort of a dual version of [BCC12], with more focus in cutting plane models to summarize jointly derivative information and with the strategy to replicate RHS parameters backwards explained opportunely that makes the lower bound calculations we advertise possible to the method at the expense of a slight increase in memory usage proportional to the number of stages and scenarios of the discretized SDDP problem. The same idea of free-floating cuts that we employ here to build the algorithm is also used to solve some multilevel and multistage stochastic equilibrium problems [BSLDS20].

In Section 2 we make some preliminary considerations. In Section 3 we show our approach for the two-stage case. In Section 4 we show the multistage case. In Section 5 we report the numerical experiments supporting our methods. In Section 6 we show extensions based on the dual SDDP.

2 Preliminaries

The subdifferential [RW09] of a finite-valued convex function P at a point z is defined as $\partial P(z) = \{\gamma : P(z') \geq P(z) + \gamma^\intercal(z' - z) \quad \forall z'\}$. The directional derivative $P'(z; d)$ of P at z and a direction d always exists and is such that $P'(z; d) = \max\{\gamma^\intercal d : \gamma \in \partial P(z)\}$.

We denote by conv D the smallest convex set containing D. The connection of the directional derivative of P can be made with the cutting planes model as follows. If we consider a max-type function $P(z) = \max\{f_j(z) : j = 1, \ldots, J\}$ where f_j are smooth convex functions, it is well known that

$$\partial P(z) = \text{conv}\{\nabla f_j(z) : f_j(z) = P(z)\}. \tag{1}$$

In this section we consider the widely used finite-valued convex value function given by

$$Q(x, h) := \min_{y \geq 0}\{c^\intercal y : Wy = Rh - Bx\}. \tag{2}$$

For a fixed pair $x = \hat{x}$ and $h = \hat{h}$ we can compute a Lagrange multiplier $\hat{\lambda}$ of the equality constraints associated with the optimization problem defining $Q(\hat{x}, \hat{h})$. Doing so for fixed a $h = \hat{h}$ yields the widely used formula

$$Q(x, \hat{h}) \geq Q(\hat{x}, \hat{h}) - \hat{\lambda}^\intercal B(x - \hat{x}) \quad \forall x. \tag{3}$$

Nonetheless, with the same multiplier $\hat{\lambda}$ already obtained we can compute the more general inequality

$$Q(x, h) \geq Q(\hat{x}, \hat{h}) - \hat{\lambda}^\intercal B(x - \hat{x}) + \hat{\lambda}^\intercal R(h - \hat{h}) \quad \forall x, \quad \forall h. \tag{4}$$

Therefore, defining $Q_{\hat{h}}(x) = Q(x, \hat{h})$, it follows that

$$-B^\intercal\hat{\lambda} \in \partial Q_{\hat{h}}(\hat{x}) \quad \text{and} \quad (-B^\intercal\hat{\lambda}, R^\intercal\hat{\lambda}) \in \partial Q(\hat{x}, \hat{h}).$$

Imagine now that $Q(x, h)$ represents a cost-to-go function [PP91; CP99] used in a cutting planes method (CPM). Then, x is a decision and h is a scenario already sampled. Using (4) instead of (3) implies that we are not going to decide over all dependencies of the cut (4). Instead, we shall have the free-floating part $\hat{\lambda}^\intercal R(h - \hat{h})$ depending on the new samples and the other traditional part depending on the decision x. Note that it is as if the free-floating part is not there if we do not re-sample h.

Inequality (4) combined with a proper strategy to replicate the RHS vectors backwards in MSO so that the dependence on the relevant RHS vectors is explicit is the topic of the paper. In other words, we are halfway done. Moreover, the use of (4) instead of (3) implies, as shall be clear later, that the dual iterates used to build cuts do not have a fixed dimension and other matrices also have to be dynamic. Observing these details is important to understand the changes in the algorithm because what is relevant is the computation of the lower bounding linear forms similar to (4).

Although we do not present the statistical theory for our approach, it can be easily explained by analogy with the mean and variance of the empirical mean. Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and $\{X_i : i \in \mathbb{N}\}$ be a sequence of independent and identically distributed random variables with finite mean $\mathbb{E}X_i = \mu$ and finite variance Var $X_i = \sigma^2$. The empirical mean with n samples is given by

$$\overline{X} = \frac{X_1 + \cdots + X_n}{n}. \tag{5}$$

It is well known that

$$\mathbb{E}\overline{X} = \mu, \quad \text{Var}\,\overline{X} = \frac{\sigma^2}{n} \quad \text{and} \quad \mathbb{P}(|X_i - \mu| \geq k\sigma) \leq \frac{1}{k^2}. \tag{6}$$

In our case, the variable X_i stands for optimal value of a sample average problem. Formulae (6) say that if the variance of X_i is small, then the probability that it is much different from the mean is low. It also says that the variance of the sample average decreases with more samples. Our method allows to compute fast a good proxy for the variance of X_i. Knowing σ^2 is important to apply formulae (6).

3 Two-Stage Stochastic Problems

The traditional two-stage stochastic problem (2TSP) consists of S scenarios in the second stage represented by a sequence $(h_{21}, \ldots, h_{2S})$ of RHS vectors. An actual sample of this RHS sequence is denoted by $(\hat{h}_{21}, \ldots, \hat{h}_{2S})$. It consists in solving

$$\min_x \quad c_1^\intercal x + S^{-1} \sum_{s=1}^{S} Q_s(x, \hat{h}_{2s}) \quad \text{s.t.} \quad x \in D_1 := \{x \geq 0 : W_1 x = h_1\} \tag{7}$$

where for $s = 1, \ldots, S$ we take

$$Q_s(x, h_{2s}) := \min_{y_s \geq 0} \{c_2^\intercal y_s : W_2 y_s = h_{2s} - B_2 x\}. \tag{8}$$

For convenience, it useful to define the full cost-to-go function given by

$$Q(x, h_{21}, \ldots, h_{2S}) := S^{-1} \sum_{s=1}^{S} Q_s(x, h_{2s}). \tag{9}$$

The 2TSP with continuous expectation in the objective is considered in [SM98; BM06]. To solve the continuous problem often one tries to analyze (7) for different samples $(\hat{h}_{21}, \ldots, \hat{h}_{2S})$ and values S. Our procedure consists in analyzing fast and empirically if the optimal value of (7) depends on the sample $(\hat{h}_{21}, \ldots, \hat{h}_{2S})$ for a fixed S because the expectation is not sample-dependent.

Note that although we do not consider (c, W, B) dependent on s, we could have done so. However, our technique would apply only for the sensitivity of the RHS vectors. In practice, this possibility is already accounted for on the notation Q_s that expresses the value function of the scenario problems separately. The algorithm for the sensitivity of 2TSPs is shown below.

Algorithm 3.1 (Sensitivity of 2TSPs with respect to RHS vectors)**.**

Initialization. *Set* $k = 1$. *Take* S *and a base sample* $(\hat{h}_{21}, \ldots, \hat{h}_{2S})$.

Step 1: Solve Sample Problem using Modified Cuts.

Initialization. *Take* $x^k \in D$.

Step 1.1: Get Subgradient. *For* $s = 1, \ldots, S$ *compute* $(\alpha_s^k, \beta_s^k) \in \partial Q_s(x^k, \hat{h}_{2s})$ *based on* (4).

Step 1.2: Define the Scenario Model. *Take*

$$Q_s^k(x, h_{2s}) := \max_{i=1,\ldots,k} \{Q_s(x^i, \hat{h}_{2s}) + (\alpha_s^i)^\intercal (x - x^i)\} + (\beta_s^i)^\intercal (h_{2s} - \hat{h}_{2s})\}. \tag{10}$$

Step 1.3: Define the Aggregate Model. *Take*

$$Q^k(x, h_{21}, \ldots, h_{2S}) := S^{-1} \sum_{s=1}^{S} Q_s^k(x, h_{2s}). \tag{11}$$

Step 1.4: Next Iterate. *Compute* $x^{k+1} \in \arg\min_x \quad c_1^\intercal x + Q^k(x, \hat{h}_{21}, \ldots, \hat{h}_{2S})$ *s.t.* $x \in D_1$.

Step 1.5: CPM Stopping Test. *Go to Step 2 if the standard gap is zero.*

Step 1.6: Loop. *Set* $k = k + 1$ *and go back to Step 1.1.*

Step 2: Evaluate Fast Lower Bounds. *For* $l = 1, \ldots, L$ *sample* $(\hat{h}_{21}^l, \ldots, \hat{h}_{2S}^l)$ *and compute*

$$\hat{v}^l := \min_x \quad c_1^\intercal x + Q^k(x, \hat{h}_{21}^l, \ldots, \hat{h}_{2S}^l) \quad \textit{s.t.} \quad x \in D_1. \tag{12}$$

Step 3: Output. *Compute statistics of interest over* $\{\hat{v}^l : l = 1, \ldots, L\}$.

Step 2 of Algorithm 3.1 consists in solving many LPs. For the current standards, this is not something to be concerned about. Nonetheless, step 2 can be further simplified computing one lower bounding cut for the value function of the first stage problem as a function of the scenarios $(h_{21}, \dots, h_{2S})$ so that $\hat{v}^l$ would be estimated evaluating an affine expression, which is absolutely inexpensive.

Note that the actual cuts appearing in step 1.4 are equal to the traditional ones based on formula (3) because $Q^k(x, h_{21}, \dots, h_{2S})$ is evaluated at $(\hat{h}_{21}, \dots, \hat{h}_{2S})$ and this makes the free-floating part mentioned in (4) vanish. Therefore, the inner CPM inside step 1 finishes with finitely many iterations under mild conditions [Sha11; Kel60]. There is at least one modification of Algorithm 3.1 that one is tempted to consider that is to improve the representation of $Q(x, h_{21}, \dots, h_{2S})$ using more base scenarios $(\hat{h}_{21}, \dots, \hat{h}_{2S})$. This would translate into repeating step 1 some times. However, at least for our experience, this does not help much because the statistical properties of $\{\hat{v}^l : l = 1, \dots, L\}$ hardly change. The free-floating cuts we propose also allow to re-sample the base scenario inside step 1 and make one forward-backward for each new base scenario. While all these modifications are possible, once the sensitivity of the optimal values to new scenarios is small, as we check numerically, there is no reason for much more re-sampling.

4 Multistage Stochastic Problems

In this section we show how to perform the sensitivity of the results of the SDDP algorithm with respect to the RHS vectors. Naturally, the method is the extension of the one for the two-stage case applied recursively and backwards. In the last section, the cost-to-go function at $t = 1$ also depends on $(h_{21}, \dots, h_{2S})$. Analogously, for the SDDP, the cost-to-go function at stage t also depends on all the RHS vectors of the forward problems. With S stage-wise independent scenarios per stage and a total of T stages, these vectors can be represented by a sequence denoted by

$$h[t] := (h_{\tau s} \quad : \quad \tau = t+1, \dots, T, \quad s = 1, \dots, S). \tag{13}$$

We understand $h[t]$ as an empty tuple if $t \geq T$. Again, the actual sampling of these vectors is denoted by $\hat{h}[t]$. The first stage feasible set is the same as in the two-stage case and the feasible sets for $t = 2, \dots, T$ and $s = 1, \dots, S$ are given by

$$D_{ts}(x_{t-1}, h_{ts}) := \{x_{ts} \geq 0 \quad : \quad W_t x_{ts} = h_{ts} - B_t x_{t-1}\}. \tag{14}$$

Note that the feasible set D_{ts} depends also on h_{ts}. The first stage problem is given by

$$\min_{x_1} \quad c_1^\intercal x_1 + Q_2(x_1, h[1]) \quad \text{s.t.} \quad x_1 \in D_1 \tag{15}$$

where for $t = 2, \dots, T$ and $s = 1, \dots, S$ we have

$$Q_t(x_{t-1}, h[t-1]) := S^{-1} \sum_{r=1}^{S} Q_{tr}(x_{t-1}, h_{ts}, h[t]) \tag{16}$$

and

$$Q_{ts}(x_{t-1}, h_{ts}, h[t]) := \min_{x_{ts}} \quad c_t^\intercal x_{ts} + Q_{t+1}(x_{ts}, h[t]) \quad \text{s.t.} \quad x_{ts} \in D_{ts}(x_{t-1}, h_{ts}) \tag{17}$$

with

$$Q_{T+1}(\cdot, \cdot) \equiv 0. \tag{18}$$

For instance, note that (16) generalizes (9). Again, we do not take (c_t, W_t, B_t) depending on the scenario. However, as already explained we could have taken, in which case our method would be applied to the RHS sensitivity only. For both cases, the notation $Q_{ts}(x_{t-1}, h_{ts}, h[t])$ is general enough. As usual, our method is also based on building polyhedral approximations of Q_{ts} denoted Q_{ts}^k with the upper index k. The aggregate value function used on the algorithm below is defined for all k by

$$Q_t^k(x_{t-1}, h[t-1]) := S^{-1} \sum_{r=1}^{S} Q_{tr}^k(x_{t-1}, h_{ts}, h[t]). \tag{19}$$

Algorithm 4.1 (Sensitivity of SDDP problems with respect to RHS vectors)**.**

Initialization. *Set* $k = 1$*. Take* S *and a base sample* $\hat{h}[1]$*. Take* $M > 0$ *large. Set* $Q^k_{ts}(\cdot,\cdot,\cdot) \equiv -M$ *for all* $t = T, \dots, 2$ *and all* $s = 1, \dots, S$*. For all* $t = T, \dots, 2$ *the functions* $Q^k_t(\cdot,\cdot)$ *are given by formula* (19)*.*

Step 1: Solve Sample SDDP Problem using Modified Cuts.

Step 1.1: Sample Scenario. *Sample a scenario* $\omega^k_t \in \{1, \dots, S\} \quad \forall t = 2, \dots, T$*.*

Step 1.2: Modified Forward. *Compute*

$$\hat{x}^k_1 \in \arg\min_{x_1} \quad c_1^{\top} x_1 + Q^k_{12}(x_1, \hat{h}[1]) \quad \textit{s.t.} \quad x_1 \in D_1 \tag{20}$$

and for $t = 2, \dots, T$ *and* $s = \omega^k_t$ *compute*

$$\hat{x}^k_t \in \arg\min_{x_{ts}} \quad c_t^{\top} x_{ts} + Q^k_{t+1}(x_{ts}, \hat{h}[t]) \quad \textit{s.t.} \quad x_{ts} \in D_{ts}(\hat{x}^k_{t-1}, \hat{h}_{ts}). \tag{21}$$

Step 1.3: Modified Backward. *For all* $t = T, \dots, 2$ *and all* $s = 1, \dots, S$ *compute*

$$x^k_{ts} \in \arg\min_{x_{ts}} \quad c_t^{\top} x_{ts} + Q^{k+1}_{t+1}(x_{ts}, \hat{h}[t]) \quad \textit{s.t.} \quad x_{ts} \in D_{ts}(\hat{x}^k_{t-1}, \hat{h}_{ts}) \tag{22}$$

where $Q^{k+1}_{T+1}(\cdot,\cdot) \equiv 0$ *and for* $t = T-1, \dots, 1$ *take* $Q^{k+1}_{t+1}(\cdot,\cdot)$ *given by formula* (19)*. For* $t = T-1, \dots, 1$ *and* $s = 1, \dots, S$ *take* $Q^{k+1}_{t+1,s}(\cdot,\cdot,\cdot)$ *as a maximum between* $Q^k_{t+1,s}(\cdot,\cdot,\cdot)$ *and a free-floating cut of* $Q_{t+1,s}(\cdot,\cdot,\cdot)$ *at the point* $(\hat{x}^k_t, \hat{h}_{ts}, \hat{h}[t])$ *as explained in* (4)*.*

Step 1.4: Lower Bound. *Compute*

$$x^k_1 \in \arg\min_{x_1} \quad c_1^{\top} x_1 + Q^{k+1}_2(x_1, \hat{h}[1]) \quad \textit{s.t.} \quad x_1 \in D_1. \tag{23}$$

Step 1.5: Stopping Test. *Go to Step 2 if the lower and the average upper bounds are close enough.*

Step 1.6: Loop. *Set* $k = k + 1$ *and go back to Step 1.1.*

Step 2: Evaluate Fast Lower Bounds. *For* $l = 1, \dots, L$ *sample a new sequence* $\hat{h}^l[1]$ *and compute*

$$\hat{v}^l_1 := \min_{x_1} \quad c_1^{\top} x_1 + Q^k_1(x_1, \hat{h}^l[1]) \quad \textit{s.t.} \quad x_1 \in D_1. \tag{24}$$

Step 3: Output. *Compute statistics of interest over* $\{\hat{v}^l_1 : l = 1, \dots, L\}$*.*

As explained previously for the two-stage case, some variations of Algorithm 4.1 could have been considered. Among the most obvious possibilities, we could run step 1 more times or we could have proposed to sample trees inside step 1 and run one forward-backward for each new tree. Note that this is all possible within the free-floating cuts approach. Because the number of possible discretizations of the true SDDP is far greater than our ability to solve at least a minor portion of them, running step 1 more times re-using cuts already computed tends not to change the results in step 2. On the other hand, the sampling in step 2 can be quite exhaustive because L can be quite big. The problem with sampling trees inside step 1 is that the stopping test 1.5 would be arbitrary, as opposed to the current of having solved the base tree sampled in the start.

The convergence proof for the SDDP method inside step 1 is the same of the traditional case [Sha11] because the polyhedral approximations $Q^k_{t+1}(x_{ts}, \hat{h}[t])$ are the same. This is so because the free-floating parts of the cuts are evaluated exactly at the base point $h = \hat{h}$. Use formula (4) for the analogy. The contribution of the free-floating part of the cuts is only at step 2.

5 Experiments

The experiments consider the joint management of a cascade with a total of $G = 3$ hydros for $T = 12$ months. For ease of explanation we consider the configuration of the cascade, given next, fixed throughout. All experiments are run on an Intel i7 1.90GHz machine, with 15Gb of RAM, Ubuntu 18.04.3 LTS, CPLEX 12.10 and C++.

The water flows from the top of the cascade to the bottom. The hydro in the top is identified with $g = 1$, the next with $g = 2$ and so on. There is at most one hydro immediately next any other. The g-th hydro receives water from the rain and possibly the water released from the only upwards hydro. The hydros are used to produce energy that is sold at deterministic prices for simplicity. The inflows from the rain are stochastic. The objective is to maximize the expected profits of selling energy. First we show the management problem in the deterministic setting and then explain the differences for the stochastic setting.

The constants for each hydro are: $\boldsymbol{\rho}_g$ is the productivity of hydro g, $\boldsymbol{\pi}_t$ is the price of energy at time t, $\boldsymbol{a}_{gt}$ is the inflow at hydro g and time t, $\overline{\boldsymbol{v}}_g$ is the max volume of hydro g, $\underline{\boldsymbol{v}}_g$ is the dead volume of hydro g, $\overline{\boldsymbol{w}}_g$ is the maximum spillage capacity, $\overline{\boldsymbol{e}}_g$ and $\underline{\boldsymbol{e}}_g$ are, respectively, the maximum and minimum outflows for hydro g. The variables for each hydro are: u_{gt} is the turbined water at hydro g and time t, v_{gt} is the reservoir volume at hydro g and time t, w_{gt} is the spilled water at hydro g and time t. The problem solved is shown below, where some decisions are not present if $g - 1 < 1$.

$$\begin{aligned} \max \quad & \sum_{t,g} \boldsymbol{\rho}_g \boldsymbol{\pi}_t u_{gt} \\ \text{subject to} \quad & v_{g,t+1} = v_{gt} - u_{gt} - w_{gt} + u_{g-1,t} + w_{g-1,t} + \boldsymbol{a}_{lt}, \\ & u_{gT} + w_{gT} \leq v_{gT} + \boldsymbol{a}_{gT}, \\ & u_{gt} \leq v_{gt} - \underline{\boldsymbol{v}}_g, \\ & v_{gt} \in [0, \overline{\boldsymbol{v}}_g], \quad w_{gt} \in [0, \overline{\boldsymbol{w}}_g], \quad u_{gt} \in [0, \overline{\boldsymbol{u}}_g], \quad u_{gt} + w_{gt} \in [\underline{\boldsymbol{e}}_g, \overline{\boldsymbol{e}}_g]. \end{aligned} \tag{25}$$

Naturally, problem (25) suffers from the end-of-period effect. However, we use it during the experiments for illustrative purposes. The constraints $u_{gt} + w_{gt} \in [\underline{\boldsymbol{e}}_g, \overline{\boldsymbol{e}}_g]$ are written due to environmental reasons. Nonetheless, $u_{gt} + w_{gt} \in [\underline{\boldsymbol{e}}_g, \overline{\boldsymbol{e}}_g]$ and $w_{gt} \in [0, \overline{\boldsymbol{w}}_g]$ are neglected on the experiments below. Therefore, there is no need to report the quantities $\underline{\boldsymbol{e}}_g, \overline{\boldsymbol{e}}_g$ and $\overline{\boldsymbol{w}}_g$. The constraint $u_{gt} \leq v_{gt} - \underline{\boldsymbol{v}}_g$ says that water turbined is only of what exceeds the dead volume. The constraint $u_{gT} + w_{gT} \leq v_{gT} + \boldsymbol{a}_{gT}$ just says that at the end the reservoir level has to be non-negative and disregards the decisions of the other hydros.

Let us now explain what are the differences in the stochastic setting. The inflows $\boldsymbol{a}_{gt}$ are changed to $\boldsymbol{a}_{gts}$ for all $s = 1, \ldots, S$ because they are stochastic and assumed to be stage-wise independent. Still inside the SDDP framework, the inflows can also be modeled as an auto-regressive process (such as ARIMA) where the noise is stage-wise independent. We do not write the SDDP problem for simplicity. However, recall that it would have the stage-wise decomposition as opposed to the deterministic formulation above.

Indeed, such ARIMA model mentioned above is one of the motivations of this work because the noise follows a continuous distribution in which case the sensitivities are important. For instance, the tendency is that we would need more scenarios to get a good representation of the uncertainty per stage if G is big. Nonetheless, the resulting problem could be computationally too hard and then we would like to be at least aware of sensitivities of the cutting planes model to the sample of the inflows. Instead of fitting an ARIMA, we make the experiments taking a deterministic profile and then adding different types of noises around the profile. We expect that distributions with heavier tails would need more scenarios to be represented properly. The deterministic profiles for prices and inflows are shown below.

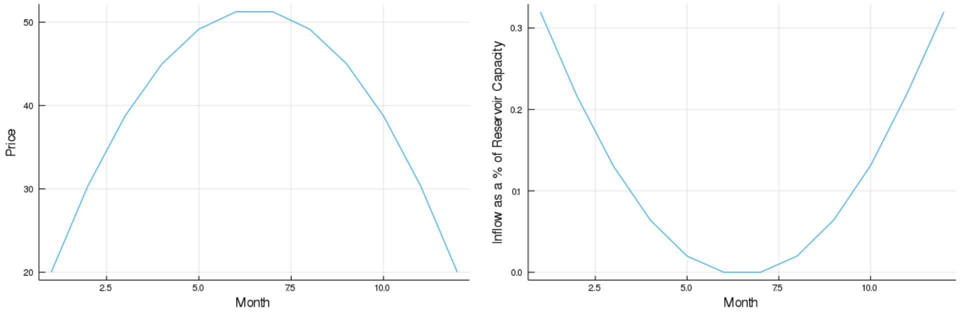

Figure 1: Note that the prices and inflows are “out-of-phase". The dry months happen around $t = 6$ as well as higher prices of energy. After adding noises to these profiles, we truncate negative results to zero.

We take the efficiencies $\boldsymbol{\rho}_g = 1$ for $g = 1, 2, 3$. The dead volumes are 10% of the maximum volumes and the initial volumes are 15% of the maximum volumes. The maximum turbining capacity per month is 50% of the maximum volume. The maximum volumes are $\overline{\boldsymbol{v}}_1 = 1.6, \overline{\boldsymbol{v}}_2 = 1.0$ and $\overline{\boldsymbol{v}}_3 = 1.6$. The limits for spillage, max outflow and min outflow are not used.

Now, all the data of the problem is given, except for the type of noise around the profiles in Figure 1 and the total scenarios used on the discretization. These last two items are the only that vary on the experiments that follow, which are organized into some parts reported next.

5.1 Validity of Lower Bounds Across Trees

In this subsection we check that the lower bounds we obtain are indeed valid. This is done by comparing the lower bounds of step 2 of Algorithm 4.1 with the lower and upper bounds obtained solving the new trees *separately* with the usual SDDP. For this purpose, the distributions of the noises applied to Figure 1 do not matter, as is clear from algorithms 3.1 and 4.1 because the lower bound is valid for all trees. The distribution just influences the rate of convergence of the sample average to the continuous problem. We also report on the SDDP* columns the solution of the new trees reusing previous free-floating cuts. The results are shown in Table 1.

New Tree	LB (Step 2)	LB (SDDP)	UB (SDDP)	LB (SDDP*)	UB (SDDP*)
$l = 1$	-736.49	-726.82	-710.18	-726.63	-726.01
$l = 2$	-735.22	-727.14	-707.41	-727.00	-725.97
$l = 3$	-734.44	-723.79	-702.66	-723.67	-720.59
$l = 4$	-729.25	-726.33	-708.59	-713.95	-713.40
$l = 5$	-728.22	-722.77	-703.15	-717.29	-712.17

Table 1: We use $S = 10$ scenarios per stage. The scenarios are generated adding a uniform noise around the inflow profile with range $[0, \eta]$, where η is the average inflow for the respective hydro. For each tree we perform 70 forward-backward iterations. First we solve the base tree. We find a lower bound of -724.37 and an upper bound of -706.58 (the numbers are negative because maximization turns into minimization). We can see that the SDDP* columns show lower gaps at the expense of bigger running times because the subproblems are bigger due to more cuts. The cuts from the solution of the base tree are passed to the $l = 1$ tree and the resulting cuts to $l = 2$ and so on. At $l = 5$ all the previous cuts are being reused.

New Tree	LB (Step 2)	LB (SDDP)	UB (SDDP)	LB (SDDP*)	UB (SDDP*)
$l = 1$	-743.16	-733.40	-716.59	-733.28	-731.43
$l = 2$	-734.63	-725.30	-707.34	-725.20	-717.59
$l = 3$	-735.90	-727.28	-712.83	-726.70	-722.25
$l = 4$	-736.23	-726.10	-708.37	-725.71	-725.05
$l = 5$	-734.38	-724.60	-703.41	-724.40	-718.35

Table 2: Here we use $S = 20$ and the same setting of Table 1. The base tree LB is -728.93 and UB is -712.04.

5.2 Evaluation of Scenario Reduction Techniques

In the context of scenario reduction [OSPMD10; DGKR03] we are given a big number of scenarios $\overline{S}$ and want to find a subset of size S of these scenarios that represents well the problem with $\overline{S}$ scenarios. The motivating assumption is that solving the problem with $\overline{S}$ scenarios is too hard or impossible. The techniques proposed [OSPMD10; DGKR03] are based on heuristics because selecting the subset of size S cannot take longer than solving the problem with size $\overline{S}$. Our technique of cut-sharing across trees can be used to evaluate a subset of size S once it is selected and solved.

Precisely, we can (i) use a technique of scenario reduction to select a subset of size S out of the $\overline{S}$ scenarios and (ii) sample, without replacement, other sets of S scenarios from the $\overline{S}$ and evaluate a fast lower bound. If the set of S scenarios is well selected by the scenario reduction method, the cutting planes model will not show significant sensitivity to other samples with the same size. This subsection tests this idea, but making the scenario reduction by selecting a subset of size S randomly. The results are shown in tables 3 and 4.

S	Lower Bound	Upper Bound	Expectation $\hat{v}_1^l$	Std. Dev. $\hat{v}_1^l$	$\max_l \lvert\hat{v}_1^l - \mathbb{E}\hat{v}_1^l\rvert$
1	-743.37	-729.29	-736.09	20.02	56.89
5	-721.61	-705.45	-742.30	13.01	36.19
10	-742.42	-725.64	-740.23	06.94	21.40
30	-718.12	-698.56	-744.64	03.85	11.65
50	-725.42	-703.48	-742.37	01.62	06.19
70	-727.42	-711.02	-739.51	00.19	00.69

Table 3: The scenarios are generated adding a uniform noise around the inflow profile with range $[0, \eta]$, where η is the average inflow for the respective hydro. We take $\overline{S} = 70$ and take randomly (without replacement) a base tree with S scenarios per stage. The first and second columns report the lower and upper bounds for the base tree after making 70 forward-backward iterations. We realize an out-of-sample analysis of the resulting cutting planes model taking $L = 400$ in step 2 of Algorithm 4.1. The new groups of S scenarios inside step 2 are sampled without replacement. We observe that the closer S is to $\overline{S}$, the more representative the results of the base tree are.

S	Lower Bound	Upper Bound	Expectation $\hat{v}_1^l$	Std. Dev. $\hat{v}_1^l$	$\max_l \lvert\hat{v}_1^l - \mathbb{E}\hat{v}_1^l\rvert$
1	-522.11	-511.82	-483.03	22.38	61.05
5	-487.49	-475.99	-481.63	09.13	25.07
10	-476.29	-467.22	-481.72	06.89	18.02
30	-480.82	-468.74	-481.95	02.88	09.13
50	-483.77	-474.89	-481.88	01.71	05.40
70	-481.42	-470.39	-482.11	00.01	00.05

Table 4: This table has the same settings of Table 3, but with a uniform noise with 40% of the original range. Note that for $S = \overline{S}$ the out-of-sample test is equivalent to randomly reordering the scenarios, which makes more numerical errors from the multiplier $\hat{\lambda}$ in (4) become apparent. This is the reason that for $S = 70$ we have the lower bound different from the expectation of $\hat{v}_1^l$.

5.3 Continuous Distributions with Unbounded Supports

We use noises with compact support on subsections 5.1 and 5.2. In this subsection we use our technique to evaluate the effect of sampling from continuous distributions with unbounded supports or heavy tails on the sensitivity of cutting plane models.

For our particular application, the occurrence of extreme values is equivalent to either a lot of rain flooding the reservoirs or no rain at all. For similar applications, the characterization of an extreme event can be done via the occurrence of a single bad realization of a one dimensional random variable. For some other applications, the extreme event is at ultimate analysis characterized by high costs. In other words, one would need to know the cost associated with the realizations to determine if it is extreme or not because the joint effects of the random variables could change the judgment. From an algorithmic perspective, our technique might be helpful because one would (i) solve a base tree, (ii) get estimates of the cost for many trees and (iii) select trees with the worst costs to be called extreme realizations, possibly improving the cost estimates for the trees selected and iterating back to (ii).

Next we show sensitivities of cutting plane models to the tree when the noise is normal. We take $L = 2000$ trees for the out-of-sample analysis and make 18 forward-backward iterations (much less than the previous experiments due to memory usage issues). For this reason, tables 5 and 6 report the statistical significance of the lower bounds obtained. The difference between tables 5 and 6 is that the nornal distribution of Table 6 has 50% of the standard deviation of the one for Table 5. Again, the normal noise is discretized with S samples to build the sample average problem. The first two columns of tables 5 and 6 refer to the base tree solved. We observe that "convergence" for Table 6 happens easier than for Table 5. As explained before, for big $S \geq 100$ we have the lower bound slightly different from the expectation of $\hat{v}_1^l$ because of numerical errors from the free-floating part of the cut. The quantity $\max_l |\hat{v}_1^l - \mathbb{E}\hat{v}_1^l|$ goes to zero because the problem has bounded feasible sets and is always feasible. Otherwise, sampling enough would make it diverge. The evolution of the histograms for $\hat{v}_1^l$ as a function of S for both tables 5 and 6 are shown in Figure 2.

S	Lower Bound	Upper Bound	Expectation $\hat{v}_1^l$	Std. Dev. $\hat{v}_1^l$	$\max_l \lvert\hat{v}_1^l - \mathbb{E}\hat{v}_1^l\rvert$
1	-306.42	-259.73	-424.46	91.87	305.69
5	-477.26	-423.01	-484.57	39.91	144.39
10	-426.86	-345.59	-467.20	28.50	111.90
30	-447.75	-430.41	-439.40	17.26	60.29
50	-461.76	-460.61	-486.18	12.52	42.03
70	-461.55	-433.26	-486.50	10.79	38.04
100	-467.25	-443.87	-484.30	9.08	33.20
150	-453.88	-418.31	-483.25	7.72	25.77
200	-468.83	-435.76	-487.70	6.35	22.23

Table 5: This tables shows the sensitivity of the cutting planes for a normal noise with zero mean. Recall that resulting negative inflows are truncated to zero.

S	Lower Bound	Upper Bound	Expectation $\hat{v}_1^l$	Std. Dev. $\hat{v}_1^l$	$\max_l \lvert\hat{v}_1^l - \mathbb{E}\hat{v}_1^l\rvert$
1	-368.58	-336.22	-288.87	52.06	214.27
5	-399.61	-353.66	-359.04	22.16	82.26
10	-355.54	-311.00	-358.68	16.40	69.45
30	-345.73	-304.63	-349.68	9.49	35.13
50	-338.62	-324.00	-345.04	7.21	25.36
70	-336.81	-286.73	-343.06	6.26	22.30
100	-345.41	-352.81	-349.11	5.09	16.79
150	-349.43	-317.25	-349.11	4.14	16.87
200	-344.54	-324.00	-348.55	3.72	13.72

Table 6: The normal noise for this experiments has 50% the standard deviation of the one for Table 5.

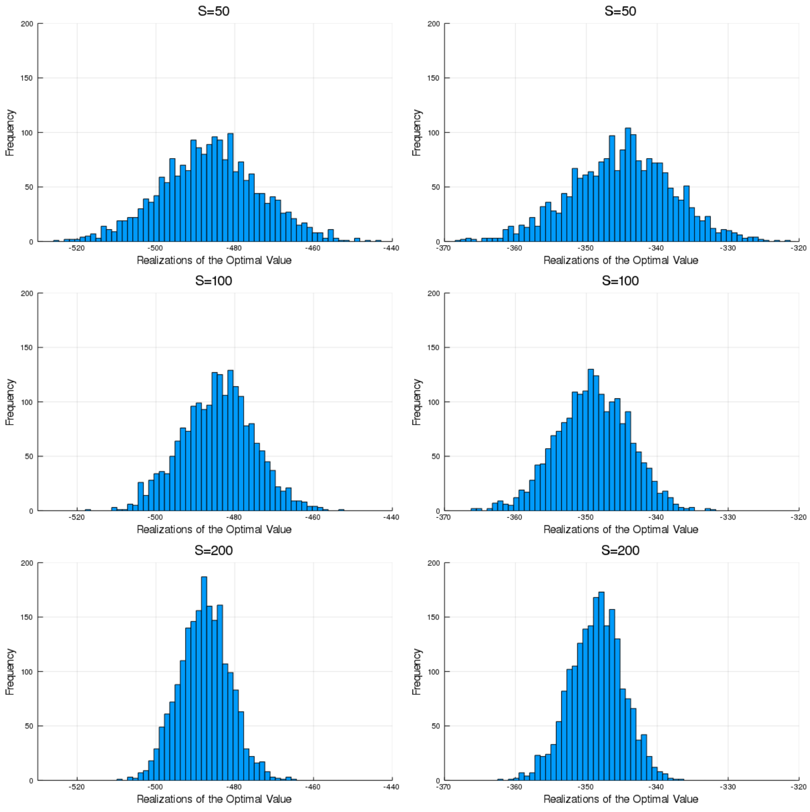

Figure 2: Histograms for the realizations of $\hat{v}_1^l$ for Table 5 (left) and Table 6 (right).

6 Extensions via Duality

In this section we replicate the dual problems found in [GSC20] and explain how our strategy for cut-sharing across trees applies. The dynamic programming equations (DPE) for the dual SDDP method [LCCLP20; GSC20] can be derived by (i) writing the dual of the deterministic equivalent primal problem and (ii) identifying stage-wise decomposition for the dual using concave value functions. Once this is performed, in spite of problems relating to feasibility that should be dealt with carefully [GSC20], the solution procedures for the dual tend to look familiar because the structure of the dual is somewhat similar to the usual primal.

The data process is denoted by $\xi_t = (c_t, W_t, B_t, h_t)$ and assumed to have finitely many realizations for all stages denoted by ξ_{ts} for $s = 1, \dots, S$. We also use $c_t = c_t(\xi_t)$, $W_t = W_t(\xi_t)$, $B_t = B_t(\xi_t)$ and $h_t = h_t(\xi_t)$. The history of the process is denoted by $\xi_{[t]} = (\xi_1, \dots, \xi_t)$. Note that ξ_1 is deterministic. The stage-wise independence translates into assuming that ξ_t is independent of $\xi_{[t-1]}$. The primal and dual multistage stochastic deterministic equivalent problems are respectively given by

$$\min_{x_t(\cdot)} \quad \mathbb{E}\big[\sum_{t=1}^{T} c_t x_t\big] \quad \text{s.t.} \quad x_t \geq 0, \quad W_1 x_1 = h_1, \quad W_t x_t = h_t - B_t x_{t-1} \quad \forall t = 2, \dots, T \tag{26}$$

and

$$\max_{\pi_t(\cdot)} \quad \mathbb{E}\big[\sum_{t=1}^{T} h_t \pi_t\big] \quad \text{s.t.} \quad W_T^\intercal \pi_T \leq c_T, \quad W_{t-1}^\intercal \pi_{t-1} + \mathbb{E}_{|\xi[t-1]}\big[B_t^\intercal \pi_t\big] \leq c_{t-1} \quad \forall t = 2, \dots, T \tag{27}$$

where the optimization is performed over policies $x_t = x_t(\xi_{[t]})$ and $\pi_t = \pi_t(\xi_{[t]})$ so that the non-antecipativity constraints are already implicitly being considered and the probability space where the expectation is considered is properly constructed. Standard duality theory applies to the pair (26)-(27).

As is widely known, the upper bound for the primal SDDP is calculated via Monte Carlo simulation. It is considered a weakness of the method because this process is slow and stochastic. One of the motivations of [LCCLP20; GSC20] was to generate deterministic upper bounds for the optimal value of a SDDP problem. The first stage problem of the dual SDDP is given

$$\max_{\pi_1} \quad h_1^\intercal \pi_1 + V_2(\pi_1, \xi_1). \tag{28}$$

For $t = 2, \dots, T-1$ the definition of $V_t(\pi_{t-1}, \xi_{t-1})$ is given by

$$\max_{\pi_{t1}, \dots, \pi_{tS}} \quad S^{-1} \sum_{s=1}^{S} h_{ts}^\intercal \pi_{ts} + S^{-1} \sum_{s=1}^{S} V_{t+1}(\pi_{ts}, \xi_{ts}) \quad \text{s.t.} \quad W_{t-1}^\intercal \pi_{t-1} + S^{-1} \sum_{s=1}^{S} B_{ts}^\intercal \pi_{ts} \leq c_{t-1}. \tag{29}$$

Finally, the definition of $V_T(\pi_{T-1}, \xi_{T-1})$ is

$$\max_{\pi_{T1}, \dots, \pi_{TS}} \quad S^{-1} \sum_{s=1}^{S} h_{Ts}^\intercal \pi_{Ts} \quad \text{s.t.} \quad W_{T-1}^\intercal \pi_{T-1} + S^{-1} \sum_{s=1}^{S} B_{Ts}^\intercal \pi_{Ts} \leq c_{T-1}, \quad W_{Ts}^\intercal \pi_{Ts} \leq c_{Ts} \quad \forall s. \tag{30}$$

First of all, note that (28) is unconstrained. Therefore, we run into bad iterates if the first forward passes are performed without care. This is so because we use upper bounding cutting plane models for these value functions. Secondly, the subproblems might be bigger because they involve all $\pi_{t1}, \dots, \pi_{tS}$ for all scenarios.

For our purposes, we have to note only that inside (29) the cost vectors of the primal appear in the RHS as well as π_{t-1}. Therefore, we can generate free-floating cuts approximating V_t in terms not only of π_{t-1} but also in terms of the realizations of the cost. This information is passed backwards replicating the RHS parameters in (29) analogously to what we have done before. The two-stage case is written below to illustrate. The first stage problem is the still (28) and the second stage value function $V_2(\pi_1, c_{21}, \dots, c_{2S})$ is

$$\max_{\pi_{21}, \dots, \pi_{2S}} \quad S^{-1} \sum_{s=1}^{S} h_{2s}^\intercal \pi_{2s} \quad \text{s.t.} \quad W_1^\intercal \pi_1 + S^{-1} \sum_{s=1}^{S} B_{2s}^\intercal \pi_{2s} \leq c_1, \quad W_{2s}^\intercal \pi_{2s} \leq c_{2s} \quad \forall s. \tag{31}$$

Assume given an iterate $(\hat{\pi}_1, \hat{c}_{21}, \ldots, \hat{c}_{2S})$ where $\hat{c}_{21}, \ldots, \hat{c}_{2S}$ are obtained via sampling. Using formulas similar to (4) we can generate supergradient vectors $(\hat{\alpha}_1, \hat{\beta}_{21}, \ldots, \hat{\beta}_{2S})$ such that

$$V_2(\pi_1, c_{21}, \ldots, c_{2S}) \leq V_2(\hat{\pi}_1, \hat{c}_{21}, \ldots, \hat{c}_{2S}) + \hat{\alpha}^\intercal(\pi_1 - \hat{\pi}_1) + \sum_{s=1}^{S} \hat{\beta}_{2s}^\intercal (c_{2s} - \hat{c}_{2s}) \quad \forall \pi_1, \quad \forall c_{2s}. \tag{32}$$

Inside a cutting planes method, formula (32) is used to improve a polyhedral approximation of the first stage problem (28) so that upper bounds for new cost vectors can be computed fast once one base tree is solved.

Acknowledgments Research of the author is funded by CNPq.

References

[BCC12] J. F. Bonnans, Z. Cen, and T. Christel. "Sensitivity Analysis of Energy Contracts by Stochastic Programming Techniques". *Numerical Methods in Finance* 12.1 (2012), pp. 447–471 (cit. on p. 2).

[BM06] G. Bayraksan and D. P. Morton. "Assessing solution quality in stochastic programs". *Mathematical Programming* 108.1 (2006), pp. 495–514 (cit. on pp. 1, 4).

[BS00] J. F. Bonnans and A. Shapiro. *Perturbation Analysis Of Optimization Problems*. Springer, 2000 (cit. on p. 2).

[BSLDS20] P. Borges, C. Sagastizábal, L. Liberti, C. D'Ambrósio, and M. Solodov. "Stochastic Dynamics of Multi-Owned Cascaded Hydro Systems". *In Preparation* (2020) (cit. on p. 2).

[CP99] Z. Chen and W. Powell. "Convergent cutting plane and partial sampling algorithm for multistage stochastic linear programs with recourse". *Journal of Optimization Theory and Applications* 102.1 (1999), pp. 497–524 (cit. on pp. 1, 3).

[DGKR03] J. Dupacova, N. Growe-Kuska, and W. Romisch. "Scenario reduction in stochastic programming: An approach using probability metrics". *Mathematical Programming* 95 (2003), pp. 493–511 (cit. on p. 8).

[GSC20] V. Guigues, A. Shapiro, and Y. Cheng. "Duality and sensitivity analysis of multistage linear stochastic programs". `http://www.optimization-online.org/DB_FILE/2019/11/7483.pdf` (2020) (cit. on pp. 2, 11).

[HS06] J. Higle and S. Sen. "Multistage stochastic convex programs: Duality and its implications". *Annals of Operations Research* 142.1 (2006), pp. 129–146 (cit. on p. 2).

[IM96] G. Infanger and D. P. Morton. "Cut sharing for multistage stochastic linear programs with interstage dependency". *Mathematical Programming* 75.1 (1996), pp. 241–256 (cit. on p. 1).

[Kel60] J. Kelley. "The cutting-plane method for solving convex programs". *Journal of the Society for Industrial and Applied Mathematics* 8.1 (1960), pp. 703–712 (cit. on pp. 1, 5).

[LCCLP20] V. Leclère, P. Carpentier, J.-P. Chancelier, A. Lenoir, and F. Pacaud. "Exact Converging Bounds for Stochastic Dual Dynamic Programming via Fenchel Duality". *SIAM Journal on Optimization* 30.2 (2020), pp. 1223–1250 (cit. on pp. 2, 11).

[LSW06] J. Linderoth, A. Shapiro, and S. Wright. "The empirical behavior of sampling methods for stochastic programming". *Annals of Operations Research* 142.1 (2006), pp. 215–241 (cit. on p. 1).

[LW03] J. Linderoth and S. Wright. "Decomposition Algorithms for Stochastic Programming on a Computational Grid". *Computational Optimization and Applications* 24.1 (2003), pp. 207–250 (cit. on p. 1).

[MMF11] T. Homem-de Mello, V. L. Matos, and E. C. Finardi. "Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling". *Energy Systems* 2 (2011), pp. 1–31 (cit. on p. 1).

[MMF17] V. L. Matos, D. P. Morton, and E. C. Finardi. “Assessing policy quality in a multistage stochastic program for long-term hydrothermal scheduling”. *Annals of Operations Research* 253 (2017), pp. 713–731 (cit. on p. 1).

[OSPMD10] W. L. Oliveira, C. Sagastizábal, D. D. J. Penna, M. E. P. Maceira, and J. M. Damázio. “Optimal scenario tree reduction for stochastic streamflows in power generation planning problems”. *Optimization Methods and Software* 25 (2010), pp. 917–936 (cit. on p. 8).

[PME13] A. Philpott, V. Matos, and F. Erlon. “On solving multistage stochastic programs with coherent risk measures”. *Operations Research* 61.4 (2013), pp. 957–970 (cit. on p. 2).

[PP91] M. Pereira and L. Pinto. “Multi-stage stochastic optimization applied to energy planning”. *Mathematical Programming* 52.1 (1991), pp. 359–375 (cit. on pp. 1, 3).

[Roc99] R. Rockafellar. “Duality and optimality in multistagestochastic programming”. *Annals of Operations Research* 85.0 (1999), pp. 1–19 (cit. on p. 2).

[RW09] T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, 2009 (cit. on p. 3).

[Sha11] A. Shapiro. “Analysis of stochastic dual dynamic programming method”. *European Journal of Operational Research* 209.1 (2011), pp. 63–72 (cit. on pp. 5, 6).

[SM98] A. Shapiro and T. Homem-de Mello. “A simulation-based approach to two-stage stochastic programming with recourse”. *Mathematical Programming* 81 (1998), pp. 301–325 (cit. on pp. 1, 4).

[TW20] G. Terça and D. Wozabal. “Envelope Theorems for Multi-Stage Linear Stochastic Optimization”. *http://www.optimization-online.org/DB_FILE/2018/06/6666.pdf* (2020) (cit. on p. 2).