DUALITY AND SENSITIVITY ANALYSIS OF MULTISTAGE LINEAR STOCHASTIC PROGRAMS

VINCENT GUIGUES*, ALEXANDER SHAPIRO†, AND YI CHENG ‡

Abstract. In this paper we investigate the dual of a Multistage Stochastic Linear Program (MSLP) to study two related questions for this class of problems. The first of these questions is the study of the optimal value of the problem as a function of the involved parameters. For this sensitivity analysis problem, we provide formulas for the derivatives of the value function with respect to the parameters and illustrate their application on an inventory problem. Since these formulas involve optimal dual solutions, we need an algorithm that computes such solutions to use them, i.e., we need to solve the dual problem.

In this context, the second question we address is the study of solution methods for the dual problem. Writing Dynamic Programming equations for the dual, we can use an SDDP type method, called Dual SDDP, which solves these Dynamic Programming equations computing a sequence of nonincreasing deterministic upper bounds on the optimal value of the problem. However, applying this method will only be possible if the Relatively Complete Recourse (RCR) holds for the dual. Since the RCR assumption may fail to hold (even for simple problems), we design two variants of Dual SDDP, namely Dual SDDP with penalizations and Dual SDDP with feasibility cuts, that converge to the optimal value of the dual (and therefore primal when there is no duality gap) problem under mild assumptions. We also show that optimal dual solutions can be obtained computing dual solutions of the subproblems solved when applying Primal SDDP to the original primal MSLP.

The study of this second question allows us to take a fresh look at the class of MSLP with interstage dependent cost coefficients. Indeed, for this class of problems, cost-to-go functions are non-convex and solution methods were so far using SDDP for a Markov chain approximation of the cost coefficients process. For these problems, we propose to apply Dual SDDP with penalizations to the cost-to-go functions of the dual which are concave. This algorithm converges to the optimal value of the problem.

Finally, as a proof of concept of the tools developed, we present the results of numerical experiments computing the sensitivity of the optimal value of an inventory problem as a function of parameters of the demand process and compare Primal and Dual SDDP on the inventory and a hydro-thermal planning problems.

Key words. Stochastic optimization, Sensitivity analysis, SDDP, Dual SDDP, Relatively complete recourse.

AMS subject classifications. 90C15, 90C90, 90C30

1. Introduction. Duality plays a key role in optimization. For generic optimization problems, weak duality allows to bound the optimal value. Dual information is also used in many optimization algorithms such as Uzawa algorithm [2], primal-dual projected gradient [21] or Stochastic Dual Dynamic Programming (SDDP) [22]. Moreover, for several classes of optimization problems, the dual is easier to solve than the primal problem, for instance when it is amenable to decomposition techniques such as price decomposition [4]. Even when there is a duality gap between the primal and dual optimal values, solving the dual already gives a bound on the optimal value, as mentioned earlier. Duality is also a fundamental tool in the reformulation of Robust Optimization problems, see for instance [3]. Finally, derivatives of the value

*School of Applied Mathematics, Fundação Getulio Vargas, 190 Praia de Botafogo, Rio de Janeiro, Brazil, (vincent.guigues@fgv.br). Research of this author was partially supported by CNPq grants 204872/2018-9 and 311289/2016-9.

† School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA, (ashapiro@isye.gatech.edu). Research of this author was partly supported by NSF grant 1633196.

‡ School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA, (cheng.yi@gatech.edu).

function of classes of optimization problems can be related to optimal dual solutions, see [5], [24] and more recently [10, 12, 8] for the characterization of subdifferentials, subgradients, and ε-subgradients of value functions of convex optimization problems.

For stochastic control problems, stochastic Lagrange multipliers were already used in [16, 17, 18]. In the context of multistage stochastic programs, duality was studied in [26, 14], see also [28] for a review. More recently, the sensitivity analysis of multistage stochastic programs was discussed in [6] and [30]. In [6] the authors study the sensitivity with respect to parameters driving the considered price model. The corresponding parameters are in the objective function and the analysis of the estimate of marginal price is based on Danskin's theorem with the SDDP method used for the numerical calculations. In [30], the authors use the Envelope Theorem for the sensitivity analysis. The required derivatives are described in terms of Lagrange multipliers associated with the value functions.

In this paper, focusing our attention on the dual of a Multistage Stochastic Linear Program (MSLP), we are able to provide insights into three important problems for MSLPs: sensitivity analysis, computation of a sequence of deterministic upper bounds on the optimal value which converges to the optimal value, and use of duality to solve Dynamic Programming (DP) equations on the dual which are simpler to solve (in the sense that they have convex cost-to-go functions) than primal DP equations for problems with interstage dependent cost coefficients. Our main contributions are summarized below.

Sensitivity analysis of MSLPs. We explain how to compute derivatives of the optimal value, seen as a function of the problem parameters, of a MSLP in terms of dual optimal solutions. Therefore, the construction of the dual problem is essential for our approach, contrary to [6]. With respect to the sensitivity analysis [30], in our approach, we do not use value functions directly, which are not known and can only be approximated, but rather construct the dual problem which is solved by an SDDP type algorithm, called Dual SDDP.

Writing Dynamic Programming equations for the dual problem. A simple but crucial ingredient for our developments and subsequent analysis of solution methods for the dual problem of a MSLP is to write DP equations for that dual problem. We are not aware of another paper with these equations. However, a similar study was done in [19]. More precisely, for a stochastic linear control problem with uncertainty in the right-hand-side, in [19], DP equations are written for the conjugate of the cost-to-go functions and using an SDDP type method for these DP equations, a sequence of upper bounds on the MSP optimal value is constructed which is the sequence of conjugate of the approximate first stage cost-to-go functions evaluated at the initial state x_0. Our approach has the advantage of being much simpler: contrary to derivations in [19] which require some algebra, our DP equations can be immediately obtained from the dual problem formulation, this latter being known (given in [28] for instance). On top of that, we relax two assumptions made in [19]: (a) the relatively complete recourse assumption of the dual and (b) randomness in the right-hand-side of the constraints only and interstage independent. The next three paragraphs describe how the scope of (a) and (b) was extended in our analysis.

Dual SDDP for dual problems without relatively complete recourse. In [19], it is assumed that the dual problem of the considered MSLP satisfies an assumption (Assumption 3) stronger than relatively complete recourse. This assumption may not be easy to check or may not be satisfied (for instance it is not satisfied for the inventory and hydro-thermal problems considered in Section 5). Therefore, it is desirable to extend the scope of Dual SDDP in such a way that it can still compute a deterministic converging sequence of upper bounds without this assumption. We present two variants of Dual SDDP that can do that: Dual SDDP with penalizations and Dual SDDP with feasibility cuts.

Dual SDDP for dual problems with all problem data random. Our DP equations are written for problems with uncertainty in all parameters. We explain how to apply Dual SDDP for such problems that do not satisfy (b) above.

Dual SDDP for problems with interstage dependent cost coefficients. Finally, we also relax assumption (b) considering problems having interstage dependent cost coefficients. Writing DP equations for the corresponding dual problem, we can apply Dual SDDP algorithm to solve these equations, which, interestingly, have concave cost-to-go functions whereas primal cost-to-go functions are not convex. This is in sharp contrast with the solution methods proposed so far such as [6, 20] which apply SDDP on the primal cost-to-go functions using a Markov chain approximation of the cost coefficients process.

The outline of the paper is the following. Our building blocks are elaborated in Section 2 where we write DP equations for the dual, we explain how to build upper bounding functions for the cost-to-go functions of the dual using penalizations, and study the dynamics of Lagrange multipliers. Sensitivity analysis of MSLPs is conducted in Section 3 while Dual SDDP and its variants are studied in Section 4. Finally, the results of numerical simulations testing the tools developed on an inventory and an hydro-thermal problem are presented in Section 5. The interested reader can find and test the code of all implementations and of Primal and Dual SDDP for MSLPs at https://github.com/vguigues/Dual_SDDP_Library_Matlab and https://github.com/vguigues/Primal_SDDP_Library_Matlab. Proofs are collected in the Appendix.

2. Duality of multistage linear stochastic programs.

2.1. Writing Dynamic Programming equations for the dual. Consider the multistage linear stochastic program

$$
\begin{array}{ll}
\min\limits_{x_t \geq 0} & \mathbb{E}\left[\sum_{t=1}^{T} c_t^\top x_t\right] \\
\text{s.t.} & A_1 x_1 = b_1, \\
 & B_t x_{t-1} + A_t x_t = b_t,\ t = 2, ..., T.
\end{array} \tag{2.1}
$$

Here vectors $c_t = c_t(\xi_t) \in \mathbb{R}^{n_t}$, $b_t = b_t(\xi_t) \in \mathbb{R}^{m_t}$ and matrices $B_t = B_t(\xi_t)$, $A_t = A_t(\xi_t)$ are functions of random process $\xi_t \in \mathbb{R}^{d_t}$, $t = 1, ..., T$ (with ξ_1 being deterministic). We denote by $\xi_{[t]} = (\xi_1, ..., \xi_t)$ the history of the data process up to time t and by $\mathbb{E}_{|\xi_{[t]}}$ the corresponding conditional expectation. The optimization in (2.1) is performed over functions (policies) $x_t = x_t(\xi_{[t]})$, $t = 1, ..., T$, of the data process satisfying the feasibility constraints.

The Lagrangian of problem (2.1) is

$$
L(x, \pi) = \mathbb{E}\left[\sum_{t=1}^{T} c_t^\top x_t + \pi_t^\top (b_t - B_t x_{t-1} - A_t x_t)\right] \tag{2.2}
$$

in variables[1] $x = (x_1(\xi_{[1]}), \ldots, x_T(\xi_{[T]}))$ and $\pi = (\pi_1(\xi_{[1]}), \ldots, \pi_T(\xi_{[T]}))$ with the convention that $x_0 = 0$. Dualization of the feasibility constraints leads to the following

[1] Note that since ξ_1 is deterministic, the first stage decision x_1 is also deterministic; we write it as $x_1(\xi_{[1]})$ for uniformity of notation, and similarly for π_1.

dual of problem (2.1) (cf., [28, Section 3.2.3]):

$$
\begin{array}{lll}
(2.3) & \max\limits_{\pi} & \mathbb{E}\big[\sum_{t=1}^{T} b_t^\top \pi_t\big] \\
& \text{s.t.} & A_T^\top \pi_T \le c_T, \\
& & A_{t-1}^\top \pi_{t-1} + \mathbb{E}_{|\xi_{[t-1]}}\left[B_t^\top \pi_t\right] \le c_{t-1},\ t = 2,...,T.
\end{array}
$$

The optimization in (2.3) is over policies $\pi_t = \pi_t(\xi_{[t]})$, $t = 1,...,T$.

Unless stated otherwise, we make the following assumption throughout the paper.

(A1) The process $\xi_1,...,\xi_T$ is stagewise independent (i.e., random vector ξ_{t+1} is independent of $\xi_{[t]}$, $t = 1,...,T-1$), and distribution of ξ_t has a finite support, $\{\xi_{t1},\dots,\xi_{tN_t}\}$ with respective probabilities p_{tj}, $j = 1,...,N_t$, $t = 2,...,T$. We denote by $A_{tj}, B_{tj}, c_{tj}, b_{tj}$ the respective scenarios corresponding to ξ_{tj}.

Since the random process ξ_t, $t = 1,...,T$, has a *finite* number of realizations (scenarios), problem (2.1) can be viewed as a large linear program and (2.3) as its dual. By the standard theory of linear programming we have the following.

PROPOSITION 2.1. *Suppose that problem* (2.1) *has a finite optimal value. Then the optimal values of problems* (2.1) *and* (2.3) *are equal to each other and both problems have optimal solutions.*

We can write the following dynamic programming equations for the dual problem (2.3). At the last stage $t = T$, given π_{T-1} and $\xi_{[T-1]}$, we need to solve the following problem with respect to π_T:

$$
\begin{array}{lll}
(2.4) & \max\limits_{\pi_T} & \mathbb{E}[b_T^\top \pi_T] \\
& \text{s.t.} & A_T^\top \pi_T \le c_T, \\
& & A_{T-1}^\top \pi_{T-1} + \mathbb{E}\left[B_T^\top \pi_T\right] \le c_{T-1}.
\end{array}
$$

Since ξ_T is independent of $\xi_{[T-1]}$, the expectation in (2.4) is unconditional with respect to the distribution of ξ_T. In terms of scenarios the above problem can be written as

$$
\begin{array}{lll}
(2.5) & \max\limits_{\pi_{T1},\dots,\pi_{TN_T}} & \sum\limits_{j=1}^{N_T} p_{Tj} b_{Tj}^\top \pi_{Tj} \\
& \text{s.t.} & A_{Tj}^\top \pi_{Tj} \le c_{Tj},\ j = 1,...,N_T, \\
& & A_{T-1}^\top \pi_{T-1} + \sum\limits_{j=1}^{N_T} p_{Tj} B_{Tj}^\top \pi_{Tj} \le c_{T-1}.
\end{array}
$$

The optimal value $V_T(\pi_{T-1}, \xi_{T-1})$ and an optimal solution[2] $(\bar{\pi}_{T1},\dots,\bar{\pi}_{TN_T})$ of problem (2.5) are functions of vectors π_{T-1} and c_{T-1} and matrix A_{T-1}. And so on going backward in time, using the stagewise independence assumption, we can write the respective dynamic programming equations for $t = T-1,...,2$, as

$$
\begin{array}{lll}
(2.6) & \max\limits_{\pi_{t1},\dots,\pi_{tN_t}} & \sum\limits_{j=1}^{N_t} p_{tj}\left[b_{tj}^\top \pi_{tj} + V_{t+1}(\pi_{tj}, \xi_{tj})\right] \\
& \text{s.t.} & A_{t-1}^\top \pi_{t-1} + \sum\limits_{j=1}^{N_t} p_{tj} B_{tj}^\top \pi_{tj} \le c_{t-1},
\end{array}
$$

with $V_t(\pi_{t-1}, \xi_{t-1})$ being the optimal value of problem (2.6). Finally at the first stage the following problem should be solved

$$
(2.7) \qquad \max_{\pi_1} b_1^\top \pi_1 + V_2(\pi_1, \xi_1).
$$

[2]Note that problem (2.5) may have more than one optimal solution. In case of finite number of scenarios the considered linear program always has a solution provided its optimal value is finite.

These dynamic programming equations can be compared with the dynamic programming equations for primal problem (2.1), where the respective cost-to-go (value) function $Q_t(x_{t-1}, \xi_{tj})$, $j = 1, ..., N_t$, is given by the optimal value of

$$\begin{array}{rl} \min\limits_{x_t \geq 0} & c_{tj}^\top x_t + \mathcal{Q}_{t+1}(x_t) \\ \text{s.t.} & B_{tj} x_{t-1} + A_{tj} x_t = b_{tj}, \end{array} \tag{2.8}$$

with

$$\mathcal{Q}_{t+1}(x_t) := \mathbb{E}[Q_{t+1}(x_t, \xi_{t+1})] = \sum_{j=1}^{N_t} p_{t+1j} Q_{t+1}(x_t, \xi_{t+1j}).$$

Let us make the following observations about the dual problem.

(i) Unlike in the primal problem, the optimization (maximization) problems (2.5) and (2.6) do not decompose into separate problems with respect to each π_{tj} and should be solved as one linear program with respect to $(\pi_{t1}, ..., \pi_{tN_t})$.

(ii) The value function $V_t(\pi_{t-1}, \xi_{t-1})$ is a concave function of π_{t-1}.

(iii) If A_t and c_t, $t = 2, ..., T$, are deterministic, then $V_t(\pi_{t-1})$ is only a function of π_{t-1}.

2.2. Relatively complete recourse. The following definition of Relatively Complete Recourse (RCR) is applied to the dual problem. Recall that we assume that the set of possible realizations (scenarios) of the data process is finite.

Definition 2.2. *We say that a sequence $\bar{\pi}_t$, $t = 1, ..., T$, is generated by the forward (dual) process if $\bar{\pi}_1 \in \mathbb{R}^{m_1}$ and for $\pi_{t-1} = \bar{\pi}_{t-1}$, $t = 2, ..., T$, going forward in time, $\bar{\pi}_t$ coincides with some π_{tj}, $j = 1, ..., N_t$, where $\pi_{t1}, \dots, \pi_{tN_t}$ is a feasible solution of the respective dynamic program - program (2.6) for $t = 2, ..., T-1$, and program (2.5) for $t = T$. We say that the dual problem (2.3) has* Relatively Complete Recourse (RCR) *if at every stage $t = 2, ..., T$, for any generated π_{t-1} by the forward process, the respective dynamic program has a feasible solution at stage t for every realization of the random data.*

Without RCR it could happen that $V_t(\pi_{t-1}, \xi_{t-1}) = -\infty$ for a generated π_{t-1} and $\xi_{t-1} = \xi_{t-1j}$. Unfortunately, it could happen that the dual problem does not have the RCR property even if the primal problem has it. This could happen even in the two stage case. One way to deal with the problem of absence of RCR in numerical procedures is to use feasibility cuts, we will discuss this later. Another way is the following penalty approach which will be used in Section 4. The infeasibility of problem (2.5) can happen because of its last constraint. In order to deal with this, consider the following relaxation of problem (2.5):

$$\begin{array}{rl} \max\limits_{\pi_{T1}, ..., \pi_{TN_T}, \zeta_T \geq 0} & \sum\limits_{j=1}^{N_T} p_{Tj} b_{Tj}^\top \pi_{Tj} - v_T^\top \zeta_T \\ \text{s.t.} & A_{Tj}^\top \pi_{Tj} \leq c_{Tj}, \; j = 1, ..., N_T, \\ & A_{T-1}^\top \pi_{T-1} + \sum\limits_{j=1}^{N_T} p_{Tj} B_{Tj}^\top \pi_{Tj} \leq c_{T-1} + \zeta_T, \end{array} \tag{2.9}$$

where v_T is a vector with positive components. We have that problem (2.9) is always feasible and hence its optimal value $\tilde{V}_T(\pi_{T-1}, \xi_{T-1}) > -\infty$. We also have that

$$\tilde{V}_T(\pi_{T-1}, \xi_{T-1}) \geq V_T(\pi_{T-1}, \xi_{T-1}), \tag{2.10}$$

with the equality holding if $\zeta_T = 0$ in the optimal solution of (2.9). If $V_T(\pi_{T-1},\xi_{T-1})$ is finite, this equality holds if the components of vector υ_T are large enough.

Similarly, problems (2.6) can be relaxed to

$$\begin{array}{cll} \text{(2.11)} & \max\limits_{\pi_{t1},\dots,\pi_{tN_t},\zeta_t\geq 0} & \sum\limits_{j=1}^{N_t} p_{tj}\left[b_{tj}^\top \pi_{tj} + \tilde{V}_{t+1}(\pi_{tj},\xi_{tj})\right] - \upsilon_t^\top \zeta_t \\ & \text{s.t.} & A_{t-1}^\top \pi_{t-1} + \sum\limits_{j=1}^{N_t} p_{tj} B_{tj}^\top \pi_{tj} \leq c_{t-1} + \zeta_t, \end{array}$$

with vector υ_t having positive components. In that way, the infeasibility problem is avoided and the obtained value gives an upper bound for the optimal value of the dual problem. Note that for sufficiently large vectors υ_t this upper bound coincides with the optimal value of the dual problem.

2.3. Dynamics of Lagrange multipliers. Let us consider for the moment the two stage setting, i.e., $T = 2$. The primal problem can be written as

$$\text{(2.12)} \qquad \min_{x_1\geq 0} c_1^\top x_1 + \mathbb{E}\left[Q(x_1,\xi_2)\right] \text{ s.t. } A_1 x_1 = b_1,$$

where $Q(x_1,\xi_2)$ is the optimal value of the second stage problem

$$\text{(2.13)} \qquad \min_{x_2\geq 0} c_2(\xi_2)^\top x_2 \text{ s.t. } B_2(\xi_2)x_1 + A_2(\xi_2)x_2 = b_2(\xi_2).$$

The Lagrangian of problem (2.13) is

$$L(x_1,x_2,\lambda,\xi_2) = c_2(\xi_2)^\top x_2 + \lambda^\top (b_2(\xi_2) - B_2(\xi_2)x_1 - A_2(\xi_2)x_2).$$

In the dual form, $Q(x_1,\xi_{2j})$ is given by the optimal value of the problem

$$\text{(2.14)} \qquad \max_{\lambda_j} (b_{2j} - B_{2j}x_1)^\top \lambda_j \text{ s.t. } c_{2j} - A_{2j}^\top \lambda_j \geq 0.$$

We have that if $x_1 = \bar{x}_1$ is an optimal solution of the first stage problem, then optimal Lagrange multipliers π_{2j} are given by the optimal solution of problem (2.14).

This can be extended to the multistage setting of problem (2.1) (recall that the stagewise independence condition is assumed). At the last stage $t = T$, given optimal solution $\bar{x}_{T-1}$, the following problem should be solved

$$\text{(2.15)} \qquad \min_{x_T\geq 0} c_T(\xi_T)^\top x_T \;\text{ s.t. }\; B_T(\xi_T)\bar{x}_{T-1} + A_T(\xi_T)x_T = b_T(\xi_T).$$

For a realization $\xi_T = \xi_{Tj}$, the dual of problem (2.15) is the problem

$$\text{(2.16)} \qquad \max_{\lambda_j} (b_{Tj} - B_{Tj}\bar{x}_{T-1})^\top \lambda_j \;\text{ s.t. }\; c_{Tj} - A_{Tj}^\top \lambda_j \geq 0.$$

We then have that π_{Tj} are given by the optimal solution of problem (2.16).

At stage $t = T-1$, given optimal solution $\bar{x}_{T-2}$, the following problem is supposed to be solved (see (2.8))

$$\begin{array}{cll} \text{(2.17)} & \min\limits_{x_{T-1}\geq 0} & c_{T-1}(\xi_{T-1})^\top x_{T-1} + \mathcal{Q}_T(x_{T-1}) \\ & \text{s.t.} & A_{T-1}(\xi_{T-1})x_{T-1} = b_{T-1}(\xi_{T-1}) - B_{T-1}(\xi_{T-1})\bar{x}_{T-2}. \end{array}$$

We have that $\mathcal{Q}_T(\cdot)$ is a convex piecewise linear function. Therefore for every realization $\xi_{T-1} = \xi_{T-1j}$ it is possible to represent (2.17) as a linear program and hence to write its dual. The optimal Lagrange multipliers of that dual give the corresponding Lagrange multipliers π_{T-1j}. And so on for other stages going backward in time. That is, we have the following.

Remark 2.1. *If* $(\bar{x}_1, ..., \bar{x}_T(\xi_{[T]}))$ *is an optimal solution of the primal problem, then for* $x_{t-1} = \bar{x}_{t-1}$ *the Lagrange multiplier* π_{tj} *is given by the respective Lagrange multiplier of problem* (2.8).

We also refer to [15, 25] for the dynamics of dual solutions to stochastic programs.

3. Sensitivity analysis. In this section we discuss an application of the duality analysis to a study of sensitivity of the optimal value to small perturbations of the involved parameters.

3.1. General case. Suppose now that the data $c_t(\xi_t, \theta), b_t(\xi_t, \theta), B_t(\xi_t, \theta), A_t(\xi_t, \theta)$ of problem (2.1) also depend on parameter vector $\theta \in \mathbb{R}^k$. Denote by $\vartheta(\theta)$ the optimal value of the parameterized problem (2.1) considered as a function of θ, and by $\mathfrak{S}(\theta)$ and $\mathfrak{D}(\theta)$ the sets of optimal solutions of the respective primal and dual problems. Recall that the sets $\mathfrak{S}(\theta)$ and $\mathfrak{D}(\theta)$ are nonempty provided the optimal value $\vartheta(\theta)$ is finite. Let $L(x, \pi, \theta)$ be the corresponding Lagrangian (see (2.2)) considered as a function of θ. Then we have the following formula for the directional derivatives of the optimal value function (e.g., [5, Proposition 4.27]).

Proposition 3.1. *Suppose that the data functions are continuously differentiable functions of* θ*, and for a given* $\theta = \bar{\theta}$ *the optimal value* $\vartheta(\bar{\theta})$ *is finite and the sets* $\mathfrak{S}(\bar{\theta})$ *and* $\mathfrak{D}(\bar{\theta})$ *of optimal solutions are bounded. Then*

$$\vartheta'(\bar{\theta}, h) = \max_{\pi \in \mathfrak{D}(\bar{\theta})} \min_{x \in \mathfrak{S}(\bar{\theta})} h^\top \nabla_\theta L(x, \pi, \bar{\theta}). \tag{3.1}$$

In particular if $\mathfrak{S}(\bar{\theta}) = \{\bar{x}\}$ *and* $\mathfrak{D}(\bar{\theta}) = \{\bar{\pi}\}$ *are singletons, then* $\vartheta(\cdot)$ *is differentiable at* $\bar{\theta}$ *and*

$$\nabla \vartheta(\bar{\theta}) = \nabla_\theta L(\bar{x}, \bar{\pi}, \bar{\theta}). \tag{3.2}$$

Next, as an example, we consider the sensitivity analysis of an inventory model.

3.2. Application to an inventory model. Consider the inventory model

$$\begin{array}{ll} \min & \mathbb{E}\left[\sum_{t=1}^{T} a_t(y_t - x_{t-1}) + g_t(\mathcal{D}_t - y_t)_+ + h_t(y_t - \mathcal{D}_t)_+\right] \\ \text{s.t.} & x_t = y_t - \mathcal{D}_t, y_t \geq x_{t-1}, t = 1 \ldots, T. \end{array} \tag{3.3}$$

Here $\mathcal{D}_1, ..., \mathcal{D}_T$ is a (random) demand process, a_t, g_t, h_t are the ordering, back-order penalty and holding costs per unit, respectively, x_t is the inventory level and $y_t - x_{t-1}$ is the order quantity at time t, the initial inventory level x_0 is given. We refer to [31] for a thorough discussion of that model. Note that $\mathcal{D}_t$ is a random variable whereas d_t stands for a particular realization. We assume that $g_t > a_t \geq 0$, $h_t > 0$, $t = 1, ..., T$.

In the classical setting the demand process is assumed to be stagewise independent, i.e., $\mathcal{D}_{t+1}$ is assumed to be independent of $\mathcal{D}_{[t]} = (\mathcal{D}_1, ..., \mathcal{D}_t)$ for $t = 1, ..., T-1$. In order to capture the autocorrelation structure of the demand process it is tempting to model it as, say first order, autoregressive process $\mathcal{D}_t = \mu + \phi\mathcal{D}_{t-1} + \epsilon_t$, where errors ϵ_t are assumed to be a sequence i.i.d (independent identically distributed) random variables. However this approach may result in some of the realizations of the demand process to be negative, which of course does not make sense. One way to deal with this is to make the transformation $Y_t := \log \mathcal{D}_t$ and to model Y_t as an autoregressive process. A problem with this approach is that it leads to nonlinear equations for the original process $\mathcal{D}_t$, which makes it difficult to use in the numerical algorithms discussed below.

We assume that the demand is modeled as the following multiplicative autoregressive process

$$\mathcal{D}_t = \epsilon_t(\phi\mathcal{D}_{t-1} + \mu),\ t = 1, ..., T, \tag{3.4}$$

where $\phi \in (0, 1)$, $\mu \geq 0$ are parameters and $\mathcal{D}_0 \geq 0$ is given. The errors ϵ_t are i.i.d with log-normal distributions having means and standard deviations given by $\mathbb{E}[\epsilon_t] = 1$ and $\mathrm{Var}(\epsilon_t) = \sigma^2 > 1$, respectively. This guarantees that all realizations of the demand process are positive. It is possible to view (3.4) as a linearization of the log-transformed process $\log \mathcal{D}_t$ (cf., [29]). See Section 3.2.1 for a discussion of statistical properties of the process (3.4).

The process (3.4) involves parameters ϕ and μ which are supposed to be estimated from the data. As such, these parameters are subject to estimation errors. This raises the question of sensitivity of the optimal value $\vartheta = \vartheta(\phi, \mu)$ of the corresponding problem (3.3) viewed as a function of ϕ and μ. To this end, we investigate the calculation of the derivatives $\partial\vartheta(\phi, \mu)/\partial\phi$ and $\partial\vartheta(\phi, \mu)/\partial\mu$. With these derivatives at hand, asymptotic distributions of the estimates of ϕ and μ can be translated into the asymptotics of the optimal value in a straightforward way by application of the Delta Theorem. We refer to Section 5.2 for the corresponding numerical experiments.

3.2.1. Properties of the multiplicative autoregressive process. Consider the multiplicative autoregressive process (3.4). Note that under the specified conditions the demand process is not stationary. Indeed, since the errors ϵ_t are i.i.d and $\mathbb{E}[\epsilon_t] = 1$ we have that $\mathbb{E}[\mathcal{D}_t] = \phi\mathbb{E}[\mathcal{D}_{t-1}] + \mu$ and

$$\begin{array}{lll} \mathrm{Var}(\mathcal{D}_t) & = & \mathbb{E}\left[\mathrm{Var}\big(\varepsilon_t(\phi\mathcal{D}_{t-1} + \mu)|\mathcal{D}_{t-1}\big)\right] + \mathrm{Var}\left[\mathbb{E}(\varepsilon_t(\phi\mathcal{D}_{t-1} + \mu)|\mathcal{D}_{t-1})\right] \\ & = & \mathbb{E}\left[\sigma^2(\phi\mathcal{D}_{t-1} + \mu)^2\right] + \mathrm{Var}(\phi\mathcal{D}_{t-1} + \mu) \\ & = & \sigma^2\mathbb{E}\left[(\phi\mathcal{D}_{t-1} + \mu)^2\right] + \phi^2\mathrm{Var}(\mathcal{D}_{t-1}). \end{array} \tag{3.5}$$

It follows that $\mathbb{E}[\mathcal{D}_t]$ converges to $\mu/(1 - \phi)$ as $t \to \infty$. Suppose, for example, that $\mu = 0$. Then $\mathcal{D}_t = \varepsilon_t\phi\mathcal{D}_{t-1} = \mathcal{D}_0\phi^t\prod_{\tau=1}^{t}\varepsilon_\tau,\ t = 1, ..., T,\ \mathbb{E}[\mathcal{D}_t] = \mathcal{D}_0\phi^t \to 0$, and $\mathrm{Var}(\mathcal{D}_t) = \mathcal{D}_0^2\phi^{2t}[(1 + \sigma^2)^t - 1]$. Therefore if $\phi^2(1 + \sigma^2) < 1$, then $\mathrm{Var}(\mathcal{D}_t) \to 0$; and if $\phi^2(1 + \sigma^2) > 1$, then $\mathrm{Var}(\mathcal{D}_t) \to \infty$ provided $\mathcal{D}_0 > 0$.

4. Dual SDDP. In this section, using the results of Section 2, we discuss an adaptation of the cutting planes approach for the approximation of the value functions of the dual problem, similar to the standard SDDP method and called Dual SDDP. The interested reader can find the implementation of Primal SDDP and all variants of Dual SDDP described in this section at https://github.com/vguigues/Dual_SDDP_Library_Matlab and https://github.com/vguigues/Primal_SDDP_Library_Matlab.

We will make the following assumption.

(A2) Primal problem (2.1) satisfies the RCR assumption.

We first consider the case where only b_t and B_t are random in ξ_t.

4.1. Dual SDDP for problems with uncertainty in b_t and B_t. In Dual SDDP, concave value functions $V_t, t = 2, \ldots, T$, are approximated at the end of iteration k by polyhedral upper bounding functions V_t^k given by:

$$V_t^k(\pi_{t-1}) = \min_{0 \leq i \leq k} \overline{\theta}_t^i + \langle \overline{\beta}_t^i, \pi_{t-1} \rangle \tag{4.6}$$

where $\overline{\theta}_t^i$, $\overline{\beta}_t^i$ are coefficients whose computation is detailed below. The algorithm uses valid upper bounds on the norm of dual optimal solutions:

Lemma 4.1. *Suppose that the optimal value of primal problem (2.1) is finite and that there is $\hat{x} > 0$ feasible for primal problem (2.1). Then for every $t = 1, \ldots, T$, we can find $\underline{\pi}_t, \overline{\pi}_t \in \mathbb{R}^{m_t}$ such that dual problem (2.6) is unchanged (i.e., has the same optimal value) adding box constraints $\underline{\pi}_t \leq \pi_t \leq \overline{\pi}_t$.*

Recall that it is assumed that the number of scenarios is finite and hence problem (2.1) can be viewed as a large linear program. The assumption of existence of feasible $\hat{x} > 0$ means that problem (2.1) possesses a feasible solution with all components being strictly positive. If moreover the equality constraints of problem (2.1) are linearly independent, then this strict feasibility condition implies that the set of optimal solutions of the dual problem (i.e., the set of Lagrange multipliers) is bounded. On the other hand, in the above lemma the linear independence condition is not assumed. A proof of Lemma 4.1 and a way to obtain the corresponding bounds $\underline{\pi}_t, \overline{\pi}_t$ can be found in the Appendix.

As mentioned earlier, a difficulty to solve the dual problem with an SDDP type method is that RCR may not be satisfied by the dual problem, even if RCR holds for the primal. We propose two variants of Dual SDDP to solve the Dual problem even if RCR does not hold for the dual: Dual SDDP with penalizations and Dual SDDP with feasibility cuts.

Dual SDDP with penalizations. Dual SDDP with penalizations is based on the developments of Section 2.2. It introduces slack variables in the constraints which may become infeasible for some past decisions in the subproblems solved in the forward passes of Dual SDDP. Slack variables are penalized in the objective function with sequences $(v_{tk})_k$ of positive penalizing coefficients. Therefore, all subproblems solved in forward and backward passes of this variant of Dual SDDP, called Dual SDDP with penalizations, are always feasible and at iteration k, the method builds polyhedral upper bounding function $\mathcal{V}_t^k$ for $\mathcal{V}_t$ of form (4.6) (see Proposition 4.2). Similarly to SDDP, trial points are generated in a forward pass and cuts for $\mathcal{V}_t$ are computed in a backward pass. The detailed Dual SDDP method with penalizations is as follows.

Initialization. For $t = 2, \ldots, T$, take for $\mathcal{V}_t^0$ an affine upper bounding function for $\mathcal{V}_t$ and $\mathcal{V}_{T+1}^0 \equiv 0$. Set iteration counter k to 1.

Step 1: forward pass of iteration k (computation of dual trial points). For the first stage of the forward pass, we compute an optimal solution π_1^k of

$$
(4.7) \qquad V^{k-1} = \begin{array}{l} \max\limits_{\pi_1} \; b_1^\top \pi_1 + \mathcal{V}_2^{k-1}(\pi_1) \\ \underline{\pi}_1 \leq \pi_1 \leq \overline{\pi}_1. \end{array}
$$

Recall that the optimal value of the first stage problem does not change adding box constraints $\underline{\pi}_1 \leq \pi_1 \leq \overline{\pi}_1$ for appropriate values $\underline{\pi}_1$ and $\overline{\pi}_1$. The introduction of these box constraints ensures that the optimal value of (4.7) (which is an *approximate* first stage problem due to the approximation of $\mathcal{V}_2$ by $\mathcal{V}_2^{k-1}$) is finite for all iterations.

For stage $t = 2, \ldots, T-1$, given π_{t-1}^k, we compute an optimal solution of

$$
(4.8) \qquad \begin{array}{ll} \max\limits_{\pi_{t1}, \ldots, \pi_{tN_t}, \zeta_t \geq 0} & \sum\limits_{j=1}^{N_t} p_{tj} \left[b_{tj}^\top \pi_{tj} + \mathcal{V}_{t+1}^{k-1}(\pi_{tj}) \right] - v_{tk}^\top \zeta_t \\ \text{s.t.} & A_{t-1}^\top \pi_{t-1}^k + \sum\limits_{j=1}^{N_t} p_{tj} B_{tj}^\top \pi_{tj} \leq c_{t-1} + \zeta_t, \\ & \underline{\pi}_t \leq \pi_{tj} \leq \overline{\pi}_t. \end{array}
$$

An optimal solution of the problem above has N_t components $(\pi_{t1}, \pi_{t2}, \ldots, \pi_{tN_t})$ for π_t. We generate a realization $\tilde{\xi}_t^k$ of $\xi_t^k \sim \xi_t$ independently of previous realizations $\tilde{\xi}_2^1, \ldots, \tilde{\xi}_{T-1}^1, \ldots, \tilde{\xi}_2^k, \ldots, \tilde{\xi}_{t-1}^k$, and take $\pi_t^k = \pi_{tj_t(k)}$ where index $j_t(k)$ satisfies $\tilde{\xi}_t^k = \xi_{tj_t(k)}$.

Step 2: backward pass of iteration k (computation of new cuts). We first compute a new cut for V_T. Let $(\alpha, \delta, \overline{\Psi}, \underline{\Psi})$ be an optimal solution of[3]

$$
\begin{array}{l}
\displaystyle\min_{\alpha,\delta,\overline{\Psi},\underline{\Psi}} \ \delta^\top (c_{T-1} - A_{T-1}^\top \pi_{T-1}^k) + c_T^\top \sum_{j=1}^{N_T} \alpha_j + \sum_{j=1}^{N_T} \overline{\Psi}_j^\top \overline{\pi}_T - \sum_{j=1}^{N_T} \underline{\Psi}_j^\top \underline{\pi}_T \\
A_T \alpha_j + p_{Tj} B_{Tj} \delta - \underline{\Psi}_j + \overline{\Psi}_j = p_{Tj} b_{Tj}, \ j = 1, \ldots, N_T, \\
0 \le \delta \le \upsilon_{Tk}, \alpha_j, \underline{\Psi}_j, \overline{\Psi}_j \ge 0 \ j = 1, \ldots, N_T.
\end{array}
\tag{4.9}
$$

The new cut for V_T has coefficients given by

$$
\overline{\theta}_T^k = \delta^\top c_{T-1} + c_T^\top \sum_{j=1}^{N_T} \alpha_j + \sum_{j=1}^{N_T} \overline{\Psi}_j^\top \overline{\pi}_T - \sum_{j=1}^{N_T} \underline{\Psi}_j^\top \underline{\pi}_T, \ \overline{\beta}_T^k = -A_{T-1}\delta.
$$

For $t = T-1, \ldots, 2$, compute an optimal solution $(\delta, \nu, \overline{\Psi}, \underline{\Psi})$ of

$$
\begin{array}{l}
\displaystyle\min_{\delta,\nu,\overline{\Psi},\underline{\Psi}} \ \delta^\top \left[c_{t-1} - A_{t-1}^\top \pi_{t-1}^k \right] + \sum_{i=0}^{k} \overline{\theta}_{t+1}^i \sum_{j=1}^{N_t} \nu_i(j) + \sum_{j=1}^{N_t} \overline{\Psi}_j^\top \overline{\pi}_t - \sum_{j=1}^{N_t} \underline{\Psi}_j^\top \underline{\pi}_t \\
\displaystyle p_{tj} B_{tj} \delta - \sum_{i=0}^{k} \nu_i(j) \overline{\beta}_{t+1}^i - \underline{\Psi}_j + \overline{\Psi}_j = p_{tj} b_{tj}, \ j = 1, \ldots, N_t, \\
\displaystyle\sum_{i=0}^{k} \nu_i(j) = p_{tj}, \underline{\Psi}_j, \overline{\Psi}_j \ge 0, \ j = 1, \ldots, N_t, \\
\nu_0, \ldots, \nu_k \ge 0, 0 \le \delta \le \upsilon_{tk},
\end{array}
\tag{4.10}
$$

and the cut coefficients

$$
\overline{\theta}_t^k = \delta^\top c_{t-1} + \sum_{i=0}^{k} \overline{\theta}_{t+1}^i \sum_{j=1}^{N_t} \nu_i(j) + \sum_{j=1}^{N_t} \overline{\Psi}_j^\top \overline{\pi}_t - \sum_{j=1}^{N_t} \underline{\Psi}_j^\top \underline{\pi}_t, \ \overline{\beta}_t^k = -A_{t-1}\delta.
$$

Step 3: Do $k \leftarrow k+1$ and go to Step 1.

The validity of the cuts computed in the backward pass of Dual SDDP with penalizations is shown in Proposition 4.2.

Proposition 4.2. *Consider Dual SDDP algorithm with penalizations. Let Assumptions (A1) and (A2) hold. Then for every $t = 2, \ldots, T$, the sequence V_t^k is a nonincreasing sequence of upper bounding functions for V_t, i.e., for every $k \ge 1$ we have $V_t \le V_t^k \le V_t^{k-1}$ and therefore (V^k) (recall that V^{k-1} is the optimal value of (4.7)) is a nonincreasing deterministic sequence of upper bounds on the optimal value of (2.1).*

To understand the effect of the sequence of penalizing parameters (υ_{tk}) on Dual SDDP with penalizations, we define the following Dynamic Programming equations (see also

[3] We suppressed the dependence of the optimal solution on T and k to alleviate notation.

Lemma 6.1 in the Appendix):

(4.11)

$$V_T^\gamma(\pi_{T-1}) = \begin{cases} \max\limits_{\pi_{T1},\ldots,\pi_{TN_T},\zeta_T \geq 0} & \sum\limits_{j=1}^{N_T} p_{Tj} b_{Tj}^\top \pi_{Tj} - \gamma \mathbf{e}^\top \zeta_T \\ \text{s.t.} & A_{Tj}^\top \pi_{Tj} \leq c_{Tj},\ j=1,\ldots,N_T, \\ & A_{T-1}^\top \pi_{T-1} + \sum\limits_{j=1}^{N_T} p_{Tj} B_{Tj}^\top \pi_{Tj} \leq c_{T-1} + \zeta_T, \end{cases}$$

for $t = 2, \ldots, T-1$:

$$V_t^\gamma(\pi_{t-1}) = \begin{cases} \max\limits_{\pi_{t1},\ldots,\pi_{tN_t},\zeta_t \geq 0} & \sum\limits_{j=1}^{N_t} p_{tj} \left[b_{tj}^\top \pi_{tj} + V_{t+1}^\gamma(\pi_{tj}) \right] - \gamma \mathbf{e}^\top \zeta_t \\ \text{s.t.} & A_{t-1}^\top \pi_{t-1} + \sum\limits_{j=1}^{N_t} p_{tj} B_{tj}^\top \pi_{tj} \leq c_{t-1} + \zeta_t, \end{cases} \tag{4.12}$$

and we define the first stage problem

$$\max_{\pi_1} \ \pi_1^\top b_1 + V_2^\gamma(\pi_1), \tag{4.13}$$

where $\mathbf{e}$ is a vector of ones and γ is a positive real number. As we will see below, V_t^γ can be seen as an upper bounding concave approximation of V_t which gets "closer" to V_t when γ increases. For inventory problem (3.3), it is easy to see that functions V_t in DP equations (2.5), (2.6), (2.7) and functions V_t^γ in DP equations (4.11), (4.12), (4.13) (obtained using in these equations data c_t, b_t A_t, B_t, corresponding to the inventory problem) are only functions of one-dimensional state variable π_{t-1}. Therefore, Dynamic Programming can be used to solve these Dynamic Programming equations and obtain good approximations of functions V_t and V_t^γ. To obtain these approximations, we need to obtain approximations of the domains of functions V_t and compute approximations of these functions on a set of points in that domain. To observe the impact of penalizing term γ on V_t^γ, we run Dynamic Programming both on DP equations (2.5), (2.6), (2.7) and on DP equations (4.11), (4.12), (4.13) for $\gamma = 1$, 100, and 1000, on an instance of the inventory problem with $T = 20$ and $N_t = 20$. The corresponding graphs of V_2 (bold dark solid line) and of V_2^γ for $\gamma = 1, 10$, 1000, are represented in Figure 1. We observe that all functions V_2^γ are, as expected, concave upper bounding functions for V_2 finite everywhere. We also see that on the domain of V_2, V_2^γ gets closer to V_2 when γ increases and eventually coincides with V_2 on this domain when γ is sufficiently large. Similar graphs were observed for remaining functions V_t, V_t^γ, $t = 3, \ldots, T$. Therefore, convergence of Dual SDDP with penalizations requires the coefficients v_{tk} to become arbitrarily large. Proof of the following theorem is given in the Appendix.

Theorem 4.3. *Consider optimization problem* (2.1) *and Dual SDDP with penalizations applied to the dual of this problem. Let Assumptions (A1) and (A2) hold. Assume that samples ξ_t^ℓ, $t = 2, \ldots, T$, $\ell \geq 1$, in the forward passes are independent, that $v_{tk+1} \geq v_{tk}$ for all t, k, and that $\lim_{k \to +\infty} v_{tk} = +\infty$ for all stage t. Then the sequence V^k is a deterministic sequence of upper bounds on the optimal value of* (2.1) *which converges almost surely to the optimal value of this problem.*

Dual SDDP with feasibility cuts. For dual problems not satisfying the RCR assumption, a subproblem for a given stage t in the forward pass can be infeasible. In this situation, as was done in Section 5 of [10] for SDDP, we can build a feasibility cut

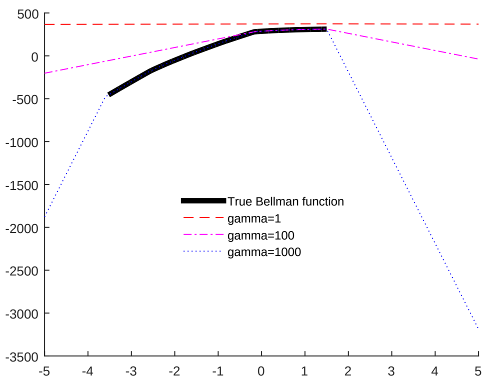

FIG. 1. *Graph of* V_2 *and of* V_2^γ *for* $\gamma = 1, 100, 1000$.

for stage $t-1$ and go back to the previous stage $t-1$ to resolve the problem with that feasibility cut added, and so on until a sequence of feasible states is obtained for all stages. In this context, no penalized slack variables are used, neither in the forward nor in the backward pass. Since the adaptations from [10] are simple, we skip the details of the derivations of this SDDP method applied to the dual. It will be tested in the numerical experiments of Section 5.

4.2. Dual SDDP for problems with uncertainty in all parameters. We have seen in Section 2.1 how to write DP equations on the dual problem of a MSLP when all data (A_t, B_t, c_t, b_t) in (ξ_t) is random. In this situation, cost-to-go functions V_t are functions $V_t(\pi_{t-1}, \xi_{t-1})$ of both past decision π_{t-1} and past value ξ_{t-1} of process (ξ_t). Also recall that functions $V_t(\cdot, \xi_{t-1})$ are concave for all ξ_{t-1}. Therefore, Dual SDDP with penalizations from the previous section must be modified as follows. For each stage $t = 2, \ldots, T$, instead of computing just one approximation of a single function (function V_t), we now need to compute approximations of N_t functions, namely concave cost-to-go functions $V_t(\cdot, \xi_{t-1j})$, $j = 1, \ldots, N_t$. The approximation V_{tj}^k computed for $V_t(\cdot, \xi_{t-1j})$ at iteration k is a polyhedral function V_{tj}^k given by:

$$V_{tj}^k(\pi_{t-1}) = \min_{0 \le i \le k} \overline{\theta}_{tj}^i + \langle \overline{\beta}_{tj}^i, \pi_{t-1} \rangle.$$

Therefore more computational effort is needed. However, the adaptations of the method can be easily written. More specifically, at iteration k, in the forward pass, dual trial points are obtained replacing $V_t(\cdot, \xi_{t-1j})$ by V_{tj}^{k-1} and in the backward pass a cut is computed at stage t for $V_t(\cdot, \xi_{t-1j_k})$ with j_k satisfying $\xi_{t-1j_k} = \tilde{\xi}_{t-1}^k$ where $\tilde{\xi}_{t-1}^k$ is the sampled value of ξ_{t-1} at iteration k.

4.3. Dual SDDP for problems with interstage dependent cost coefficients. We consider problems of form (2.1) where costs c_t affinely depend on their past while b_t are stagewise independent. Specifically, similar to derivations of Section 3.2, suppose that c_t follow a multiplicative vector autoregressive process of form

$$c_t = \varepsilon_t \circ \left(\sum_{j=1}^p \Phi_{tj} c_{t-j} + \mu_t \right), \tag{4.14}$$

with $(x \circ y)_i = x_i y_i$ denoting the componentwise product, and where matrices Φ_{tj} and vectors $\mu_t \ge 0$ as well as $c_1, \ldots, c_{2-p} \ge 0$ are given.

We assume that the process (b_t, ε_t) is stagewise independent and that the support of b_t, ε_t is the finite set

$$\{(b_{t1}, \varepsilon_{t1}), \ldots, (b_{tN_t}, \varepsilon_{tN_t})\},$$

with $\varepsilon_{ti} > 0$ and $p_{ti} = \mathbb{P}\{(b_t, \varepsilon_t) = (b_{ti}, \varepsilon_{ti})\}$, $i = 1, \ldots, N_t$. For some values of Φ_{tj} (for instance for matrices with nonnegative entries), this guarantees that all realizations of the price process $\{c_t\}$ are positive. The developments which follow can be easily extended to other linear models for $\{c_t\}$, for instance SARIMA or PAR models, see [9] for the definition of state vectors of minimal size for generalized linear models.

Using the notation $c_{t_1:t_2} = (c_{t_1}, c_{t_1+1}, \ldots, c_{t_2-1}, c_{t_2})$ for $t_1 \leq t_2$ integer, for the corresponding primal problem (of the form (2.1)), we can write the following Dynamic Programming equations: define $\mathcal{Q}_{T+1} \equiv 0$ and for $t = 2, \ldots, T$,

$$\mathcal{Q}_t(x_{t-1}, c_{t-p:t-1}) = \mathbb{E}_{b_t, \varepsilon_t}\Big[Q_t(x_{t-1}, c_{t-p:t-1}, b_t, \varepsilon_t)\Big] \tag{4.15}$$

where $Q_t(x_{t-1}, c_{t-p:t-1}, b_t, \varepsilon_t)$ is given by

(4.16)

$$\begin{array}{l}\min\limits_{x_t \geq 0} \Big[\varepsilon_t \circ \Big(\sum\limits_{j=1}^{p} \Phi_{tj} c_{t-j} + \mu_t\Big)\Big]^\top x_t + \mathcal{Q}_{t+1}\Big(x_t, c_{t+1-p:t-1}, \varepsilon_t \circ \Big(\sum\limits_{j=1}^{p} \Phi_{tj} c_{t-j} + \mu_t\Big)\Big)\\ A_t x_t + B_t x_{t-1} = b_t,\end{array}$$

while the first stage problem is

$$\begin{array}{l}\min\limits_{x_1 \geq 0} \ c_1^\top x_1 + \mathcal{Q}_2(x_1, c_{2-p:1})\\ A_1 x_1 = b_1.\end{array}$$

Standard SDDP does not apply directly to solve Dynamic Programming equations (4.15)-(4.16) because functions $\mathcal{Q}_t$ given by (4.15)-(4.16) are not convex. Nevertheless, we can use the Markov Chain discretization variant of SDDP to solve Dynamic Programming equations (4.15)-(4.16). On the other hand, as pointed above, it is possible to apply SDDP for the dual problem with the added state variables. Along the lines of Section 2.1 we can write Dynamic Programming equations for the dual, now with function V_t depending on $\pi_{t-1}, c_{t-1}, \ldots, c_{t-p}$.

These functions are concave and therefore we can apply Dual SDDP with penalizations to these DP equations to build polyhedral approximations of these functions V_t of form

$$V_t^k(\pi_{t-1}, c_{t-1}, \ldots, c_{t-p}) = \min_{0 \leq i \leq k} \theta_t^i + \langle \beta_{t0}^i, \pi_{t-1} \rangle + \sum_{j=1}^{p} \langle \beta_{tj}^i, c_{t-j} \rangle \tag{4.17}$$

at iteration k.

5. Numerical experiments.

In this section, we report numerical results obtained applying Primal SDDP and variants of Dual SDDP to the inventory problem and to the Brazilian interconnected power system problem. All methods were implemented in Matlab and run on an Intel Core i7, 1.8GHz, processor with 12,0 Go of RAM. Optimization problems were solved using Mosek [1].

5.1. Dual SDDP for the inventory problem.

We consider the inventory problem (3.3) with parameters $a_t = 1.5 + \cos(\frac{\pi t}{6})$, $p_{ti} = \frac{1}{N}$ where N is the number of realizations for each stage, $\xi_{tj} = (5 + 0.5t)(1.5 + 0.1 z_{tj})$ where $(z_{t1}, \ldots, z_{tN})$ is a sample from the standard Gaussian distribution, $x_0 = 10$, $g_t = 2.8$, and $h_t = 0.2$.

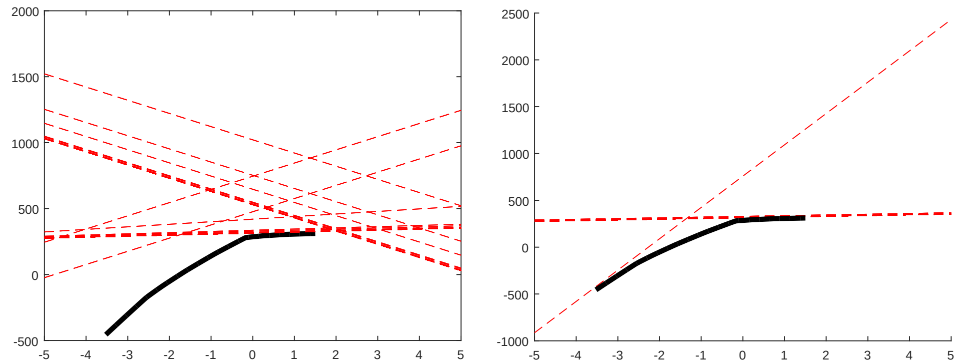

FIG. 2. *Graph of V_2 (bold black solid line) and cuts computed for V_2 by Dual SDDP with penalizations $v_{tk} = 100$ (left panel) and Dual SDDP with feasibility cuts (right panel).*

Illustrating the correctness of DP equations (2.5), (2.6), (2.7) and checking the convergence of the variants of Dual SDDP. We solve this inventory problem using Dynamic Programming applied both to DP equations (2.5), (2.6), (2.7) and to DP equations (4.11)-(4.12) for $\gamma = 1, 10, 1000$. In this latter case, we obtain approximations of functions V_t^γ. We also run Primal SDDP, Dual SDDP with feasibility cuts, and Dual SDDP with penalties $v_{tk} = 1$, 10, 1000, on the same instance, knowing that Dual SDDP variants were run for 100 iterations (the upper bounds computed by these methods stabilize in less than 10 iterations) and Primal SDDP was stopped when the gap is < 0.1 where the gap is defined as $\frac{Ub-Lb}{Ub}$ where Ub and Lb correspond to upper and lower bounds computed by Primal SDDP along iterations. The lower bound Lb is the optimal value of the first stage problem and the upper bound Ub is the upper end of a 97.5%-one-sided confidence interval on the optimal value obtained using the sample of total costs computed by all previous forward passes. With this stopping criterion and the considered instance of the inventory problem, Primal SDDP was run for 232 iterations.

In Figure 2, we report the graph of V_2 and the cuts computed for V_2 by Dual SDDP with feasibility cuts (right panel) and Dual SDDP with penalties $v_{tk} = 100$ (left panel). All cuts are, as expected, upper bounding affine functions for V_2 on its domain. However, it is interesting to notice that for Dual SDDP with feasibility cuts, few different cuts are computed and these cuts are tangent or very close to V_2 at the trial points. On the contrary, Dual SDDP with penalties may compute many cuts dominated by others on the domain of V_2. Therefore, cut selection techniques, for instance along the lines of [11] [13] using Limited Memory Level 1 cut selection, could be useful for Dual SDDP.

We report in Table 1 the approximate optimal values and the time needed to compute them with Primal SDDP, Dual SDDP, and Dynamic Programming applied to respectively (2.5), (2.6), (2.7) and (4.11), (4.12), (4.13) with $\gamma = 1, 100, 1000$. The approximate optimal values reported are the last upper bound computed for variants of Dual SDDP and the last lower bound computed for Primal SDDP. All approximate optimal values are very close (showing that all variants were correctly implemented) and Dynamic Programming is much slower than the other sampling-based algorithms. For Dual SDDP with penalization, if penalties are too small the upper bound can be $+\infty$ while if penalties are sufficiently large the algorithm converges to an optimal policy.

Method	Optimal value	CPU time (s.)
DP on (2.5), (2.6), (2.7)	321.6	685
DP on (4.11), (4.12), (4.13), $\gamma = 1$	$+\infty$	2 860
DP on (4.11), (4.12), (4.13), $\gamma = 100$	322.2	3 808
DP on (4.11), (4.12), (4.13), $\gamma = 1000$	321.8	3 376
Primal SDDP	322.5	105
Dual SDDP with penalties, $v_{tk} = 1$	2 131.4	9.4
Dual SDDP with penalties, $v_{tk} = 100$	322.5	11.3
Dual SDDP with penalties, $v_{tk} = 1000$	322.5	11.9
Dual SDDP with feasibility cuts	322.5	10.6

TABLE 1

Optimal value and CPU time needed (in seconds) to compute them on an instance of the inventory problem with $T = N_t = 20$ by Dynamic Programming (DP), Primal SDDP, and variants of Dual SDDP.

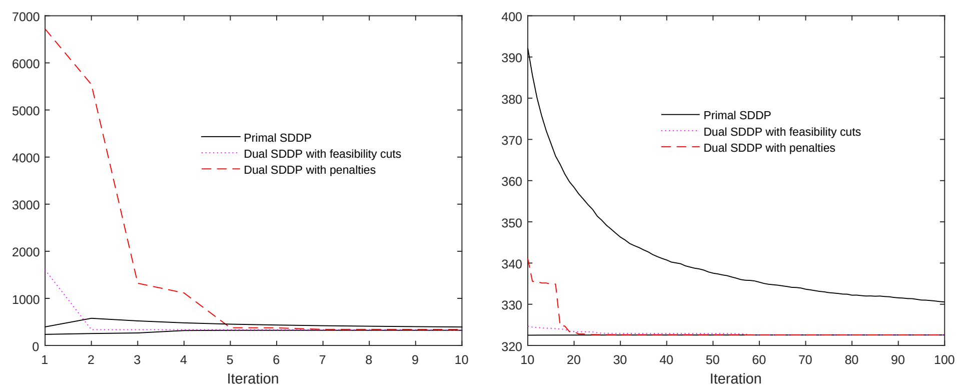

FIG. 3. *Left: upper and lower bounds computed by Primal SDDP and upper bounds computed by Dual SDDP with feasibility cuts and Dual SDDP with penalties $v_{tk} = 1000$ for the first 10 iterations. Right: same outputs for iterations 10,$\ldots, 100$.*

Finally, we report for this instance in Figure 3 the evolution of the lower bound Lb and upper bound Ub computed by Primal SDDP and the upper bounds computed by Dual SDDP with penalties $v_{tk} = 1000$ and Dual SDDP with feasibility cuts. With Dual SDDP, the upper bound is naturally large at the first iteration but decreases much quicker than the upper bound Ub computed by Primal SDDP, especially for Dual SDDP with feasibility cuts, with all upper bounds converging to the optimal value of the problem.

Tests on a larger instance. We now run Primal and Dual SDDP on a larger instance with $T = 100$ and $N_t = 100$ for 600 iterations. The evolution of the upper bounds computed along the iterations of Dual SDDP (both with feasibility cuts and with penalizations $v_{tk} = 1000$) and of the upper and lower bounds computed by Primal SDDP are reported in Table 2 for iterations $2, 3, 5, 10, 50, 100, 200, 300, 400, 500$, and 600. We see that for the first iterations, the upper bound decreases more quickly with the variants of Dual SDDP, the most important decrease being obtained for Dual SDDP with feasibility cuts. However, on this instance, the convergence of Dual SDDP with feasibility cuts is slower, i.e., a solution of high accuracy is obtained quicker using Dual SDDP with penalizations. More precisely, we fix confidence levels $\varepsilon = 0.2, 0.15, 0.1, 0.05, 0.01$, and for each confidence level, we compute the time needed, running Primal and Dual SDDP in parallel, to obtain a solution with relative accuracy ε stopping the algorithm when the upper bound `Ub_D` computed by a variant of Dual SDDP and the lower bound Lb, computed by Primal SDDP, satisfies (`Ub_D`-Lb)/`Ub_D`$<\varepsilon$. The results are reported in Table 3. In this table, we also report the time needed to obtain a solution of relative accuracy ε using only the information provided by Primal SDDP, stopping the algorithm when (Ub-Lb)/Ub$<\varepsilon$.

We observe that if ε is not too small, the smallest CPU time is obtained combining Primal SDDP with Dual SDDP with feasibility cuts while when ε is small (0.05 and 0.01) the smallest CPU time is obtained combining Primal SDDP with Dual SDDP with penalizations. For $\varepsilon = 0.05$ and 0.01, 600 iterations are even not enough to get a solution of relative accuracy ε using Primal SDDP or combining Primal SDDP and Dual SDDP with feasibility cuts.

Iteration	`Primal SDDP` Lb	`Primal SDDP` Ub	`Dual SDDP with feasibility cuts`	`Dual SDDP with penalties`
2	656.4	25 443	20 002	20 015
3	713.1	19 340	8 693.1	20 012
5	3361.8	14 800	7 246.8	19 993
10	5330.1	10 662	5 736.6	16 452
50	5483.1	6 594.5	5721.8	5500.9
100	5483.5	6 039.2	5715.1	5484.8
200	5483.6	5 762.4	5710.0	5484.2
300	5483.7	5 671.0	5704.6	5484.0
400	5483.7	5 625.3	5702.7	5483.9
500	5483.7	5 597.9	5702.5	5483.8
600	5483.7	5 579.9	5702.2	5483.8

TABLE 2

For an instance of the inventory problem with $T = N_t = 100$, lower bound Lb and upper bound Ub computed by Primal SDDP and upper bounds computed by Dual SDDP with feasibility cuts and Dual SDDP with penalties $v_{tk} = 1000$ along iterations.

In Figure 4, we report the cumulative CPU time along iterations of all methods. We see that each iteration requires a similar computational bulk and the CPU time increases exponentially with the number of iterations.

5.2. Sensitivity analysis for the inventory problem. Consider the inventory problem of Section 5.1 with $(\mathcal{D}_t)$ as in (3.4) and $T = 10$ stages. For this problem, the derivatives from Proposition 3.1 are given by

$$\partial\vartheta(\phi,\mu)/\partial\phi = \partial L(\bar{x},\bar{y},\bar{\pi})/\partial\phi = \mathbb{E}\left[\sum_{t=1}^{T}\bar{\pi}_t\epsilon_t\mathcal{D}_{t-1}\right], \tag{5.1}$$

$$\partial\vartheta(\phi,\mu)/\partial\mu = \partial L(\bar{x},\bar{y},\bar{\pi})/\partial\mu = \mathbb{E}\left[\sum_{t=1}^{T}\bar{\pi}_t\epsilon_t\right], \tag{5.2}$$

where $(\bar{x},\bar{y})$ is an optimal solution of the primal problem and $\bar{\pi}$ are the corresponding Lagrange multipliers. Our goal is to compute these derivatives solving the primal and dual problems by respectively Primal and Dual SDDP.

ε	`Primal SDDP`	`Dual SDDP with feasibility cuts`	`Dual SDDP with penalties` $v_{tk} = 1000$
0.2	300.2	29.5	35.8
0.15	459.8	35.8	41.2
0.1	825.6	48.3	48.3
0.05	2366.2	96.1	61.5
0.01	-	-	103.2

TABLE 3

Time needed (in seconds) to obtain a solution of relative accuracy ε with Primal SDDP, Dual SDDP with feasibility cuts, and Dual SDDP with penalties $v_{tk} = 1000$ for an instance of the inventory problem with $T = N_t = 100$.

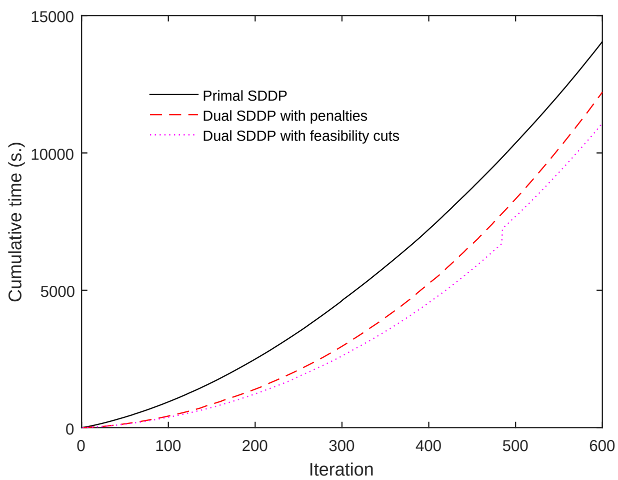

FIG. 4. *Cumulative CPU time along iterations of Primal SDDP, Dual SDDP with feasibility cuts, and Dual SDDP with penalizations $v_{tk} = 1000$.*

We consider 4 instances with $(\phi, \mu) = (0.01, 0.1)$, $(0.01, 3.0)$, $(0.001, 0.1)$, and $(0.001, 3.0)$. The remaining parameters of these instances are those from the previous section. We discretize both the primal and dual problem into $N_t = 100$ samples for each stage $t = 2, \ldots, 10$. We take the relative error $\varepsilon = 0.01$ for the stopping criterion and use 10 000 Monte Carlo simulations to estimate the expectations in (5.1), (5.2). For Primal SDDP, the upper bound Ub and lower bound Lb at termination are given in Table 4 for the four instances.

Bound	`Instance 1`	`Instance 2`	`Instance 3`	`Instance 4`
Ub	17.9176	478.687	15.3940	404.242
Lb	17.9163	475.017	15.3927	402.913

TABLE 4

Upper and lower bounds at the last iteration of Primal SDDP.

The optimal mean values of Lagrangian multipliers for the demand constraints computed, for a given stage $t \geq 2$, averaging over the 10 000 values obtained simulating 10 000 forward passes after termination, are given in Table 5. In this table, `LM` stands for the multipliers obtained using Primal SDDP as explained in Remark 2.1 whereas `Dual` stands for the multipliers obtained using Dual SDDP with penalties. The fact that the multipliers obtained are close for both methods illustrates the validity of the two alternatives we discussed in Sections 3-4 to compute derivatives of the value function of a MSP.

Stage	Instance 1		Instance 2		Instance 3		Instance 4	
	LM	Dual	LM	Dual	LM	Dual	LM	Dual
2	0.2465	0.2373	1.6701	1.66959	0.0444	0.0328	1.666	1.666
3	0.3218	0.31095	1.4098	1.4120	0.1421	0.1340	1.406	1.409
4	0.3268	0.3221	0.9862	0.9861	0.19439	0.18974	0.984	0.984
5	0.3086	0.3058	0.6330	0.6329	0.2145	0.2128	0.6327	0.6327
6	0.3408	0.3412	0.49998	0.499897	0.2708	0.2717	0.4999	0.4998
7	0.5026	0.5051	0.63397	0.63397	0.4378	0.4418	0.6339	0.6339
8	0.7047	0.7049	0.8348	0.8340	0.6404	0.6413	0.8349	0.8334
9	0.8985	0.9032	1.0322	1.0343	0.83501	0.8401	1.0315	1.0343
10	1.1022	1.1037	1.2302	1.2365	1.03926	1.04091	1.23	1.23

TABLE 5

Comparison between optimal Lagrange multipliers from Primal SDDP and Dual SDDP with penalties.

With optimal dual solutions $\{\bar{\pi}_t\}$ and the realizations of $\{D_t\}$ and $\{\epsilon_t\}$ at hand, we are able to compute the sensitivity of the optimal value with respect to ϕ and μ, using (5.1) and (5.2), with expectations estimated for 10 000 Monte Carlo simulations. We benchmark our method against the finite-difference method. Specifically, for value function ϑ, the finite-difference method approximates the derivative with respect to u_0 by $v'(u_0) \approx \frac{v(u_0+\delta)-v(u_0-\delta)}{2\delta}$ for some small δ.

The sensitivity of the optimal value of the inventory problem with respect to (ϕ, μ) is displayed in Table 6. In this table, S-ϕ and S-μ denote the derivatives with respect to ϕ and μ computed by our method, and fd-ϕ, fd-μ denote the derivatives computed by the finite-difference method. In order to measure the difference between the two methods, we also compute S-gap-ϕ and S-gap-μ, where S-gap-$\phi := \frac{|\text{fd-}\phi - \text{S-}\phi|}{|\text{fd-}\phi|} \times 100\%$ and S-gap-$\mu := \frac{|\text{fd-}\mu - \text{S-}\mu|}{|\text{fd-}\mu|} \times 100\%$.

Instance	fd-ϕ	S-ϕ	S-gap-ϕ(%)	fd-μ	S-μ	S-gap-μ(%)
1	403.604	401.094	0.622	164.578	164.158	0.255
2	10 716.111	10 671.262	0.419	185.346	184.847	0.270
3	269.514	269.443	0.026	134.646	134.463	0.136
4	7 780.570	7 770.274	0.132	158.017	158.001	0.0101

TABLE 6

Sensitivity of the optimal value with respect to ϕ and μ by the two methods.

We observe that the derivatives obtained by both methods are close to each other, especially when ϕ and μ are small. This is because small ϕ and μ gives rise to less variability in the demand. Note also that the finite-difference method is more time consuming since it requires computing the optimal value twice. Instead, our method only needs to solve the model once. Moreover, computing the Lagrange multipliers does not significantly consume CPU time, as they are generated as a by-product of Primal SDDP. Alternatively, as discussed above, one can compute the optimal multipliers using Dual SDDP with penalties. Another drawback of the finite-difference method lies in its numerical instability. Indeed, the method is more accurate when δ

is very small. However, the division by a very small number generates bias while our approach is more stable.

5.3. Dual SDDP for an hydro-thermal generation problem. We repeat the experiments of Section 5.1 for the Brazilian interconnected power system problem discussed in [7] for $T = 12$ stages and $N_t = 50$ inflow realizations for every stage. These realizations are obtained calibrating log-normal distributions for each month of the year using historical data of inflows and sampling from these distributions. The data used for these simulations (including the inflow scenarios) is available on Github[4].

We solve this problem using Primal SDDP and Dual SDDP with penalizations. For this variant of Dual SDDP, a general procedure to define sequences of penalizations (v_{tk}) ensuring convergence of the corresponding Dual SDDP method is to take $v_{tk} = \gamma_0 \alpha^{k-1}\mathbf{e}$, $k \geq 1$, $t = 2, \ldots, T$, with $\alpha > 1$, $\gamma_0 > 0$. For numerical reasons, we also take a large upper bound U for these sequences and use

$$v_{tk} = \min(U, \gamma_0 \alpha^{k-1})\mathbf{e},\ k \geq 1, t = 2, \ldots, T. \tag{5.3}$$

We consider three variants of Dual SDDP: for the first variant, denoted by `Dual SDDP 1`, v_{tk} are as in (5.3) with $\gamma_0 = 10^4$, $\alpha = 1.3$, $U = 10^{10}$. To illustrate the fact that for constant sequences $v_{tk} = \gamma_0$, Dual SDDP converges (resp. does not converge) for sufficiently large constants γ_0 (resp. sufficiently small constants γ_0) we also define two other variants corresponding to $U = +\infty$, $\gamma_0 = 10^9$, $\alpha = 1$, and $U = +\infty$, $\gamma_0 = 10^6$, $\alpha = 1$, in (5.3), respectively denoted by `Dual SDDP 2` and `Dual SDDP 3`.

We run Dual SDDP for 1000 iterations and Primal SDDP for 3000 iterations. The evolution of the upper and lower bounds computed by the methods for the first 1000 iterations is given in Figure 5.[5]

More precisely, the values of these bounds for iterations 2, 5, 10, 50, 100, 150, 200, 250, 300, 350, 400, 1000, and 3000 are reported in Table 7. We observe that parameter γ_0 for `Dual SDDP 3` is too small to allow this method to converge to the optimal value of the problem whereas the other two variants `Dual SDDP 1` and `Dual SDDP 2` of Dual SDDP converge. Naturally, these methods start with large upper bounds but after a few tens of iterations the upper bounds with `Dual SDDP 1` and `Dual SDDP 2` are better than the upper bound computed by Primal SDDP. In particular, it is interesting to notice that the best (lowest) upper bounds are obtained with the variant of Dual SDDP that uses adaptive penalizations, i.e., penalizations that increase with the number of iterations before reaching value U in (5.3).

We also report in Table 8 the relative error $\frac{\texttt{Upper}_M(i) - \texttt{Lower}_{\texttt{SDDP}}(i)}{\texttt{Upper}_M(i)}$ for iterations $i = 100$, 200, 300, 400, 500, 800, and 1000 for all methods M where $\texttt{Upper}_M(i)$ and $\texttt{Lower}_{\texttt{SDDP}}(i)$ are respectively the upper bound computed by method M at iteration i and the lower bound computed by Primal SDDP at iteration i. For iterations 300 on, the relative error is much smaller with variants of Dual SDDP, meaning that Primal SDDP overestimates the optimality gap.

However, each iteration of Dual SDDP takes more time as can be seen in Figure 6 which reports the cumulative CPU time for all methods. More precisely, running Dual and Primal SDDP in parallel, we can compute the time needed to obtain a solution of relative accuracy ε using the standard stopping criterion for Primal SDDP (see [27]) or using the lower bound from Primal SDDP and the upper bound from

[4] https://github.com/vguigues/Primal_SDDP_Library_Matlab

[5] The upper bounds for Primal SDDP are computed as explained in Section 5.1.

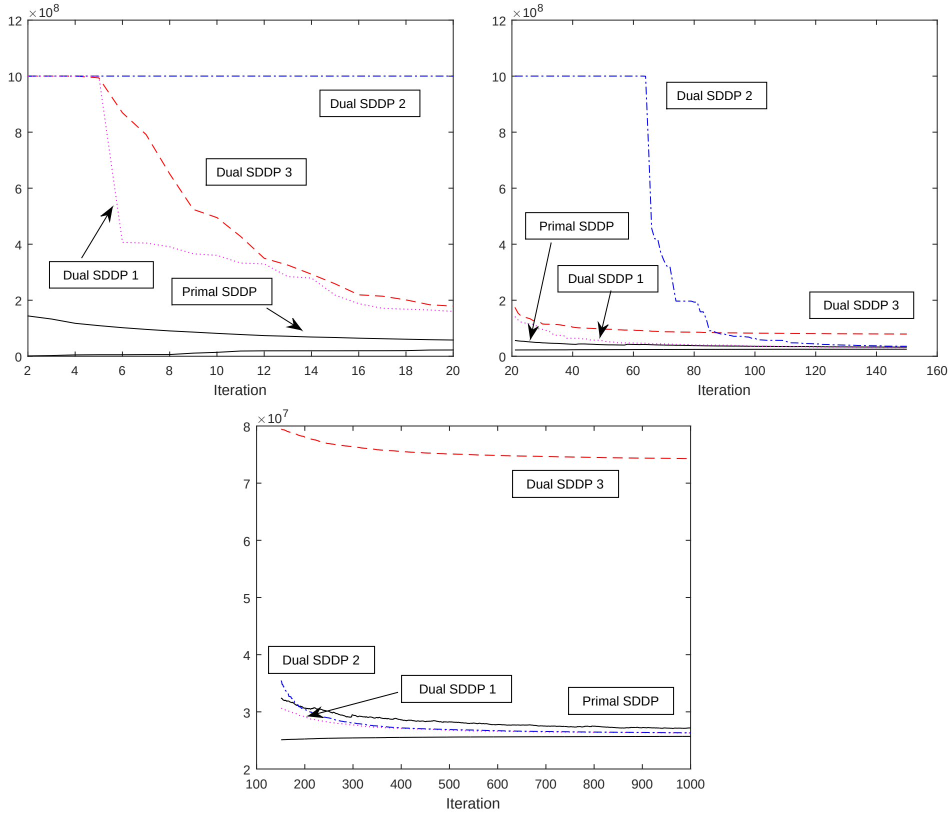

FIG. 5. *Top left: upper and lower bounds computed by Primal SDDP and upper bounds computed by* `Dual SDDP 1`*,* `Dual SDDP 2`*, and* `Dual SDDP 3`*, for the first 20 iterations for an instance of the hydro-thermal problem with* $T = 12$*,* $N_t = 50$*. Top right: same outputs for iterations 21,..., 150. Bottom: same outputs for iterations 151,..., 1000.*

Dual SDDP, and computing the relative error obtained with these bounds each time a new bound (either lower bound or upper bound) is computed. The results are reported in Table 9. We see that due to the fact that Dual SDDP iterations are more time consuming, for all relative accuracies but one, the use of the stopping criterion based on Dual SDDP upper bounds requires more computational bulk. From this experiment, performed on a larger problem (in terms of size of the state vector and number of control variables for each stage) than the inventory problem of Section 5.1, it seems that the use of Dual SDDP for a stopping criterion of Primal SDDP will decrease the overall computational bulk only for small problems (having a limited to small number of controls, state variables, and scenarios).

REFERENCES

[1] E. D. Andersen and K.D. Andersen. *The MOSEK optimization toolbox for MATLAB manual. Version 7.0*, 2013. https://www.mosek.com/.

[2] K.J. Arrow, L. Hurwicz, and H. Uzawa. *Iterative methods for concave programming in Studies in linear and nonlinear programming*. Stanford University Press, 1958.

[3] A. Ben-Tal and A. Nemirovski. Robust Convex Optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[4] D.P. Bertsekas. *Nonlinear Programming, 2nd ed. Belmont, MA.* Athena Scientific, 1999.

Iteration	Primal SDDP Lb	Primal SDDP Ub	Dual SDDP 1	Dual SDDP 2	Dual SDDP 3
2	1.317	143.98	1000.2	1000.2	1000.2
5	5.5588	109.36	1000.2	1000.2	994.04
10	14.032	81.728	360.40	1000.2	495.08
50	23.670	41.346	54.999	1000.2	96.720
100	24.787	35.502	36.322	64.072	82.494
150	25.111	32.447	30.685	35.595	79.465
200	25.249	30.672	29.076	30.404	78.059
250	25.374	30.079	28.215	28.943	76.917
300	25.436	29.434	27.710	28.030	76.344
350	25.477	29.014	27.309	27.532	75.852
400	25.526	28.626	27.110	27.188	75.526
1000	25.703	27.175	26.304	26.335	74.292
3000	25.798	26.883	-	-	

TABLE 7

For an instance of the hydro-thermal problem with $T = 12$, $N_t = 50$, lower bound Lb and upper bound Ub computed by Primal SDDP and upper bounds computed by variants of Dual SDDP along iterations. All costs have been divided by 10^6 to improve readability.

Iteration	Primal SDDP	Dual SDDP 1	Dual SDDP 2
100	0.30	0.32	0.61
200	0.18	0.13	0.17
300	0.14	0.08	0.09
400	0.11	0.06	0.06
500	0.09	0.05	0.05
800	0.07	0.03	0.03
1000	0.05	0.02	0.02

TABLE 8

Relative error as a function of the number of iterations for `Primal SDDP`, `Dual SDDP 1`, *and* `Dual SDDP 2`.

[5] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems.* Springer Series in Operations Research. Springer, New York, 2000.

[6] J.F. Bonnans, Z. Cen, and Th. Christel. *Sensitivity analysis of energy contracts by stochastic programming techniques*, pages 447 – 471. Numerical Methods in Finance, Springer Proceeding in Mathematics 12 (2012).

[7] L. Ding, S. Ahmed, and A. Shapiro. A python package for multi-stage stochastic programming. *Optimization Online*, 2019.

[8] V. Guigues. Inexact Stochastic Mirror Descent for two-stage nonlinear stochastic programs. *Mathematical Programming, to appear.*

[9] V. Guigues. SDDP for some interstage dependent risk-averse problems and application to hydro-thermal planning. *Computational Optimization and Applications*, 57:167–203, 2014.

[10] V. Guigues. Convergence analysis of sampling-based decomposition methods for risk-averse multistage stochastic convex programs. *SIAM Journal on Optimization*, 26:2468–2494, 2016.

[11] V. Guigues. Dual dynamic programing with cut selection: Convergence proof and numerical experiments. *European Journal of Operational Research*, 258:47–57, 2017.

[12] V. Guigues. Inexact cuts in Stochastic Dual Dynamic Programming. *Siam Journal on Optimization*, 30:407–438, 2020.

[13] V. Guigues and M. Bandarra. Single cut and multicut SDDP with cut selection for multistage

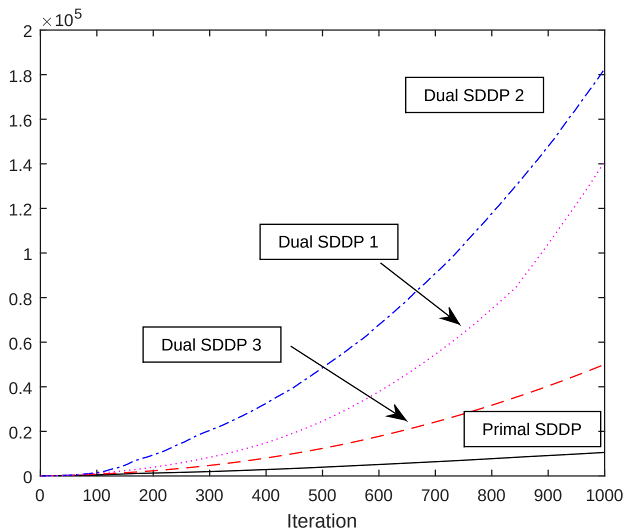

Fig. 6. *Cumulative CPU time for Primal SDDP, Dual SDDP 1, Dual SDDP 2, and Dual SDDP 3.*

ε	Primal SDDP	Dual SDDP 1	Dual SDDP 2
0.3	515	1 042	4 133
0.2	1 167	1 895	7 446
0.15	1 659	2 910	9 882
0.1	3 168	5 114	16 387
0.075	5 359	8 003	22 457
0.05	11 124	15 738	35 113
0.04	45 391	23 449	51 381

Table 9

Time (in seconds) needed to obtain a solution of relative accuracy ε with Primal SDDP and variants of Dual SDDP for an instance of the hydro-thermal problem.

stochastic linear programs: convergence proof and numerical experiments. *arXiv*, 2019. https://arxiv.org/abs/1902.06757.

[14] J.L. Higle and S. Sen. Multistage stochastic convex programs: Duality and its implications. *Annals of Operations Research*, 142(1):129–146, 2006.

[15] J.L. Higle and S. Sen. Multistage stochastic convex programs: Duality and its implications. *Annals of operations research*, 142:129–146, 2006.

[16] H.J. Kushner. On the stochastic maximum principle: Fixed time of control. *J. Math. Anal. Appl.*, 11:78–92, 1965.

[17] H.J. Kushner. On the stochastic maximum principle with average constraints. *J. Math. Anal. Appl.*, 12:13–26, 1965.

[18] H.J. Kushner. Necessary conditions for continuous parameter stochastic optimization problems. *SIAM J. Control*, 10:550–565, 1972.

[19] V. Leclère, P. Carpentier, J-P Chancelier, A. Lenoir, and F. Pacaud. Exact converging bounds for Stochastic Dual Dynamic Programming via Fenchel duality. *SIAM J. Optimization*, 30:1223 – 1250, 2020.

[20] N. Lohndorf and A. Shapiro. Modeling Time-dependent Randomness in Stochastic Dual Dynamic Programming. *European Journal of Operational Research*, 273:650–661, 2019.

[21] A. Nedić and A. Ozdaglar. Subgradient Methods for Saddle-Point Problems. *Journal of Optimization Theory and Applications*, 142(1):205–228, 2009.

[22] M.V.F. Pereira and L.M.V.G Pinto. Multi-stage stochastic optimization applied to energy planning. *Math. Program.*, 52:359–375, 1991.

[23] A. B. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and related methods. *Oper. Res. Lett.*, 36:450–455, 2008.
[24] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis.* Springer Berlin, 1998.
[25] R.T. Rockafellar. Duality and optimality in multistage stochastic programming. *Annals of Operations Resarch*, 85:1–19, 1999.
[26] R.T. Rockafellar. Duality and optimality in multistagestochastic programming. *Annals of Operations Research*, 85(0):1–19, 1999.
[27] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209:63–72, 2011.
[28] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory, second edition.* SIAM, Philadelphia, 2014.
[29] 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(2):375–391, 2013.
[30] G. Terca and D. Wozabal. Envelope theorems for multi-stage linear stochastic optimization. Optimization Online, 2018.
[31] P.H. Zipkin. *Foundations of Inventory Management.* McGraw-Hill, Boston, 2000.

6. Appendix. In this Appendix, we prove Lemma 4.1, Proposition 4.2, and Theorem 4.3.

We first need more notation. We introduce the sequence of functions

$$(6.1)\qquad \overline{V}_T^k(\pi_{T-1}) := \begin{array}{l} \max\limits_{\pi_{T1},\ldots,\pi_{TN_T},\zeta_T} \displaystyle\sum_{j=1}^{N_T} p_{Tj} b_{Tj}^\top \pi_{Tj} - v_{Tk}^\top \zeta_T \\ A_T^\top \pi_{Tj} \le c_T,\ j=1,\ldots,N_T, \\ A_{T-1}^\top \pi_{T-1} + \displaystyle\sum_{j=1}^{N_T} p_{Tj} B_{Tj}^\top \pi_{Tj} \le c_{T-1} + \zeta_T, \\ \zeta_T \ge 0, \underline{\pi}_T \le \pi_{Tj} \le \overline{\pi}_T,\ j=1,\ldots,N_T, \end{array}$$

and for $t = 2,\ldots,T-1$, the sequence of functions

$$(6.2)\qquad \overline{V}_t^k(\pi_{t-1}) := \begin{array}{l} \max\limits_{\pi_{t1},\ldots,\pi_{tN_t},\zeta_t} \displaystyle\sum_{j=1}^{N_t} p_{tj}\left(b_{tj}^\top \pi_{tj} + V_{t+1}^k(\pi_{tj})\right) - v_{tk}^\top \zeta_t \\ A_{t-1}^\top \pi_{t-1} + \displaystyle\sum_{j=1}^{N_t} p_{tj} B_{tj}^\top \pi_{tj} \le c_{t-1} + \zeta_t, \\ \zeta_t \ge 0, \underline{\pi}_t \le \pi_{tj} \le \overline{\pi}_t,\ j=1,\ldots,N_t. \end{array}$$

Due to Assumption (A1) we can represent the scenarios for $\xi_1, \xi_2, \ldots, \xi_T$, by a scenario tree of depth $T+1$ where the root node n_0 associated to a stage 0 (with decision x_0 taken at that node) has one child node n_1 associated to the first stage. We denote by $\mathcal{N}$ the set of nodes and for a node n of the tree, by (x_n, π_n) a primal-dual pair at that node and by ξ_n the realization of process (ξ_t) at node n (this realization ξ_n contains in particular the realizations c_n of c_t, b_n of b_t, A_n of A_t, and B_n of B_t).

Proof of Lemma 4.1. Let $1 \le t \le T$ and let us fix a node m of stage t. Let $\overline{A}_m$ such that constraints $A_m x_m + B_m x_{F(m)} = b_m$ are rewritten in compact form $\overline{A}_m x = b_m$ in terms of vector $x = (x_n)_{n\in\mathcal{N}}$ of decision variables in the scenario tree. The dual function obtained dualizing the coupling constraints of node m is given by

$$\theta(\pi_m) = \begin{array}{l} \min\ \mathbb{E}[c^\top x] + \pi_m^\top (A_m x_m + B_m x_{F(m)} - b_m) \\ x \in \mathcal{S}_m, \end{array}$$

for $\mathcal{S}_m = \{x = (x_n)_{n\in\mathcal{N}} : x \ge 0\} \cap \mathcal{A}_m$ where $\mathcal{A}_m = \{x = (x_n)_{n\in\mathcal{N}} : A_n x_n + B_n x_{F(n)} = b_n, \forall n \ne m, n \in \mathcal{N}\}$.

By Linear Programming Duality, the optimal value $\mathcal{Q}_1(x_0)$ of primal problem (2.1) is the optimal value of the dual problem

$$\max\{\theta(\pi_m) : \pi_m \in \mathbb{R}^{m_t}\} \tag{6.3}$$

which can clearly be written as

$$\mathcal{Q}_1(x_0) = \max_{\pi_m}\{\theta(\pi_m) : \pi_m = \overline{A}_m x - b_m, x \in \mathrm{Aff}(\mathcal{S}_m)\}, \tag{6.4}$$

where $\mathrm{Aff}(\mathcal{S}_m)$ is the affine hull of $\mathcal{S}_m$. We now bound the optimal solutions of dual problem (6.4). Since (6.3) and (6.4) have the same optimal values, adding these bounds as constraints on π_m in (6.3) does not change its optimal value. Since $\hat{x} > 0$ there is $r > 0$ such that

$$\mathbb{B}(\hat{x}, r) \subseteq \{x \geq 0\}. \tag{6.5}$$

We argue that $\mathrm{Aff}(\mathcal{S}_m) = \mathcal{A}_m$. Indeed, the inclusion $\mathrm{Aff}(\mathcal{S}_m) \subseteq \mathcal{A}_m$ is clear. Now if $x \in \mathcal{A}_m$ then if $x = \hat{x}$ we have that $x \in \mathcal{S}_m \subseteq \mathrm{Aff}(\mathcal{S}_m)$ and if $x \neq \hat{x}$, recalling that $\hat{x} \in \mathcal{A}_m$ satisfies (6.5) we have that

$$y := \hat{x} + \frac{r}{2}\frac{x - \hat{x}}{\|x - \hat{x}\|} \in \mathcal{A}_m \cap \mathbb{B}(\hat{x}, r) \subseteq \mathcal{S}_m.$$

Therefore x belongs to the line that contains y and $\hat{x}$ with $y, \hat{x}$ belonging to $\mathcal{S}_m$ which implies $x \in \mathrm{Aff}(\mathcal{S}_m)$ and $\mathrm{Aff}(\mathcal{S}_m) = \mathcal{A}_m$.

It follows that

$$\mathbb{B}(\hat{x}, r) \cap \mathrm{Aff}(\mathcal{S}_m) = \mathbb{B}(\hat{x}, r) \cap \mathcal{A}_m \subseteq \mathcal{S}_m$$

and that there is $\rho_*(m) > 0$ such that

$$\mathbb{B}(0, \rho_*) \cap (\overline{A}_m \mathcal{A}_m - b_m) \subseteq \overline{A}_m(\mathbb{B}(\hat{x}, r) \cap \mathcal{A}_m) - b_m.$$

Let $\bar{\pi}_m$ be an optimal solution of problem (6.4) and let $z = 0$ if $\bar{\pi}_m = 0$ and $z = -\frac{\bar{\pi}_m}{\|\bar{\pi}_m\|_2}\rho_*$ otherwise. Observe that $z \in \mathbb{B}(0, \rho_*) \cap (\overline{A}_m \mathcal{A}_m - b_m)$ and therefore $z \in \overline{A}_m(\mathbb{B}(\hat{x}, r) \cap \mathcal{A}_m) - b_m \subseteq \overline{A}_m \mathcal{S}_m - b_m$ and z can be written $z = \overline{A}_m \tilde{x} - b_m$ for $\tilde{x} \in \mathbb{B}(\hat{x}, r) \cap \mathcal{S}_m$. It follows that

$$\begin{array}{rcl}\mathcal{Q}_1(x_0) = \theta(\bar{\pi}_m) & \leq & \mathbb{E}[c^\top \tilde{x}] + \bar{\pi}_m^\top(\overline{A}_m \tilde{x} - b_m) \\ & \leq & \mathbb{E}[c^\top \hat{x}] + r\sum_{t=1}^T \mathbb{E}[\|c_t\|_2] + \bar{\pi}_m^\top z \\ & = & \mathbb{E}[c^\top \hat{x}] + r\sum_{t=1}^T \mathbb{E}[\|c_t\|_2] - \rho_*(m)\|\bar{\pi}_m\|_2\end{array}$$

which gives for every node n of stage t that

$$\|\bar{\pi}_n\|_2 \leq \max_{m \in \mathtt{Nodes}(t)} \frac{\mathbb{E}[c^\top \hat{x}] - \mathcal{Q}_1(x_0) + r\sum_{t=1}^T \mathbb{E}[\|c_t\|_2]}{\rho_*(m)}$$

with corresponding box constraints $\underline{\pi}_t, \overline{\pi}_t$ where $\mathtt{Nodes}(t)$ are the nodes of stage t. □

Proof of Proposition 4.2. We show by induction on k that $V_t \leq V_t^k$ for $t = 2, \ldots, T$. For $k = 0$ these relations hold by definition. Assume that for some $k \geq 1$ we have $V_t \leq V_t^{k-1}$ for $t = 2, \ldots, T$. We show by backward induction on t that

$V_t \leq V_t^k$ for $t = 2, \ldots, T$. Observe that for any π_{T-1}, optimization problem (6.1) with optimal value $\overline{V}_T^k(\pi_{T-1})$ is feasible. Indeed, since primal problem (2.1) is feasible and has a finite optimal value, the corresponding dual problem is feasible which implies that there is $\pi_{T1}, \ldots, \pi_{TN_T}$ satisfying $A_T^\top \pi_{Tj} \leq c_T$, $\underline{\pi}_T \leq \pi_{Tj} \leq \overline{\pi}_T$, $j = 1, \ldots, N_T$, and for every such points we can find $\zeta_T \geq 0$ satisfying the remaining constraints in (6.1). Therefore $\overline{V}_T^k(\pi_{T-1})$ is finite for every π_{T-1} and is the optimal value of the corresponding dual optimization problem, i.e., for any π_{T-1} we get

$$\overline{V}_T^k(\pi_{T-1}) = \begin{array}{l} \min\limits_{\alpha,\delta,\overline{\Psi},\underline{\Psi}} \delta^\top (c_{T-1} - A_{T-1}^\top \pi_{T-1}) + c_T^\top \sum\limits_{j=1}^{N_T} \alpha_j + \sum\limits_{j=1}^{N_T} \overline{\Psi}_j^\top \overline{\pi}_T - \sum\limits_{j=1}^{N_T} \underline{\Psi}_j^\top \underline{\pi}_T \\ A_T \alpha_j + p_{Tj} B_{Tj} \delta - \underline{\Psi}_j + \overline{\Psi}_j = p_{Tj} b_{Tj},\ j = 1, \ldots, N_T, \\ 0 \leq \delta \leq v_{Tk}, \alpha_j, \underline{\Psi}_j, \overline{\Psi}_j \geq 0,\ j = 1, \ldots, N_T. \end{array}$$

Using this dual representation and the definition of $\overline{\theta}_T^k, \overline{\beta}_T^k$, we get for every π_{T-1}:

$$\overline{\theta}_T^k + \langle \overline{\beta}_T^k, \pi_{T-1} \rangle \geq \overline{V}_T^k(\pi_{T-1}). \tag{6.6}$$

Recalling representation (6.1) for $\overline{V}_T^k(\pi_{T-1})$, observe that for every $\pi_{T-1} \in \text{dom}(V_T)$ we have $\overline{V}_T^k(\pi_{T-1}) \geq V_T(\pi_{T-1})$ whereas for $\pi_{T-1} \notin \text{dom}(V_T)$ we have $V_T(\pi_{T-1}) = -\infty$ while $\overline{V}_T^k(\pi_{T-1})$ is finite, which shows that for every π_{T-1} we have $\overline{V}_T^k(\pi_{T-1}) \geq V_T(\pi_{T-1})$, which, combined with (6.6) and the induction hypothesis, gives

$$V_T^k(\pi_{T-1}) \geq V_T(\pi_{T-1})$$

for everyπ_{T-1}.

Now assume that $V_{t+1}^k(\pi_t) \geq V_{t+1}(\pi_t)$ for all π_t for some $t \in \{2, \ldots, T-1\}$. We want to show that $V_t^k(\pi_{t-1}) \geq V_t(\pi_{t-1})$ for all π_{t-1}. First observe that for every π_{t-1}, linear program (6.2) with optimal value $\overline{V}_t^k(\pi_{t-1})$ is feasible and has a finite optimal value. Therefore we can express $\overline{V}_t^k(\pi_{t-1})$ as the optimal value of the corresponding dual problem given by

$$\begin{array}{l} \min\limits_{\delta,\nu,\overline{\Psi},\underline{\Psi}} \delta^\top \left[c_{t-1} - A_{t-1}^\top \pi_{t-1}\right] + \sum\limits_{i=0}^{k} \overline{\theta}_{t+1}^i \sum\limits_{j=1}^{N_t} \nu_i(j) + \sum\limits_{j=1}^{N_t} \overline{\Psi}_j^\top \overline{\pi}_t - \sum\limits_{j=1}^{N_t} \underline{\Psi}_j^\top \underline{\pi}_t \\ p_{tj} B_{tj} \delta - \sum\limits_{i=0}^{k} \nu_i(j) \overline{\beta}_{t+1}^i - \underline{\Psi}_j + \overline{\Psi}_j = p_{tj} b_{tj},\ j = 1, \ldots, N_t, \\ \sum\limits_{i=0}^{k} \nu_i(j) = p_{tj}, \underline{\Psi}_j, \overline{\Psi}_j \geq 0,\ j = 1, \ldots, N_t, \\ \nu_0, \ldots, \nu_k \geq 0, 0 \leq \delta \leq v_{tk}. \end{array} \tag{6.7}$$

Using this representation of $\overline{V}_t^k$ and the definition of $\overline{\theta}_t^k, \overline{\beta}_t^k$, we obtain for every π_{t-1}:

$$\overline{\theta}_t^k + \langle \overline{\beta}_t^k, \pi_{t-1} \rangle \geq \overline{V}_t^k(\pi_{t-1}). \tag{6.8}$$

Next, recalling representation (6.2) for $\overline{V}_t^k(\pi_{t-1})$ and the induction hypothesis, we get

$$\overline{V}_t^k(\pi_{t-1}) \geq \widehat{V}_t^k(\pi_{t-1}) \tag{6.9}$$

where

$$\widehat{V}_t^k(\pi_{t-1}) := \begin{array}{l} \max\limits_{\pi_{t1},\ldots,\pi_{tN_t},\zeta_t} \sum\limits_{j=1}^{N_t} p_{tj}\left(b_{tj}^\top \pi_{tj} + V_{t+1}(\pi_{tj})\right) - v_{tk}^\top \zeta_t \\ A_{t-1}^\top \pi_{t-1} + \sum\limits_{j=1}^{N_t} p_{tj} B_{tj}^\top \pi_{tj} \le c_{t-1} + \zeta_t, \\ \zeta_t \ge 0, \underline{\pi}_t \le \pi_{tj} \le \overline{\pi}_t,\ j = 1, ..., N_t. \end{array}$$

Similarly to the induction step $t = T$, for every π_{t-1}, we have

$$\widehat{V}_t^k(\pi_{t-1}) \ge V_t(\pi_{t-1}). \tag{6.10}$$

Combining (6.8), (6.9), and (6.10) with the induction hypothesis, we obtain $V_t^k(\pi_{t-1}) \ge V_t(\pi_{t-1})$ for all π_{t-1} which achieves the proof of the induction step t.

In particular $V_2^{k-1} \ge V_2$ which implies that V^{k-1} is greater than or equal to the optimal value of dual problem (2.3) which is also, by linear programming duality, the optimal value of primal problem (2.1). □

The proof of Theorem 4.3 is based on the following lemma:

Lemma 6.1. *Suppose that the multistage problem* (2.1) *has a finite optimal value. Then for sufficiently large values of the components of vectors* v_t*, in the dynamic equations* (2.11)*, the optimal value of the multistage problem defined by these dynamic equations coincides with the optimal value of the original problem* (2.1)*.*

Proof. As it was already mentioned, since it is assumed that the number of scenarios is finite, we can view problem (2.1) as a large linear program (deterministic equivalent) written under the form

$$\min_x c^\top x \ \text{ s.t. } \mathcal{A}x = b,\ x \ge 0. \tag{6.11}$$

Also since (2.1) has a finite optimal value, it has a nonempty set of optimal solutions and there is a bounded optimal solution of (6.11). Let us fix such an optimal solution $\bar{x}$. We have that problem (6.11) can be written

$$\min_x c^\top x \ \text{ s.t. } \mathcal{A}x = b,\ 0 \le x \le \bar{x}. \tag{6.12}$$

The dynamic programming equations (2.5) - (2.7) represent the standard dual of (2.1). We can also think about that dual as a large linear programming problem of the form (this is the dual of (6.11)):

$$\max_\pi b^\top \pi \ \text{ s.t. } \mathcal{A}^\top \pi \le c. \tag{6.13}$$

Similarly the deterministic equivalent of penalized dynamic equations (2.11) can be written as:

$$\max_{\pi,\zeta} b^\top \pi - v^\top \zeta \ \text{ s.t. } \mathcal{A}^\top \pi \le c + \zeta, \zeta \ge 0. \tag{6.14}$$

Next, from optimality conditions of linear programs, (x, π) is an optimal primal-dual pair for (6.11)-(6.13) if and only if

$$x^\top(\mathcal{A}^\top \pi - c) = 0,\ \mathcal{A}x = b,\ x \ge 0,\ \mathcal{A}^\top \pi \le c. \tag{6.15}$$

The corresponding optimality conditions for (6.14) are

(6.16)
$$x^\top(\mathcal{A}^\top\pi - c - \zeta) - \zeta^\top\gamma = 0,\ \mathcal{A}^\top\pi \le c + \zeta,\ \zeta \ge 0,\ \mathcal{A}x = b,\ x \ge 0,\ \gamma \ge 0,\ x = v - \gamma.$$

Now let $\bar{\pi}$ be an optimal dual solution, i.e., an optimal solution of (6.13). Then (6.15) is satisfied with $(x, \pi) = (\bar{x}, \bar{\pi})$. It follows that if $v \ge \bar{x}$, then $(x, \pi, \zeta, \gamma) = (\bar{x}, \bar{\pi}, 0, v - \bar{x})$ with $\zeta = 0$ satisfies (6.16), and hence $(\bar{\pi}, \bar{\zeta}) = (\bar{\pi}, 0)$ is an optimal solution of (6.14) showing that the optimal value of (6.14) is $b^\top\bar{\pi} = c^\top\bar{x}$, i.e., the optimal value of (6.11). We obtain that for $v \ge \bar{x}$, the optimal values of problems (6.13) and (6.14) do coincide.[6] □

Proof of Theorem 4.3. Dual SDDP with penalizations is SDDP applied to Dynamic Programming equations corresponding to a linear program with finite optimal value, satisfying relatively complete recourse with discrete uncertainties of finite support. Since samples $\tilde{\xi}_t^k$ in Dual SDDP with penalizations are independent, we can follow the convergence proof of SDDP for linear programs from [23] to obtain that V^k converges to the optimal value of the penalized linear programs, which, by Lemma 6.1 (observe that the Lemma can be applied since $\lim_{k\to+\infty} v_{tk} = +\infty$), is the optimal value of (2.1). □

[6] Observe that the dual of (6.14) is given by

$$\min_x c^\top x \ \text{ s.t. } \mathcal{A}x = b,\ 0 \le x \le v,$$

and for $v \ge \bar{x}$, this linear program has the same optimal value as (6.12), which, as we have seen, is equivalent to primal problem (2.1).