INEXACT CUTS IN DETERMINISTIC AND STOCHASTIC DUAL DYNAMIC PROGRAMMING APPLIED TO LINEAR OPTIMIZATION PROBLEMS

Vincent Guigues
School of Applied Mathematics, FGV
Praia de Botafogo, Rio de Janeiro, Brazil
vguigues@fgv.br

Abstract. We introduce an extension of Dual Dynamic Programming (DDP) to solve linear dynamic programming equations. We call this extension IDDP-LP which applies to situations where some or all primal and dual subproblems to be solved along the iterations of the method are solved with a bounded error (inexactly). We provide convergence theorems both in the case when errors are bounded and for asymptotically vanishing errors. We extend the analysis to stochastic linear dynamic programming equations, introducing Inexact Stochastic Dual Dynamic Programming for linear programs (ISDDP-LP), an inexact variant of SDDP applied to linear programs corresponding to the situation where some or all problems to be solved in the forward and backward passes of SDDP are solved approximately. We also provide convergence theorems for ISDDP-LP for bounded and asymptotically vanishing errors. Finally, we present the results of numerical experiments comparing SDDP and ISSDP-LP on a portfolio problem with direct transaction costs modelled as a multistage stochastic linear optimization problem. On these experiments, ISDDP-LP allows us to obtain a good policy faster than SDDP.

Stochastic programming and Decomposition algorithms and Monte Carlo sampling and SDDP and Inexact cuts in SDDP

AMS subject classifications: 90C15, 90C90.

1. Introduction

Multistage stochastic convex programs are useful to model many real-life applications in engineering and finance, see for instance [22] and references therein. A popular solution method for such problems is Stochastic Dual Dynamic Programming (SDDP, pioneered by [17]) which introduces sampling in the Nested Decomposition (ND) algorithm [5, 4]. It was extended and analysed in several publications: extension for problems with interstage dependent processes [15], [7], adaptations for risk-averse problems [13, 12, 21, 16], regularizations [2, 14], cut selection [18, 19, 11], extension to problems with integer variables [24], convergence proofs for linear programs [20], for nonlinear risk-neutral programs [6], and for nonlinear risk-averse programs [8]. Recently, Inexact SDDP (ISDDP) was proposed in [10]: it uses inexact cuts in SDDP applied to Multistage Stochastic NonLinear Problems (MSNLPs). The motivations for ISDDP are twofold:

(i) first, when SDDP is applied to nonlinear problems, only approximate solutions for the subproblems solved in the backward and forward passes are available. ISDDP allows us to build valid cuts on the basis of approximate solutions to these subproblems.
(ii) Second, for the first iterations and the first stages, the cuts computed by SDDP can be quite distant from the corresponding recourse function in the neighborhood of the trial point at which the cut was computed (see for instance the numerical experiments in [11, 9]), making this cut dominated by other "more relevant" cuts in this neighborhood as the method progresses. Therefore, it is natural to try and solve less accurately, inexactly, the subproblems in the forward and backward passes for the first iterations and stages and to increase the precision of the computed solutions as the algorithm progresses to decrease the overall computational bulk.

The goal of this paper is to pursue this line of research considering linear instead of nonlinear programs. More precisely, we propose and analyse a variant of SDDP applied to Multistage Stochastic Linear Programs

(MSLPs) called ISDDP-LP (Inexact SDDP for Linear Programs), which allows us to build cuts, called inexact cuts, on the basis of feasible (not necessarily optimal and eventually far from optimal) solutions to the subproblems solved in the forward and backward passes of the method. The combination of inexact cuts with Benders Decomposition [3] was first proposed by [23] for two-stage stochastic linear programs. Therefore, ISDDP-LP can be seen as an extension to a multistage setting of the algorithm presented in [23].

The main results of this paper are the following:

(A) we propose an extension of DDP (Dual Dynamic Programming, the deterministic counterpart of SDDP) called IDDP-LP (Inexact DDP for Linear Programs) which builds inexact cuts for the cost-to-go functions. For a problem with T periods, when noises (error terms quantifying the inexactness) are bounded by $\bar{\delta}$ in the forward pass and by $\bar{\varepsilon}$ in the backward pass, we show in Theorem 3.2 that the limit superior of the sequence of upper bounds is at most $(\bar{\delta}+\bar{\varepsilon})\frac{T(T+1)}{2}$ distant to the optimal value of the problem and the limit inferior of the sequence of lower bounds is at most $\bar{\delta}T+\bar{\varepsilon}(T-1)$ distant to this optimal value. When noises asymptotically vanish, we show that IDDP-LP solves the original optimization problem.

(B) The study of IDDP-LP allows us to introduce and analyse ISDDP-LP which builds inexact cuts for the cost-to-go functions of a MSLP. We provide a convergence theorem (Theorem 4.1) for ISDDP-LP when noises are bounded and show in Theorem 4.2 that ISDDP-LP solves the original MSLP when noises asymptotically vanish.

(C) We compare the computational bulk of SDDP and ISDDP-LP on four instances of a portfolio optimization problem with direct transaction costs. On these instances, ISDDP-LP allows us to obtain a good policy faster than SDDP (compared to SDDP, with ISDDP-LP the CPU time decreases by a factor of 6.2%, 6.4%, 6.5%, and 11.1% for the four instances considered). It is also interesting to notice that once SDDP is implemented on a MSLP, the implementation of the corresponding ISDDP-LP with given parameters $(\delta_t^k,\, \varepsilon_t^k)$ is straightforward. Therefore, if for a given application, or given classes of problems, we can find suitable choices of parameters $(\delta_t^k,\, \varepsilon_t^k)$ either using the rules from Remark 4.3, other rules, or "playing" with these parameters running ISDDP-LP on instances, ISDDP-LP could allow us to solve similar new instances quicker than SDDP.

The paper is organized as follows. In Section 2 we explain how to build inexact cuts for the value function of a linear program (this elementary observation is used to build cuts in IDDP-LP and ISDDP-LP). In Section 3 we introduce and analyse IDDP-LP while in Section 4 we introduce and analyse ISDDP-LP. Numerical simulations are presented in Section 5.

2. Computing inexact cuts for the value function of a linear program

Let $X \subset \mathbb{R}^m$ and let $\mathcal{Q} : X \to \overline{\mathbb{R}}$ be the value function given by

$$\mathcal{Q}(x) = \left\{ \begin{array}{l} \min_{y \in \mathbb{R}^n} \; c^T y \\ y \in Y(x) := \{y \in \mathbb{R}^n : Ay + Bx = b, Cy \leq f\}, \end{array} \right. \tag{2.1}$$

for matrices and vectors of appropriate sizes. We assume:

(H) for every $x \in X$, the set $Y(x)$ is nonempty and $y \to c^T y$ is bounded from below on $Y(x)$.

If Assumption (H) holds then $\mathcal{Q}$ is convex and finite on X and by duality we can write

$$\mathcal{Q}(x) = \left\{ \begin{array}{l} \max_{\lambda,\mu} \; \lambda^T(b - Bx) + \mu^T f \\ A^T\lambda + C^T\mu = c, \mu \leq 0, \end{array} \right. \tag{2.2}$$

for $x \in X$. We will call cut for $\mathcal{Q}$ on X an affine lower bounding function for $\mathcal{Q}$ on X. We say that cut $\mathcal{C}$ is inexact at $\bar{x}$ for convex function $\mathcal{Q}$ if the distance $\mathcal{Q}(\bar{x}) - \mathcal{C}(\bar{x})$ between the values of $\mathcal{Q}$ and of the cut at $\bar{x}$ is strictly positive. When $\mathcal{Q}(\bar{x}) = \mathcal{C}(\bar{x})$ we will say that cut $\mathcal{C}$ is exact at $\bar{x}$.

The following simple proposition will be used in the sequel: it provides an inexact cut for $\mathcal{Q}$ at $\bar{x} \in X$ on the basis of an approximate solution of (2.2):

Proposition 2.1. *Let Assumption (H) hold and let $\bar{x} \in X$.*

Let $(\hat{\lambda}, \hat{\mu})$ be an ϵ-optimal basic feasible solution for dual problem (2.2) written for $x = \bar{x}$ (it is in particular an extreme point of the feasible set), i.e., $A^T \hat{\lambda} + C^T \hat{\mu} = c$, $\hat{\mu} \leq 0$, and

$$\hat{\lambda}^T (b - B\bar{x}) + \hat{\mu}^T f \geq \mathcal{Q}(\bar{x}) - \varepsilon, \tag{2.3}$$

for some $\varepsilon \geq 0$. Then the affine function

$$\mathcal{C}(x) := \hat{\lambda}^T (b - Bx) + \hat{\mu}^T f$$

is a cut for $\mathcal{Q}$ at $\bar{x}$, i.e., for every $x \in X$ we have $\mathcal{Q}(x) \geq \mathcal{C}(x)$ and the distance $\mathcal{Q}(\bar{x}) - \mathcal{C}(\bar{x})$ between the values of $\mathcal{Q}$ and of the cut at $\bar{x}$ is at most ε.

Proof. $\mathcal{C}$ is indeed a cut for $\mathcal{Q}$ (an affine lower bounding function for $\mathcal{Q}$) because $(\hat{\lambda}, \hat{\mu})$ is feasible for optimization problem (2.2). Relation (2.3) gives the upper bound ε for $\mathcal{Q}(\bar{x}) - \mathcal{C}(\bar{x})$. □

3. Inexact cuts in DDP applied to linear programs

3.1. **Algorithm.** Consider the linear program

$$\begin{array}{l} \min\limits_{x_1, \ldots, x_T \in \mathbb{R}^n} \sum_{t=1}^{T} c_t^T x_t \\ A_t x_t + B_t x_{t-1} = b_t,\ x_t \geq 0,\ t = 1, \ldots, T, \end{array} \tag{3.4}$$

where x_0 is given. For this problem we can write the following dynamic programming equations: for $t = 1, \ldots, T$,

$$\mathcal{Q}_t(x_{t-1}) = \left\{ \begin{array}{l} \min\limits_{x_t \in \mathbb{R}^n} c_t^T x_t + \mathcal{Q}_{t+1}(x_t) \\ A_t x_t + B_t x_{t-1} = b_t, x_t \geq 0 \end{array} \right. \tag{3.5}$$

with the convention that $\mathcal{Q}_{T+1}$ is null. Clearly, the optimal value of (3.4) is $\mathcal{Q}_1(x_0)$.

For convenience, we will denote

$$X_t(x_{t-1}) := \{x_t \in \mathbb{R}^n : A_t x_t + B_t x_{t-1} = b_t,\ x_t \geq 0\}.$$

We make the following assumption:

(H1-D) The set $X_1(x_0)$ is nonempty and bounded and for every $x_1 \in X_1(x_0)$, for every $t = 2, \ldots, T$, for every $x_2 \in X_2(x_1), \ldots, x_{t-1} \in X_{t-1}(x_{t-2})$, the set $X_t(x_{t-1})$ is nonempty and bounded.

In this section, we introduce a variant of DDP to solve (3.4) called Inexact DDP for linear programs (IDDP-LP) where the subproblems of the forward and backward passes are solved approximately. At iteration k, for $t = 2, \ldots, T$, convex function $\mathcal{Q}_t$ is approximated by a piecewise affine lower bounding function $\mathcal{Q}_t^k$ which is a maximum of affine lower bounding functions $\mathcal{C}_t^i$ called inexact cuts:

$$\mathcal{Q}_t^k(x_{t-1}) = \max_{1 \leq i \leq k} \mathcal{C}_t^i(x_{t-1}) \text{ with } \mathcal{C}_t^i(x_{t-1}) = \theta_t^i + \langle \beta_t^i, x_{t-1} \rangle$$

where coefficients θ_t^i, β_t^i are computed as explained below. The steps of IDDP-LP are as follows:

IDDP-LP, Step 1: Initialization. For $t = 2, \ldots, T$, take for $\mathcal{Q}_t^0$ a known lower bounding affine function for $\mathcal{Q}_t$. Set the iteration count k to 1 and $\mathcal{Q}_{T+1}^0 \equiv 0$.

IDDP-LP, Step 2: Forward pass. Using approximation $\mathcal{Q}_{t+1}^{k-1}$ of $\mathcal{Q}_{t+1}$ (computed at previous iterations), we compute a δ_t^k-optimal basic feasible solution x_t^k of the problem (it is in particular an extreme point of the feasible set)

$$\left\{ \begin{array}{l} \min_{x_t \in \mathbb{R}^n} c_t^T x_t + \mathcal{Q}_{t+1}^{k-1}(x_t) \\ x_t \in X_t(x_{t-1}^k) \end{array} \right. \tag{3.6}$$

for $t = 1, \ldots, T$, where $x_0^k = x_0$.

IDDP-LP, Step 3: Backward pass. The backward pass builds inexact cuts for $\mathcal{Q}_t$ at trial points x_{t-1}^k computed in the forward pass. For $k \geq 1$ and $t = 1, \ldots, T$, we introduce the function $\underline{\mathcal{Q}}_t^k : \mathbb{R}^n \to \overline{\mathbb{R}}$ given by

$$\underline{\mathcal{Q}}_t^k(x_{t-1}) = \left\{ \begin{array}{l} \min_{x_t \in \mathbb{R}^n} c_t^T x_t + \mathcal{Q}_{t+1}^k(x_t) \\ x_t \in X_t(x_{t-1}), \end{array} \right. \tag{3.7}$$

where $\mathcal{Q}_{T+1}^k \equiv 0$. We solve approximately the problem

$$\mathcal{Q}_T(x_{T-1}^k) = \left\{ \begin{array}{l} \displaystyle\min_{x_T \in \mathbb{R}^n} c_T^T x_T \\ A_T x_T + B_T x_{T-1}^k = b_T, x_T \geq 0, \end{array} \right. \quad \text{with dual} \left\{ \begin{array}{l} \max_\lambda \lambda^T (b_T - B_T x_{T-1}^k) \\ A_T^T \lambda \leq c_T. \end{array} \right. \tag{3.8}$$

More precisely, let λ_T^k be an ε_T^k-optimal basic feasible solution of the dual problem above (it is in particular an extreme point of the feasible set). We compute

$$\theta_T^k = \langle b_T, \lambda_T^k \rangle \text{ and } \beta_T^k = -B_T^T \lambda_T^k. \tag{3.9}$$

Using Proposition 2.1 we have that $\mathcal{C}_T^k(x_{T-1}) = \theta_T^k + \langle \beta_T^k, x_{T-1} \rangle$ is an inexact cut for $\mathcal{Q}_T$ at x_{T-1}^k which satisfies

$$\mathcal{Q}_T(x_{T-1}^k) - \mathcal{C}_T^k(x_{T-1}^k) \leq \varepsilon_T^k. \tag{3.10}$$

Then for $t = T-1$ down to $t = 2$, knowing $\mathcal{Q}_{t+1}^k \leq \mathcal{Q}_{t+1}$, consider the optimization problem

$$\underline{\mathcal{Q}}_t^k(x_{t-1}^k) = \left\{ \begin{array}{l} \displaystyle\min_{x_t} c_t^T x_t + \mathcal{Q}_{t+1}^k(x_t) \\ x_t \in X_t(x_{t-1}^k) \end{array} \right. = \left\{ \begin{array}{l} \displaystyle\min_{x_t, f} c_t^T x_t + f \\ A_t x_t + B_t x_{t-1}^k = b_t, x_t \geq 0, \\ f \geq \theta_{t+1}^i + \langle \beta_{t+1}^i, x_t \rangle, i = 1, \ldots, k. \end{array} \right. \tag{3.11}$$

Observe that due to (H1-D) the above problem is feasible and has a finite optimal value. Therefore $\underline{\mathcal{Q}}_t^k(x_{t-1}^k)$ can be expressed as the optimal value of the corresponding dual problem:

$$\underline{\mathcal{Q}}_t^k(x_{t-1}^k) = \left\{ \begin{array}{l} \displaystyle\max_{\lambda, \mu} \lambda^T (b_t - B_t x_{t-1}^k) + \sum_{i=1}^k \mu_i \theta_{t+1}^i \\ \displaystyle A_t^T \lambda + \sum_{i=1}^k \mu_i \beta_{t+1}^i \leq c_t, \ \sum_{i=1}^k \mu_i = 1, \\ \mu_i \geq 0, \ i = 1, \ldots, k. \end{array} \right. \tag{3.12}$$

Let (λ_t^k, μ_t^k) be an ε_t^k-optimal basic feasible solution of dual problem (3.12). We compute

$$\theta_t^k = \langle \lambda_t^k, b_t \rangle + \langle \mu_t^k, \theta_{t+1,k} \rangle \text{ and } \beta_t^k = -B_t^T \lambda_t^k, \tag{3.13}$$

where vector $\theta_{t+1,k}$ has components $\theta_{t+1}^i, i = 1, \ldots, k$, arranged in the same order as components $\mu_t^k(i), i = 1, \ldots, k$, of μ_t^k. Recalling that $\mathcal{C}_t^k(x_{t-1}) = \theta_t^k + \langle \beta_t^k, x_{t-1} \rangle$ and using Proposition 2.1, we have

$$\underline{\mathcal{Q}}_t^k(x_{t-1}) \geq \mathcal{C}_t^k(x_{t-1}) \quad \text{and} \quad \underline{\mathcal{Q}}_t^k(x_{t-1}^k) - \mathcal{C}_t^k(x_{t-1}^k) \leq \varepsilon_t^k. \tag{3.14}$$

Using the fact that $\mathcal{Q}_{t+1}^k(x_{t-1}) \leq \mathcal{Q}_{t+1}(x_{t-1})$, we have $\underline{\mathcal{Q}}_t^k(x_{t-1}) \leq \mathcal{Q}_t(x_{t-1})$, and therefore

$$\mathcal{Q}_t(x_{t-1}) \geq \mathcal{C}_t^k(x_{t-1}) \tag{3.15}$$

which shows that $\mathcal{C}_t^k$ is a cut for $\mathcal{Q}_t$.

IDDP-LP, Step 4: Do $k \leftarrow k+1$ and go to Step 2.

Following the proof of Lemma 1 in [20], we obtain that for all $t = 2, \ldots, T+1$, the collection of distinct values $(\theta_t^k, \beta_t^k)_k$ is finite and cut coefficients $(\theta_t^k, \beta_t^k)_k$ are uniformly bounded. Observe that this proof uses the fact that (λ_t^k, μ_t^k) are extreme points of the feasible set of (3.12). There could however be unbounded sequences of approximate optimal feasible solutions to (3.12).

3.2. **Convergence analysis.** In this section we state a convergence result for IDDP-LP in Theorem 3.2 when noises $\delta_t^k, \varepsilon_t^k$ are bounded and in Theorem 3.3 when these noises asymptotically vanish.

We will need the following simple extension of [6, Lemma A.1]:

Lemma 3.1. *Let X be a compact set, let $f : X \to \mathbb{R}$ be Lipschitz continuous, and suppose that the sequence of L-Lipschitz continuous functions $f^k, k \in \mathbb{N}$ satisfies $f^k(x) \le f^{k+1}(x) \le f(x)$ for all $x \in X,\ k \in \mathbb{N}$. Let $(x^k)_{k\in\mathbb{N}}$ be a sequence in X and assume that*

$$\varlimsup_{k\to+\infty} f(x^k) - f^k(x^k) \le S \tag{3.16}$$

for some $S \ge 0$. Then

$$\varlimsup_{k\to+\infty} f(x^k) - f^{k-1}(x^k) \le S. \tag{3.17}$$

Proof. Let us show (3.17) by contradiction. Assume that (3.17) does not hold. Then there exist $\varepsilon_0 > 0$ and $\sigma : \mathbb{N} \to \mathbb{N}$ increasing such that for every $k \in \mathbb{N}$ we have

$$f(x^{\sigma(k)}) - f^{\sigma(k)-1}(x^{\sigma(k)}) > S + \varepsilon_0. \tag{3.18}$$

Since $x^{\sigma(k)}$ is a sequence of the compact set X, it has some convergent subsequence which converges to some $x_* \in X$. Taking into account (3.16) and the fact that f^k are L-Lipschitz continuous, we can take σ such that (3.18) holds and

$$\begin{array}{rcll}
f(x^{\sigma(k)}) - f^{\sigma(k)}(x^{\sigma(k)}) & \le & S + \dfrac{\varepsilon_0}{4}, & (3.19)\\
f^{\sigma(k)-1}(x^{\sigma(k)}) - f^{\sigma(k)-1}(x_*) & > & -\dfrac{\varepsilon_0}{4}, & (3.20)\\
f^{\sigma(k)}(x_*) - f^{\sigma(k)}(x^{\sigma(k)}) & > & -\dfrac{\varepsilon_0}{4}. & (3.21)
\end{array}$$

Therefore for every $k \ge 1$ we get

$$\begin{array}{rcll}
f^{\sigma(k)}(x_*) - f^{\sigma(k-1)}(x_*) & \ge & f^{\sigma(k)}(x_*) - f^{\sigma(k)-1}(x_*) & \text{since } \sigma(k) \ge \sigma(k-1)+1,\\
& = & f^{\sigma(k)}(x_*) - f^{\sigma(k)}(x^{\sigma(k)}) & (> -\varepsilon_0/4 \text{ by } (3.21)),\\
& & +f^{\sigma(k)}(x^{\sigma(k)}) - f(x^{\sigma(k)}) & (\ge -S - \varepsilon_0/4 \text{ by } (3.19)),\\
& & +f(x^{\sigma(k)}) - f^{\sigma(k)-1}(x^{\sigma(k)}) & (> S + \varepsilon_0 \text{ by } (3.18)),\\
& & +f^{\sigma(k)-1}(x^{\sigma(k)}) - f^{\sigma(k)-1}(x_*) & (> -\varepsilon_0/4 \text{ by } (3.20)),\\
& > & \varepsilon_0/4, &
\end{array}$$

which implies $f^{\sigma(k)}(x_*) > f^{\sigma(0)}(x_*) + k\frac{\varepsilon_0}{4}$. This is in contradiction with the fact that the sequence $f^{\sigma(k)}(x_*)$ is bounded from above by $f(x_*)$. □

Theorem 3.2 (Convergence of IDDP-LP with bounded noises)**.** *Consider the sequences of decisions (x_t^k) and of functions $(\mathcal{Q}_t^k)$ generated by IDDP-LP. Assume that (H1-D) holds and that noises ε_t^k and δ_t^k are bounded: $0 \le \varepsilon_t^k \le \bar{\varepsilon}$, $0 \le \delta_t^k \le \bar{\delta}$ for some $\bar{\delta}, \bar{\varepsilon} \ge 0$.*

(i) *Then for $t = 2, \ldots, T+1$,*

$$0 \le \varliminf_{k\to+\infty} \mathcal{Q}_t(x_{t-1}^k) - \mathcal{Q}_t^k(x_{t-1}^k) \le \varlimsup_{k\to+\infty} \mathcal{Q}_t(x_{t-1}^k) - \mathcal{Q}_t^k(x_{t-1}^k) \le (\bar{\delta} + \bar{\varepsilon})(T - t + 1). \tag{3.22}$$

(ii) *The limit superior and limit inferior of the sequence of upper bounds $(\sum_{t=1}^T c_t^T x_t^k)_k$ on the optimal value $\mathcal{Q}_1(x_0)$ of (3.4) satisfy*

$$\mathcal{Q}_1(x_0) \le \varliminf_{k\to+\infty} \sum_{t=1}^T c_t^T x_t^k \le \varlimsup_{k\to+\infty} \sum_{t=1}^T c_t^T x_t^k \le \mathcal{Q}_1(x_0) + (\bar{\delta} + \bar{\varepsilon}) \frac{T(T+1)}{2}. \tag{3.23}$$

(iii) *The limit superior and limit inferior of the sequence of lower bounds $(\underline{\mathcal{Q}}_1^k(x_0))_k$ on the optimal value $\mathcal{Q}_1(x_0)$ of (3.4) satisfy*

$$\mathcal{Q}_1(x_0) - \bar{\delta}T - \bar{\varepsilon}(T-1) \le \varliminf_{k\to+\infty} \underline{\mathcal{Q}}_1^k(x_0) \le \varlimsup_{k\to+\infty} \underline{\mathcal{Q}}_1^k(x_0) \le \mathcal{Q}_1(x_0). \tag{3.24}$$

Proof. We show (i) by backward induction on t. Relation (3.22) holds for $t = T + 1$. Now assume that

$$0 \leq \varliminf_{k\to+\infty} \mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_{t+1}^k(x_t^k) \leq \varlimsup_{k\to+\infty} \mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_{t+1}^k(x_t^k) \leq (\bar{\varepsilon} + \bar{\delta})(T - t). \tag{3.25}$$

for some $t \in \{2, \ldots, T\}$. We have

$$\begin{array}{lll} 0 & \leq & \mathcal{Q}_t(x_{t-1}^k) - \mathcal{Q}_t^k(x_{t-1}^k) \leq \mathcal{Q}_t(x_{t-1}^k) - \mathcal{C}_t^k(x_{t-1}^k) \text{ since } \mathcal{Q}_t^k \geq \mathcal{C}_t^k, \\ & \leq & \bar{\varepsilon} + \mathcal{Q}_t(x_{t-1}^k) - \underline{\mathcal{Q}}_t^k(x_{t-1}^k) \text{ using (3.14) and } \varepsilon_t^k \leq \bar{\varepsilon}, \\ & \leq & \bar{\varepsilon} + \mathcal{Q}_t(x_{t-1}^k) - \underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k) \text{ by monotonicity}, \\ & \leq & \bar{\varepsilon} + \bar{\delta} + \mathcal{Q}_t(x_{t-1}^k) - c_t^T x_t^k - \mathcal{Q}_{t+1}^{k-1}(x_t^k) \text{ by definition of } x_t^k, \\ & = & \bar{\varepsilon} + \bar{\delta} + \underbrace{\mathcal{Q}_t(x_{t-1}^k) - c_t^T x_t^k - \mathcal{Q}_{t+1}(x_t^k)}_{\leq 0 \text{ by definition of } \mathcal{Q}_t} + \mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_{t+1}^{k-1}(x_t^k), \\ & \leq & \bar{\varepsilon} + \bar{\delta} + \mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_{t+1}^{k-1}(x_t^k). \end{array} \tag{3.26}$$

Using (3.25) and applying Lemma 3.1 to $x^k = x_t^k$, $f^k = \mathcal{Q}_{t+1}^k$, $f = \mathcal{Q}_{t+1}$, we obtain

$$\varlimsup_{k\to+\infty} \mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_{t+1}^{k-1}(x_t^k) \leq (\bar{\varepsilon} + \bar{\delta})(T - t). \tag{3.27}$$

Combining (3.26) and (3.27) we obtain (3.22) which achieves the proof of (i).

Since $(x_1^k, x_2^k, \ldots, x_T^k)$ is feasible for (3.4) we have $\mathcal{Q}_1(x_0) \leq \sum_{t=1}^T c_t^T x_t^k$. Using (3.26) we deduce

$$\begin{array}{lll} \mathcal{Q}_1(x_0) \leq \displaystyle\sum_{t=1}^T c_t^T x_t^k & \leq & (\bar{\varepsilon} + \bar{\delta})T + \displaystyle\sum_{t=1}^T \mathcal{Q}_t(x_{t-1}^k) - \mathcal{Q}_{t+1}^{k-1}(x_t^k) \\ & = & (\bar{\varepsilon} + \bar{\delta})T + \displaystyle\sum_{t=1}^T \Big[\mathcal{Q}_t(x_{t-1}^k) - \mathcal{Q}_{t+1}(x_t^k)\Big] + \displaystyle\sum_{t=1}^T \Big[\mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_{t+1}^{k-1}(x_t^k)\Big] \\ & = & (\bar{\varepsilon} + \bar{\delta})T + \mathcal{Q}_1(x_0) + \displaystyle\sum_{t=1}^T \Big[\mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_{t+1}^{k-1}(x_t^k)\Big]. \end{array} \tag{3.28}$$

Recalling that relation (3.27) holds for $t = 1, \ldots, T$, and passing to the limit in (3.28), we obtain (ii).

Finally,

$$\begin{array}{lll} 0 \leq \mathcal{Q}_1(x_0) - \underline{\mathcal{Q}}_1^k(x_0) & \leq & \mathcal{Q}_1(x_0) - \underline{\mathcal{Q}}_1^{k-1}(x_0) \\ & \leq & \bar{\delta} + \mathcal{Q}_1(x_0) - c_1^T x_1^k - \mathcal{Q}_2^{k-1}(x_1^k) \\ & \leq & \bar{\delta} + \mathcal{Q}_2(x_1^k) - \mathcal{Q}_2^{k-1}(x_1^k). \end{array} \tag{3.29}$$

Using (3.29) and relation (3.27) with $t = 1$, we obtain (iii). □

When all subproblems are solved exactly, i.e., when $\bar{\varepsilon} = \bar{\delta} = 0$, Theorem 3.2 shows that the sequences of upper bounds $\sum_{t=1}^T c_t^T x_t^k$ and of lower bounds $\underline{\mathcal{Q}}_1^k(x_0)$ converge to the optimal value of (3.4) and that any accumultation point of the sequence $(x_1^k, \ldots, x_T^k)$ is an optimal solution of (3.4). Therefore, in this situation, IDDP-LP can stop when $\sum_{t=1}^T c_t^T x_t^k - \underline{\mathcal{Q}}_1^k(x_0) \leq$ Tol for some parameter Tol> 0, in which case, a Tol-optimal solution to (3.4) has been found.

More generally, when noises are vanishing, i.e., when $\lim_{k\to+\infty} \varepsilon_t^k = \lim_{k\to+\infty} \delta_t^k = 0$, we can show that IDDP-LP solves (3.4) in a finite number of iterations:

Theorem 3.3 (Convergence of IDDP-LP with asymptotically vanishing noises)**.** *Consider the sequence of decisions $(x_1^k, \ldots, x_T^k)_k$ computed along the iterations of IDDP-LP. Let Assumption (H1-D) hold. Assume that all subproblems in the forward and backward passes of IDDP-LP are solved using an algorithm that necessarily outputs an extremal point of the feasible set, for instance the simplex algorithm. If $\lim_{k\to+\infty} \varepsilon_t^k = \lim_{k\to+\infty} \delta_t^k = 0$ for all t, then there is $k_0 \in \mathbb{N}$ such that for every $k \geq k_0$, $(x_1^k, x_2^k, \ldots, x_T^k)$ is an optimal solution of* (3.4)*.*

Proof. Recalling that x_1^k is an extremal point of $X_1(x_0)$ and that Assumption (H1-D) holds, IDDP-LP can only generate a finite number of different x_1^k. For each such x_1^k, $X_2(x_1^k)$ has a finite number of extremal points and x_2^k is one of these points. Therefore IDDP-LP can only generate a finite number of different x_2^k. By induction, the number of different trial points $x_1^k, x_2^k, \ldots, x_T^k$ is finite. Similarly, only a finite number of different functions $\mathcal{Q}_T^k$ can be generated (because the cut coefficients for $\mathcal{Q}_T$ are extremal points of a bounded polyhedron). For each of these functions, a finite number of different functions $\mathcal{Q}_{T-1}^k$ can be computed. Indeed, the number of different trial points x_{T-1}^k is finite and the cut coefficients λ_{T-1}^k for $\mathcal{Q}_{T-1}$ are extremal points of a bounded polyhedron. Therefore we get a finite number of cuts $\underline{\mathcal{Q}}_{T-1}^k(x_{T-1}^k)+\langle \lambda_{T-1}^k, x_{T-1}-x_{T-1}^k\rangle$. By induction, only a finite number of different functions $\mathcal{Q}_t^k, t=2,\ldots,T$, can be generated. Therefore, after some iteration k_1, every optimization subproblem solved in the forward and backward passes is a copy of an optimization problem solved previously. It follows that after some iteration k_0 all subproblems are solved exactly (optimal solutions are computed for all subproblems) and functions $\mathcal{Q}_t^k$ do not change anymore. Consequently, from iteration k_0 on, to achieve the proof, we can apply the arguments of the proof of convergence of (exact) DDP (see Theorem 6.1 in [9]). □

Remark 3.4. *[Choice of parameters δ_t^k and ε_t^k] Recalling our convergence analysis and what motivates IDDP-LP, it makes sense to choose for δ_t^k and ε_t^k sequences which decrease with k and which, for fixed k, decrease with t. A simple rule consists in defining relative errors, as long as a solver handling such errors is used to solve the problems of the forward and backward passes. Let the relative error for step t and iteration k be* `Rel_Err`$_t^k$*. We should take δ_1^k negligible (for instance 10^{-12}) to compute a valid lower bound in the first stage of the forward pass. We propose to use the relative error*

$$\texttt{Rel\_Err}_t^k = \frac{1}{k}\Big[\overline{\varepsilon} - \Big(\frac{\overline{\varepsilon}-\varepsilon_0}{T-2}\Big)(t-2)\Big], \tag{3.30}$$

for step $t \geq 2$ and iteration $k \geq 1$ (in both the forward and backward passes), which induces corresponding δ_t^k and ε_t^k for $t \geq 2, k \geq 1$. With this choice, for fixed k, the relative error linearly decreases with t: it is maximal for $t=2$ (equal to some parameter $0<\overline{\varepsilon}/k<1$, with for instance $\overline{\varepsilon}=0.1$) and minimal for $t=T$ (equal to some parameter $0<\varepsilon_0/k<1$, with for instance $\varepsilon_0=10^{-12}$).

If one wishes to provide absolute errors instead of relative errors to the solvers, we need a guess on the optimal values of the linear programs solved in the backward and forward passes. In this situation, we propose to take as an estimation of $\underline{\mathcal{Q}}_t^k(x_{t-1}^k)$ the value $\underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k)$ which is available before solving (3.12) since it was computed in the forward pass of iteration k. We take all δ_t^k negligible and define the absolute errors

$$\varepsilon_t^k = \max\Big(1, \Big|\underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k)\Big|\Big)\texttt{Rel\_Err}_t^k \tag{3.31}$$

where `Rel_Err`$_t^k$ *is given by* (3.30).

4. Inexact cuts in SDDP applied to linear programs

4.1. **Problem formulation assumptions, and algorithm.** We are interested in solution methods for linear Stochastic Dynamic Programming equations: the first stage problem is

$$\mathcal{Q}_1(x_0) = \left\{ \begin{array}{l} \min_{x_1 \in \mathbb{R}^n} c_1^T x_1 + \mathcal{Q}_2(x_1) \\ A_1 x_1 + B_1 x_0 = b_1, x_1 \geq 0 \end{array} \right. \tag{4.32}$$

for x_0 given and for $t=2,\ldots,T$, $\mathcal{Q}_t(x_{t-1}) = \mathbb{E}_{\xi_t}[\mathfrak{Q}_t(x_{t-1},\xi_t)]$ with

$$\mathfrak{Q}_t(x_{t-1},\xi_t) = \left\{ \begin{array}{l} \min_{x_t \in \mathbb{R}^n} c_t^T x_t + \mathcal{Q}_{t+1}(x_t) \\ A_t x_t + B_t x_{t-1} = b_t, x_t \geq 0, \end{array} \right. \tag{4.33}$$

with the convention that $\mathcal{Q}_{T+1}$ is null and where for $t=2,\ldots,T$, random vector ξ_t corresponds to the concatenation of the elements in random matrices A_t, B_t which have a known finite number of rows and random vectors b_t, c_t. Moreover, it is assumed that ξ_1 is not random. For convenience, we will denote

$$X_t(x_{t-1},\xi_t) := \{x_t \in \mathbb{R}^n : A_t x_t + B_t x_{t-1} = b_t,\, x_t \geq 0\}.$$

We make the following assumptions:

(H1-S) The random vectors $\xi_2,\ldots,\xi_T$ are independent and have discrete distributions with finite support.

(H2-S) The set $X_1(x_0,\xi_1)$ is nonempty and bounded and for every $x_1 \in X_1(x_0,\xi_1)$, for every $t=2,\ldots,T$, for every realization $\tilde{\xi}_2,\ldots,\tilde{\xi}_t$ of $\xi_2,\ldots,\xi_t$, for every $x_\tau \in X_\tau(x_{\tau-1},\tilde{\xi}_\tau), \tau=2,\ldots,t-1$, the set $X_t(x_{t-1},\tilde{\xi}_t)$ is nonempty and bounded.

We put $\Theta_1 = \{\xi_1\}$ and for $t \geq 2$ we will denote by $\Theta_t = \{\xi_{t1}, \ldots, \xi_{tM_t}\}$ the support of ξ_t for stage t with $p_{ti} = \mathbb{P}(\xi_t = \xi_{ti}) > 0, i = 1, \ldots, M_t$ and with vector ξ_{tj} being the concatenation of the elements in $A_{tj}, B_{tj}, b_{tj}, c_{tj}$.

Inexact SDDP applied to linear Stochastic Dynamic Programming equations (4.32), (4.33) is a simple extension of SDDP, called ISDDP-LP, where the subproblems of the forward and backward passes are solved approximately. At iteration k, for $t = 2, \ldots, T$, function $\mathcal{Q}_t$ is approximated by a piecewise affine lower bounding function $\mathcal{Q}_t^k$ which is a maximum of affine lower bounding functions $\mathcal{C}_t^i$ called inexact cuts:

$$\mathcal{Q}_t^k(x_{t-1}) = \max_{1 \leq i \leq k} \mathcal{C}_t^i(x_{t-1}) \text{ with } \mathcal{C}_t^i(x_{t-1}) = \theta_t^i + \langle \beta_t^i, x_{t-1} \rangle$$

where coefficients θ_t^i, β_t^i are computed as explained below. The steps of ISDDP-LP are as follows.

ISDDP-LP, Step 1: Initialization. For $t = 2, \ldots, T$, take for $\mathcal{C}_t^0 = \mathcal{Q}_t^0$ a known lower bounding affine function for $\mathcal{Q}_t$. Set the iteration count k to 1 and $\mathcal{Q}_{T+1}^0 \equiv 0$.

ISDDP-LP, Step 2: Forward pass. We generate a sample $\tilde{\xi}^k = (\tilde{\xi}_1^k, \tilde{\xi}_2^k, \ldots, \tilde{\xi}_T^k)$ from the distribution of $\xi^k \sim (\xi_1, \xi_2, \ldots, \xi_T)$, with the convention that $\tilde{\xi}_1^k = \xi_1$. Using approximation $\mathcal{Q}_{t+1}^{k-1}$ of $\mathcal{Q}_{t+1}$ (computed at previous iterations), we compute a δ_t^k-optimal basic feasible solution x_t^k of the problem

$$\left\{ \begin{array}{l} \min_{x_t \in \mathbb{R}^n} x_t^T \tilde{c}_t^k + \mathcal{Q}_{t+1}^{k-1}(x_t) \\ x_t \in X_t(x_{t-1}^k, \tilde{\xi}_t^k) \end{array} \right. \tag{4.34}$$

for $t = 1, \ldots, T$, where $x_0^k = x_0$ and where $\tilde{c}_t^k$ is the realization of c_t in $\tilde{\xi}_t^k$. For $k \geq 1$ and $t = 1, \ldots, T$, define the function $\underline{\mathfrak{Q}}_t^k : \mathbb{R}^n \times \Theta_t \to \overline{\mathbb{R}}$ by

$$\underline{\mathfrak{Q}}_t^k(x_{t-1}, \xi_t) = \left\{ \begin{array}{l} \min_{x_t \in \mathbb{R}^n} c_t^T x_t + \mathcal{Q}_{t+1}^k(x_t) \\ x_t \in X_t(x_{t-1}, \xi_t). \end{array} \right. \tag{4.35}$$

With this notation, we have

$$\underline{\mathfrak{Q}}_t^{k-1}(x_{t-1}^k, \tilde{\xi}_t^k) \leq \langle \tilde{c}_t^k, x_t^k \rangle + \mathcal{Q}_{t+1}^{k-1}(x_t^k) \leq \underline{\mathfrak{Q}}_t^{k-1}(x_{t-1}^k, \tilde{\xi}_t^k) + \delta_t^k. \tag{4.36}$$

ISDDP-LP, Step 3: Backward pass. The backward pass builds inexact cuts for $\mathcal{Q}_t$ at x_{t-1}^k computed in the forward pass. For $t = T+1$, we have $\mathcal{Q}_t^k = \mathcal{Q}_{T+1}^k \equiv 0$, i.e., θ_{T+1}^k and β_{T+1}^k are null. For $j = 1, \ldots, M_T$, we solve approximately the problem

$$\mathfrak{Q}_T(x_{T-1}^k, \xi_{Tj}) = \left\{ \begin{array}{l} \min\limits_{x_T \in \mathbb{R}^n} c_{Tj}^T x_T \\ A_{Tj} x_T + B_{Tj} x_{T-1}^k = b_{Tj}, x_T \geq 0, \end{array} \right. \text{ with dual } \left\{ \begin{array}{l} \max_\lambda \lambda^T (b_{Tj} - B_{Tj} x_{T-1}^k) \\ A_{Tj}^T \lambda \leq c_{Tj}. \end{array} \right. \tag{4.37}$$

Let λ_{Tj}^k be an ε_T^k-optimal basic feasible solution of the dual problem above: $A_{Tj}^T \lambda_{Tj}^k \leq c_{Tj}$ and

$$\mathfrak{Q}_T(x_{T-1}^k, \xi_{Tj}) - \varepsilon_T^k \leq \langle \lambda_{Tj}^k, b_{Tj} - B_{Tj} x_{T-1}^k \rangle \leq \mathfrak{Q}_T(x_{T-1}^k, \xi_{Tj}). \tag{4.38}$$

We compute

$$\theta_T^k = \sum_{j=1}^{M_T} p_{Tj} \langle b_{Tj}, \lambda_{Tj}^k \rangle \text{ and } \beta_T^k = -\sum_{j=1}^{M_T} p_{Tj} B_{Tj}^T \lambda_{Tj}^k. \tag{4.39}$$

Using Proposition 2.1 we have that $\mathcal{C}_T^k(x_{T-1}) = \theta_T^k + \langle \beta_T^k, x_{T-1} \rangle$ is an inexact cut for $\mathcal{Q}_T$ at x_{T-1}^k. Using (4.38), we also see that

$$\mathcal{Q}_T(x_{T-1}^k) - \mathcal{C}_T^k(x_{T-1}^k) \leq \varepsilon_T^k. \tag{4.40}$$

Then for $t = T-1$ down to $t = 2$, knowing $\mathcal{Q}_{t+1}^k \leq \mathcal{Q}_{t+1}$, for $j = 1, \ldots, M_t$, consider the optimization problem

$$\underline{\mathfrak{Q}}_t^k(x_{t-1}^k, \xi_{tj}) = \left\{ \begin{array}{l} \min\limits_{x_t} c_{tj}^T x_t + \mathcal{Q}_{t+1}^k(x_t) \\ x_t \in X_t(x_{t-1}^k, \xi_{tj}) \end{array} \right. = \left\{ \begin{array}{l} \min\limits_{x_t, f} c_{tj}^T x_t + f \\ A_{tj} x_t + B_{tj} x_{t-1}^k = b_{tj}, x_t \geq 0, \\ f \geq \theta_{t+1}^i + \langle \beta_{t+1}^i, x_t \rangle, i = 1, \ldots, k, \end{array} \right. \tag{4.41}$$

with optimal value $\underline{\mathfrak{Q}}_t^k(x_{t-1}^k, \xi_{tj})$. Observe that due to (H2-S) the above problem is feasible and has a finite optimal value. Therefore $\underline{\mathfrak{Q}}_t^k(x_{t-1}^k, \xi_{tj})$ can be expressed as the optimal value of the corresponding dual problem:

$$\text{(4.42)} \qquad \underline{\mathfrak{Q}}_t^k(x_{t-1}^k, \xi_{tj}) = \begin{cases} \max_{\lambda,\mu} \lambda^T(b_{tj} - B_{tj}x_{t-1}^k) + \sum_{i=1}^k \mu_i \theta_{t+1}^i \\ A_{tj}^T \lambda + \sum_{i=1}^k \mu_i \beta_{t+1}^i \leq c_{tj}, \ \sum_{i=1}^k \mu_i = 1, \\ \mu_i \geq 0, \ i = 1, \ldots, k. \end{cases}$$

Let $(\lambda_{tj}^k, \mu_{tj}^k)$ be an ε_t^k-optimal basic feasible solution of dual problem (4.42) and let $\underline{\mathcal{Q}}_t^k$ be the function given by $\underline{\mathcal{Q}}_t^k(x_{t-1}) = \sum_{j=1}^{M_t} p_{tj} \underline{\mathfrak{Q}}_t^k(x_{t-1}, \xi_{tj})$. We compute

$$\text{(4.43)} \qquad \theta_t^k = \sum_{j=1}^{M_t} p_{tj} \Big(\langle \lambda_{tj}^k, b_{tj} \rangle + \langle \mu_{tj}^k, \theta_{t+1,k} \rangle \Big) \text{ and } \beta_t^k = -\sum_{j=1}^{M_t} p_{tj} B_{tj}^T \lambda_{tj}^k,$$

where vector $\theta_{t+1,k}$ has components $\theta_{t+1}^i, i = 1, \ldots, k$, arranged in the same order as components $\mu_{tj}^k(i), i = 1, \ldots, k$, of μ_{tj}^k. Setting $\mathcal{C}_t^k(x_{t-1}) = \theta_t^k + \langle \beta_t^k, x_{t-1} \rangle$ and using Proposition 2.1, we have

$$\text{(4.44)} \qquad \underline{\mathcal{Q}}_t^k(x_{t-1}) \geq \mathcal{C}_t^k(x_{t-1}) \quad \text{and} \quad \underline{\mathcal{Q}}_t^k(x_{t-1}^k) - \mathcal{C}_t^k(x_{t-1}^k) \leq \varepsilon_t^k.$$

Using the fact that $\mathcal{Q}_{t+1}^k(x_{t-1}) \leq \mathcal{Q}_{t+1}(x_{t-1})$, we have $\underline{\mathfrak{Q}}_t^k(x_{t-1}, \xi_{tj}) \leq \mathfrak{Q}_t(x_{t-1}, \xi_{tj})$, $\underline{\mathcal{Q}}_t^k(x_{t-1}) \leq \mathcal{Q}_t(x_{t-1})$, and therefore

$$\text{(4.45)} \qquad \mathcal{Q}_t(x_{t-1}) \geq \mathcal{C}_t^k(x_{t-1})$$

which shows that $\mathcal{C}_t^k$ is an inexact cut for $\mathcal{Q}_t$.

ISDDP-LP, Step 4: Do $k \leftarrow k + 1$ and go to Step 2.

Similarly to IDDP-LP, the collection of distinct values $(\theta_t^k, \beta_t^k)_k$ is finite and cut coefficients $(\theta_t^k, \beta_t^k)_k$ are uniformly bounded.

4.2. **Convergence analysis.** In this section we state a convergence result for ISDDP-LP in Theorem 4.1 when noises $\delta_t^k, \varepsilon_t^k$ are bounded and in Theorem 4.2 when these noises asymptotically vanish.

We will assume that the sampling procedure in ISDDP-LP satisfies the following property:

(H3-S) The samples in the backward passes are independent: $(\tilde{\xi}_2^k, \ldots, \tilde{\xi}_T^k)$ is a realization of $\xi^k = (\xi_2^k, \ldots, \xi_T^k) \sim (\xi_2, \ldots, \xi_T)$ and $\xi^1, \xi^2, \ldots,$ are independent.

Before stating our first convergence theorem, we need more notation. Due to Assumption (H1-S), the realizations of $(\xi_t)_{t=1}^T$ form 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 (with ξ_1 deterministic). We denote by $\mathcal{N}$ the set of nodes and for a node n of the tree, we define:

- $C(n)$: the set of children nodes (the empty set for the leaves);
- x_n: a decision taken at that node;
- p_n: the transition probability from the parent node of n to n;
- ξ_n: the realization of process (ξ_t) at node n[1]: for a node n of stage t, 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.

Next, we define for iteration k decisions x_n^k for all node n of the scenario tree simulating the policy obtained in the end of iteration $k-1$ replacing cost-to-go function $\mathcal{Q}_t$ by $\mathcal{Q}_t^{k-1}$ for $t = 2, \ldots, T+1$:

[1]The same notation $\xi_{\texttt{Index}}$ is used to denote the realization of the process at node `Index` of the scenario tree and the value of the process (ξ_t) for stage `Index`. The context will allow us to know which concept is being referred to. In particular, letters n and m will only be used to refer to nodes while t will be used to refer to stages.

Simulation of the policy in the end of iteration $k-1$.

For $t = 1, \ldots, T$,

For every node n of stage $t-1$,

For every child node m of node n, compute a δ_t^k-optimal solution x_m^k of

$$\underline{\mathfrak{Q}}_t^{k-1}(x_n^k, \xi_m) = \left\{ \begin{array}{l} \inf_{x_m} \; c_m^T x_m + \mathcal{Q}_{t+1}^{k-1}(x_m) \\ x_m \in X_t(x_n^k, \xi_m), \end{array} \right. \tag{4.46}$$

where $x_{n_0}^k = x_0$.

End For

End For

End For

We are now in a position to state our first convergence theorem for ISDDP-LP:

Theorem 4.1 (Convergence of ISDDP-LP with bounded noises)**.** *Consider the sequences of decisions $(x_n^k)_{n \in \mathcal{N}}$ and of functions $(\mathcal{Q}_t^k)$ generated by ISDDP-LP. Assume that (H1-S), (H2-S), (H3-S) hold, and that noises ε_t^k and δ_t^k are bounded: $0 \le \varepsilon_t^k \le \bar{\varepsilon}$, $0 \le \delta_t^k \le \bar{\delta}$ for finite $\bar{\delta}, \bar{\varepsilon}$. Then the following holds:*

(i) *for $t = 2, \ldots, T+1$, for all node n of stage $t-1$, almost surely*

$$0 \le \varliminf_{k \to +\infty} \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) \le \varlimsup_{k \to +\infty} \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) \le (\bar{\delta} + \bar{\varepsilon})(T - t + 1); \tag{4.47}$$

(ii) *for every $t = 2, \ldots, T$, for all node n of stage $t-1$, the limit superior and limit inferior of the sequence of upper bounds $\Big(\sum_{m \in C(n)} p_m (c_m^T x_m^k + \mathcal{Q}_{t+1}(x_m^k)) \Big)_k$ satisfy almost surely*

$$\begin{array}{l} 0 \le \varliminf_{k \to +\infty} \displaystyle\sum_{m \in C(n)} p_m \Big[c_m^T x_m^k + \mathcal{Q}_{t+1}(x_m^k) \Big] - \mathcal{Q}_t(x_n^k), \\ \varlimsup_{k \to +\infty} \displaystyle\sum_{m \in C(n)} p_m \Big[c_m^T x_m^k + \mathcal{Q}_{t+1}(x_m^k) \Big] - \mathcal{Q}_t(x_n^k) \le (\bar{\delta} + \bar{\varepsilon})(T - t + 1); \end{array} \tag{4.48}$$

(iii) *the limit superior and limit inferior of the sequence of lower bounds $(\underline{\mathfrak{Q}}_1^{k-1}(x_0, \xi_1))_k$ on the optimal value $\mathcal{Q}_1(x_0)$ of (4.32) satisfy almost surely*

$$\mathcal{Q}_1(x_0) - \bar{\delta} T - \bar{\varepsilon}(T-1) \le \varliminf_{k \to +\infty} \underline{\mathfrak{Q}}_1^{k-1}(x_0, \xi_1) \le \varlimsup_{k \to +\infty} \underline{\mathfrak{Q}}_1^{k-1}(x_0, \xi_1) \le \mathcal{Q}_1(x_0). \tag{4.49}$$

Proof. The proof is a simple combination of arguments from the proof of Theorem 3.1 in [6] and of Theorem 3.2 from Section 3. For interested readers, the detailed proof is provided in the appendix. □

Theorem 4.2 below shows the convergence of ISDDP-LP in a finite number of iterations when noises $\varepsilon_t^k, \delta_t^k$ asymptotically vanish.

Theorem 4.2 (Convergence of ISDDP-LP with asymptotically vanishing noises)**.** *Consider the sequences of decisions (x_n^k) and of functions $(\mathcal{Q}_t^k)$ generated by ISDDP-LP. Let Assumptions (H1-S), (H2-S), and (H3-S) hold. If $\lim_{k \to +\infty} \delta_t^k = \lim_{k \to +\infty} \varepsilon_t^k = 0$, then ISDDP-LP converges with probability one in a finite number of iterations to an optimal solution to* (4.32), (4.33).

Proof. The arguments are similar to the proof of Theorem 3.3. Due to Assumptions (H1-S), (H2-S), ISDDP-LP generates almost surely a finite number of trial points $x_1^k, x_2^k, \ldots, x_T^k$. Similarly, almost surely only a finite number of different functions $\mathcal{Q}_t^k, t = 2, \ldots, T$, can be generated. Therefore, after some iteration k_1, every optimization subproblem solved in the forward and backward passes is a copy of an optimization problem solved previously. It follows that after some iteration k_0 all subproblems are solved exactly (optimal solutions are computed for all subproblems) and functions $\mathcal{Q}_t^k$ do not change anymore. Consequently, from iteration k_0 on, we can apply the arguments of the proof of convergence of (exact) SDDP applied to linear programs (see Theorem 5 in [20]). □

Remark 4.3. *[Choice of parameters δ_t^k and ε_t^k] As for IDDP-LP, we take δ_1^k is negligible (for instance 10^{-12}) and the relative error*

$$\texttt{Rel\_Err}_t^k = \frac{1}{k}\Big[\overline{\varepsilon} - \left(\frac{\overline{\varepsilon}-\varepsilon_0}{T-2}\right)(t-2)\Big], \tag{4.50}$$

for step $t \geq 2$ and iteration $k \geq 1$ (in both the forward and backward passes), which induces corresponding δ_t^k and ε_t^k for $t \geq 2, k \geq 1$.

However, it seems more delicate to define sound absolute errors. Ultimately, absolute errors (3.31) *used for IDDP-LP could be replaced by*

$$\varepsilon_t^k = \max\left(1, \left|\underline{\mathfrak{Q}}_t^{k-1}(x_{t-1}^k, \tilde{\xi}_t^k)\right|\right)\texttt{Rel\_Err}_t^k \tag{4.51}$$

with $\texttt{Rel\_Err}_t^k$ *still given by* (4.50)*.*

5. Numerical experiments

Our goal in this section is to compare SDDP and ISDDP-LP (denoted for short ISDDP in what follows) on the risk-neutral portfolio problem with direct transaction costs presented in Section 5.1 of [14] (see [14] for details). For this application, ξ_t is the vector of asset returns: if n is the number of risky assets, ξ_t has size $n+1$, $\xi_t(1:n)$ is the vector of risky asset returns for stage t while $\xi_t(n+1)$ is the return of the risk-free asset. We generate four instances of this portfolio problem as follows.

For fixed T (number of stages) and n (number of risky assets), the distributions of $\xi_t(1:n), t = 2, \ldots, T$, have M realizations with $p_{ti} = \mathbb{P}(\xi_t = \xi_{ti}) = 1/M$, and $\xi_1(1:n), \xi_{t1}(1:n), \ldots, \xi_{tM}(1:n)$ obtained sampling from a normal distribution with mean and standard deviation chosen randomly in respectively the intervals $[0.9, 1.4]$ and $[0.1, 0.2]$. The monthly return $\xi_t(n+1)$ of the risk-free asset is 1.01 for all t. The initial portfolio x_0 has components uniformly distributed in $[0, 10]$ (vector of initial wealth in each asset). The largest possible position in any security is set to $u_i = 20\%$. Transaction costs are known with $\nu_t(i) = \mu_t(i)$ obtained sampling from the distribution of the random variable $0.08 + 0.06\cos(\frac{2\pi}{T}U_T)$ where U_T is a random variable with a discrete distribution over the set of integers $\{1, 2, \ldots, T\}$. Our four instances of the portfolio problem are obtained taking for (M, T, n) the combinations of values $(100, 10, 50)$, $(100, 30, 50)$, $(50, 20, 50)$, and $(50, 40, 10)$. All linear subproblems of the forward and backward passes are solved numerically using Mosek solver [1] and for ISDDP, we solve approximately these subproblems limiting the number of iterations of Mosek solver as indicated in Table 2 in the Appendix. The strategy given in this table is (as indicated in Remark 4.3) to increase the accuracy (or, equivalently, increase the maximal number of iterations allowed for Mosek solver) of the solutions to subproblems as ISDDP iteration increases and for a given iteration of ISDDP, to increase the accuracy (or, equivalently, increase the maximal number of iterations allowed for Mosek solver) of the solutions to subproblems as the number of stages increases from $t = 2$ to $t = T$, knowing that we solve exactly the subproblems for the last stage T and for the first stage $t = 1$.

SDDP and ISDDP were implemented in Matlab and the code was run on a Xeon E5-2670 processor with 384 GB of RAM. For a given instance, SDDP and ISDDP were run using the same set of sampled scenarios along iterations. We stopped SDDP algorithm when the gap is $< 10\%$ and run ISDDP for the same number of iterations.[2]

On our four instances, we then simulate the policies obtained with SDDP and ISDDP on a set of 500 scenarios of returns. The gap between the two policies on these scenarios and the CPU time reduction using ISDDP are given in Table 1. In this table, the gap is defined by $100\frac{\texttt{CostISDDP}-\texttt{CostSDDP}}{\texttt{CostSDDP}}$ where `CostISDDP` and `CostSDDP` are respectively the mean cost for ISDDP and SDDP policies on the 500 simulated scenarios and the CPU time reduction is given by $100\frac{\texttt{TimeSDDP}-\texttt{TimeISDDP}}{\texttt{TimeSDDP}}$ where `TimeSDDP` and `TimeISDDP` correspond to the time needed to compute SDDP and ISDDP policies (before running the Monte Carlo simulation), respectively.

On all instances the gap is relatively small and ISDDP policy is computed faster than SDDP policy.

[2]The gap is defined as $\frac{Ub-Lb}{Ub}$ where Ub and Lb correspond to upper and lower bounds, respectively. Though the portfolio problem is a maximization problem (of the mean income), we have rewritten it as a minimization problem (of the mean loss), of form (4.32), (4.33). 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 for $N = 100$ policy realizations, see [21] for a detailed discussion on this stopping criterion.

M	T	n	Gap (%)	CPU time reduction (%)
50	20	50	0.1	6.2
50	40	10	4.2	11.1
100	10	50	0.8	6.5
100	30	50	3.4	6.4

Table 1. Empirical gap between SDDP and ISDDP policies and CPU time reduction for ISDDP over SDDP.

More precisely, we report in Figure 1 (for instances with $(M, T, n) = (100, 10, 50)$ and $(M, T, n) = (100, 30, 50)$) and Figure 2 (for instances with $(M, T, n) = (50, 20, 50)$ and $(M, T, n) = (50, 40, 10)$) three outputs along the iterations of SDDP and ISDDP: the cumulative CPU time (in seconds), the number of iterations needed for Mosek LP solver to solve all backward and forward subproblems, and the upper and lower bounds on the optimal value computed by the methods (note that the upper bounds are only computed from iteration 100 on, because the past $N = 100$ iterations are used to compute them).

These experiments (i) show that it is possible to obtain a near optimal policy quicker than SDDP solving approximately some subproblems in SDDP and (ii) confirm that ISDDP computes a valid lower bound since first stage subproblems are solved exactly. For the first iterations, this lower bound can however be distant from SDDP lower bound (see for instance the bottom left plots of Figures 1 and 2). However, both SDDP and ISDDP lower and upper bounds are quite close after 200 iterations, even if Mosek LP solver uses much less iterations to solve the subproblems with ISDDP (see the middle plots of Figures 1, 2). The total CPU time needed by ISDDP is significantly inferior but this CPU time reduction decreases when the number of iterations increases. If many iterations are required to solve the problem, after a few hundreds iterations backward and forward subproblems are solved in similar CPU time for SDDP and ISDDP and the total CPU time reduction starts to stabilize.

6. Conclusion

We introduced IDDP-LP and ISDDP-LP, the first inexact variants of DPP and SDDP applied to respectively linear programs and multistage stochastic linear programs. We studied the convergence of IDDP-LP and ISDDP-LP and presented the results of numerical experiments comparing the computational bulk of SDDP and ISDDP-LP on a portfolio problem.

Since ISDDP-LP can be much quicker than SDDP for some well chosen parameters $(\delta_t^k, \varepsilon_t^k)$ and is straightforward to implement from SDDP, it would be interesting to use ISDDP-LP on other real-life applications modelled by multistage stochastic linear programs.

As a continuation of this work, it would also be interesting to consider a variant of SDDP that builds cuts in the backward pass on the basis of approximate solutions which are not necessarily feasible (but of course asymptotically feasible to derive a convergence result).

Acknowledgments

The author's research was partially supported by an FGV grant, CNPq grants 311289/2016-9 and 401371/2014-0, and FAPERJ grant E-26/201.599/2014.

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] T. Asamov and W. Powell. Regularized decomposition of high-dimensional multistage stochastic programs with markov uncertainty. *Available at:* `https://arxiv.org/abs/1505.02227`, 2015.

[3] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[4] J. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer-Verlag, New York, 1997.

[5] J.R. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Oper. Res.*, 33:989–1007, 1985.

[6] P. Girardeau, V. Leclere, and A.B. Philpott. On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, 40:130–145, 2015.

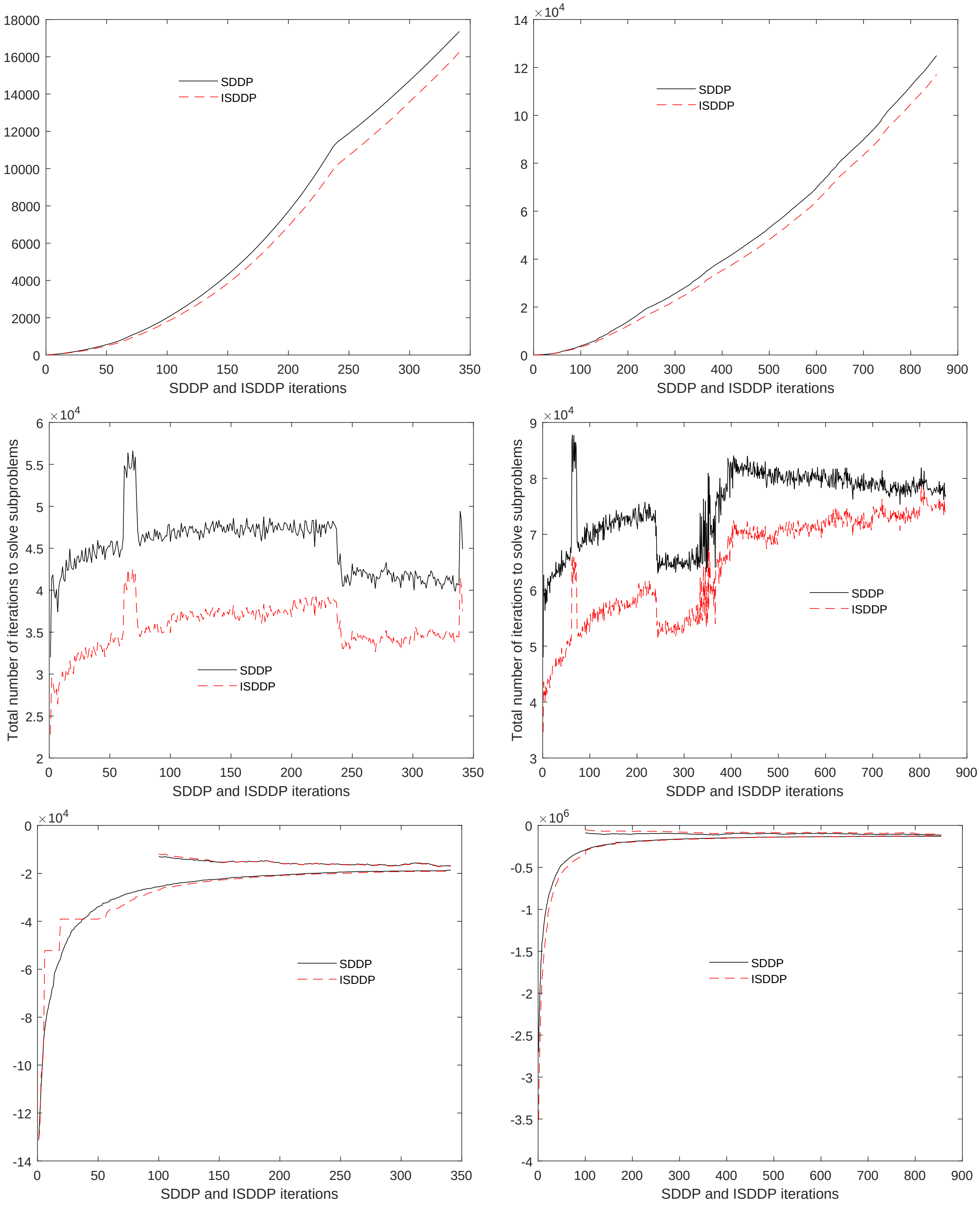

FIGURE 1. Top plots: cumulative CPU time (in seconds), middle plots: total number of iterations to solve subproblems, bottom plots: upper and lower bounds. Left plots: $M = 100$, $T = 10$, $n = 50$, right plots: $M = 100$, $T = 30$, and $n = 50$.

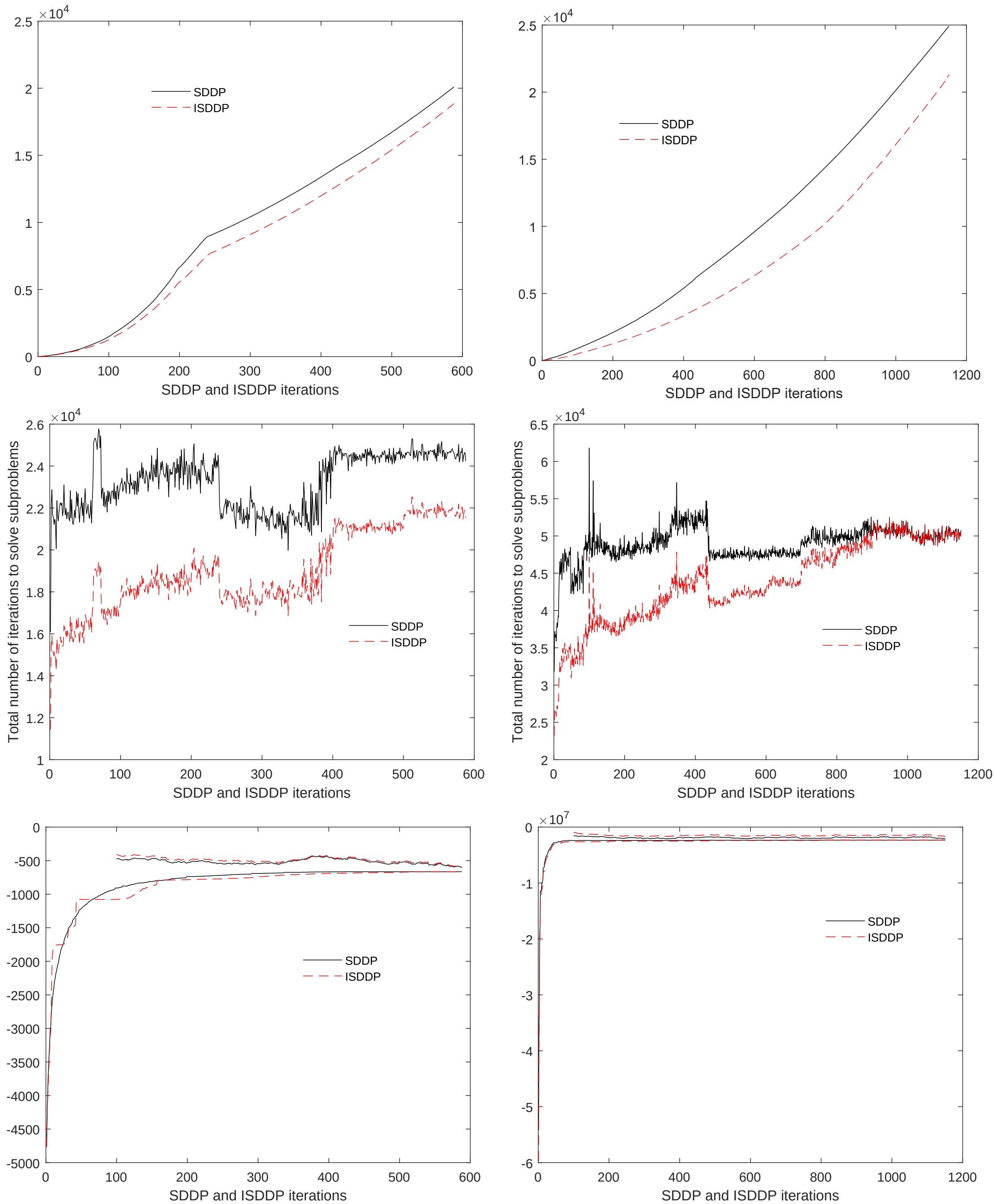

FIGURE 2. Top plots: cumulative CPU time (in seconds), middle plots: total number of iterations to solve subproblems, bottom plots: upper and lower bounds. Left plots: $M = 50$, $T = 20$, $n = 50$, right plots: $M = 50$, $T = 40$, and $n = 10$.

[7] V. Guigues. SDDP for some interstage dependent risk-averse problems and application to hydro-thermal planning. *Computational Optimization and Applications*, 57:167–203, 2014.
[8] V. Guigues. Convergence analysis of sampling-based decomposition methods for risk-averse multistage stochastic convex programs. *SIAM Journal on Optimization*, 26:2468–2494, 2016.
[9] V. Guigues. Dual dynamic programing with cut selection: Convergence proof and numerical experiments. *European Journal of Operational Research*, 258:47–57, 2017.
[10] V. Guigues. Inexact cuts for Deterministic and Stochastic Dual Dynamic Programming applied to convex nonlinear optimization problems. *arXiv*, 2017. `https://arxiv.org/abs/1707.00812`.
[11] V. Guigues and M. Bandarra. Multicut decomposition methods with cut selection for multistage stochastic programs. *Available at https://arxiv.org/abs/1705.08977.*, 2017.
[12] V. Guigues and W. Römisch. Sampling-based decomposition methods for multistage stochastic programs based on extended polyhedral risk measures. *SIAM J. Optim.*, 22:286–312, 2012.
[13] V. Guigues and W. Römisch. SDDP for multistage stochastic linear programs based on spectral risk measures. *Oper. Res. Lett.*, 40:313–318, 2012.
[14] V. Guigues, W. Tekaya, and M. Lejeune. Regularized decomposition methods for deterministic and stochastic convex optimization and application to portfolio selection with direct transaction and market impact costs. *Optimization OnLine*, 2017.
[15] G. Infanger and D. Morton. Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming*, 75:241–256, 1996.
[16] V. Kozmik and D.P. Morton. Evaluating policies in risk-averse multi-stage stochastic programming. *Mathematical Programming*, 152:275–300, 2015.
[17] M.V.F. Pereira and L.M.V.G Pinto. Multi-stage stochastic optimization applied to energy planning. *Math. Program.*, 52:359–375, 1991.
[18] Laurent Pfeiffer, Romain Apparigliato, and Sophie Auchapt. Two methods of pruning benders' cuts and their application to the management of a gas portfolio. *Research Report RR-8133, hal-00753578*, 2012.
[19] A. Philpott, V. de Matos, and E. Finardi. Improving the performance of stochastic dual dynamic programming. *Journal of Computational and Applied Mathematics*, 290:196–208, 2012.
[20] A. B. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and related methods. *Oper. Res. Lett.*, 36:450–455, 2008.
[21] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209:63–72, 2011.
[22] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia, 2009.
[23] G. Zakeri, A. Philpott, and D. Ryan. Inexact cuts in benders decomposition. *Siam Journal on Optimization*, 10:643–657, 2000.
[24] J. Zou, S. Ahmed, and X.A. Sun. Stochastic dual dynamic integer programming. *Optimization Online*, 2017.

Appendix

Proof of Theorem 4.1.

(i) We show (4.47) for $t = 2, \ldots, T+1$, and all node n of stage $t-1$ by backward induction on t. The relation holds for $t = T+1$. Now assume that it holds for $t+1$ for some $t \in \{2, \ldots, T\}$. Let us show that it holds for t. Take a node n of stage $t-1$. Observe that the sequence $\mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k)$ is almost surely bounded and nonnegative. Therefore it has almost surely a nonnegative limit inferior and a finite limit superior. Let $\mathcal{S}_n = \{k : n_t^k = n\}$ be the iterations where the sampled scenario passes through node n. For $k \in \mathcal{S}_n$ we have

(6.52)

$$
\begin{array}{lll}
0 \leq \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) & \leq & \mathcal{Q}_t(x_n^k) - \mathcal{C}_t^k(x_n^k) \\
& \leq & \varepsilon_t^k + \mathcal{Q}_t(x_n^k) - \underline{\mathcal{Q}}_t^k(x_n^k) \\
& \leq & \bar{\varepsilon} + \displaystyle\sum_{m \in C(n)} p_m \Big[\mathfrak{Q}_t(x_n^k, \xi_m) - \underline{\mathfrak{Q}}_t^k(x_n^k, \xi_m) \Big] \\
& \leq & \bar{\varepsilon} + \displaystyle\sum_{m \in C(n)} p_m \Big[\mathfrak{Q}_t(x_n^k, \xi_m) - \underline{\mathfrak{Q}}_t^{k-1}(x_n^k, \xi_m) \Big] \\
& \leq & \bar{\varepsilon} + \delta_t^k + \displaystyle\sum_{m \in C(n)} p_m \Big[\mathfrak{Q}_t(x_n^k, \xi_m) - \langle c_m, x_m^k \rangle - \mathcal{Q}_{t+1}^{k-1}(x_m^k) \Big] \\
& \leq & \bar{\varepsilon} + \bar{\delta} + \displaystyle\sum_{m \in C(n)} p_m \Big[\underbrace{\mathfrak{Q}_t(x_n^k, \xi_m) - \langle c_m, x_m^k \rangle - \mathcal{Q}_{t+1}(x_m^k)}_{\leq 0 \text{ by definition of } \mathfrak{Q}_t \text{ and } x_m^k} + \mathcal{Q}_{t+1}(x_m^k) - \mathcal{Q}_{t+1}^{k-1}(x_m^k) \Big] \\
& \leq & \bar{\varepsilon} + \bar{\delta} + \displaystyle\sum_{m \in C(n)} p_m \Big[\mathcal{Q}_{t+1}(x_m^k) - \mathcal{Q}_{t+1}^{k-1}(x_m^k) \Big].
\end{array}
$$

Using the induction hypothesis, we have for every $m \in C(n)$ that

$$\overline{\lim}_{k\to+\infty} \mathcal{Q}_{t+1}(x_m^k) - \mathcal{Q}_{t+1}^k(x_m^k) \le (\bar\delta+\bar\varepsilon)(T-t).$$

In virtue of Lemma 3.1, this implies

$$\overline{\lim}_{k\to+\infty} \mathcal{Q}_{t+1}(x_m^k) - \mathcal{Q}_{t+1}^{k-1}(x_m^k) \le (\bar\delta+\bar\varepsilon)(T-t), \tag{6.53}$$

which, plugged into (6.52), gives

$$\overline{\lim}_{k\to+\infty, k\in\mathcal{S}_n} \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) \le (\bar\delta+\bar\varepsilon)(T-t+1). \tag{6.54}$$

Now let us show by contradiction that

$$\overline{\lim}_{k\to+\infty} \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) \le (\bar\delta+\bar\varepsilon)(T-t+1). \tag{6.55}$$

If (6.55) does not hold then there exists $\varepsilon_0>0$ such that there is an infinite set of iterations k satisfying $\mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) > (\bar\delta+\bar\varepsilon)(T-t+1)+\varepsilon_0$ and by monotonicity, there is also an infinite set of iterations k in the set $K=\{k\ge 1 : \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^{k-1}(x_n^k) > (\bar\delta+\bar\varepsilon)(T-t+1)+\varepsilon_0\}$. Let $k_1<k_2<...$ be these iterations: $K=\{k_1,k_2,\ldots,\}$. Let y_n^k be the random variable which takes the value 1 if $k\in\mathcal{S}_n$ and 0 otherwise. Due to Assumption (H3-S), random variables $y_n^{k_1}, y_n^{k_2},\ldots,$ are i.i.d. and have the distribution of y_n^1. Therefore by the Strong Law of Large Numbers we get

$$\frac{1}{N}\sum_{j=1}^N y_n^{k_j} \xrightarrow{N\to+\infty} \mathbb{E}[y_n^1] > 0 \text{ a.s.}$$

Now let $z_1<z_2<\ldots$ be the iterations in $\mathcal{S}_n$: $\mathcal{S}_n=\{z_1,z_2,\ldots\}$. Relation (6.54) can be written

$$\overline{\lim}_{k\to+\infty} \mathcal{Q}_t(x_n^{z_k}) - \mathcal{Q}_t^{z_k}(x_n^{z_k}) \le (\bar\delta+\bar\varepsilon)(T-t+1),$$

which, using Lemma 3.1, implies

$$\overline{\lim}_{k\to+\infty} \mathcal{Q}_t(x_n^{z_k}) - \mathcal{Q}_t^{z_{k-1}}(x_n^{z_k}) \le (\bar\delta+\bar\varepsilon)(T-t+1).$$

Using the fact that $z_k \ge z_{k-1}+1$, we deduce that

$$\begin{array}{rcl}\overline{\lim}_{k\to+\infty,k\in\mathcal{S}_n} \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^{k-1}(x_n^k) & = & \overline{\lim}_{k\to+\infty} \mathcal{Q}_t(x_n^{z_k}) - \mathcal{Q}_t^{z_k-1}(x_n^{z_k}) \\ & \le & \overline{\lim}_{k\to+\infty} \mathcal{Q}_t(x_n^{z_k}) - \mathcal{Q}_t^{z_{k-1}}(x_n^{z_k}) \le (\bar\delta+\bar\varepsilon)(T-t+1).\end{array}$$

Therefore, there can only be a finite number of iterations that are both in K and in $\mathcal{S}_n$. This gives

$$\frac{1}{N}\sum_{j=1}^N y_n^{k_j} \xrightarrow{N\to+\infty} 0 \text{ a.s.}$$

We obtain a contradiction and therefore (6.55) must hold.

(ii) Using (6.52), we obtain for every $t=2,\ldots,T+1$, and every node n of stage $t-1$, that

$$0 \le \sum_{m\in C(n)} p_m\Big[c_m^T x_m^k + \mathcal{Q}_{t+1}(x_m^k)\Big] - \mathcal{Q}_t(x_n^k) \le \bar\delta+\bar\varepsilon + \sum_{m\in C(n)} p_m\Big[\mathcal{Q}_{t+1}(x_m^k) - \mathcal{Q}_{t+1}^{k-1}(x_m^k)\Big]. \tag{6.56}$$

Therefore

$$\underline{\lim}_{k\to+\infty} \sum_{m\in C(n)} p_m\Big[c_m^T x_m^k + \mathcal{Q}_{t+1}(x_m^k)\Big] - \mathcal{Q}_t(x_n^k) \ge 0$$

and using (6.53) we get

$$\overline{\lim}_{k\to+\infty} \sum_{m\in C(n)} p_m\Big[c_m^T x_m^k + \mathcal{Q}_{t+1}(x_m^k)\Big] - \mathcal{Q}_t(x_n^k) \le (\bar\delta+\bar\varepsilon)(T-t+1).$$

(iii) We have

$$\begin{array}{rcl}\mathcal{Q}_1(x_0) \ge \underline{\mathfrak{Q}}_1^{k-1}(x_0,\xi_1) & \ge & c_1^T x_1^k + \mathcal{Q}_2^{k-1}(x_1^k) - \delta_1^k \\ & \ge & -\bar\delta + \mathcal{Q}_1(x_0) + \mathcal{Q}_2^{k-1}(x_1^k) - \mathcal{Q}_2(x_1^k).\end{array} \tag{6.57}$$

ISDDP iteration	$[1, 20]$	$[21, 50]$	$[51, 100]$
LP solver maximal number of iterations at t	$\lceil (0.4 + 0.6\frac{(t-2)}{T-2}) I_{\max} \rceil$	$\lceil (0.45 + 0.55\frac{(t-2)}{T-2}) I_{\max} \rceil$	$\lceil (0.5 + 0.5\frac{(t-2)}{T-2}) I_{\max} \rceil$
ISDDP iteration	$[101, 200]$	$[201, 300]$	$[301, 400]$
LP solver maximal number of iterations at t	$\lceil (0.55 + 0.45\frac{(t-2)}{T-2}) I_{\max} \rceil$	$\lceil (0.6 + 0.4\frac{(t-2)}{T-2}) I_{\max} \rceil$	$\lceil (0.65 + 0.35\frac{(t-2)}{T-2}) I_{\max} \rceil$
ISDDP iteration	$[401, 500]$	$[501, 600]$	$[601, 700]$
LP solver maximal number of iterations at t	$\lceil (0.7 + 0.3\frac{(t-2)}{T-2}) I_{\max} \rceil$	$\lceil (0.75 + 0.25\frac{(t-2)}{T-2}) I_{\max} \rceil$	$\lceil (0.8 + 0.2\frac{(t-2)}{T-2}) I_{\max} \rceil$
ISDDP iteration	$[701, 800]$	$[801, 900]$	> 900
LP solver maximal number of iterations at t	$\lceil (0.85 + 0.15\frac{(t-2)}{T-2}) I_{\max} \rceil$	$\lceil (0.9 + 0.1\frac{(t-2)}{T-2}) I_{\max} \rceil$	$I_{\max}$

TABLE 2. Maximal number of iterations for Mosek LP solver for solving backward and forward passes subproblems as a function of stage $t \geq 2$, ISDDP iteration, and the number $I_{\max}$ of iterations used to solve subproblems with SDDP with high accuracy. In this table, $\lceil x \rceil$ is the smallest integer larger than or equal to x.

Using (6.57) and (6.53) with $t = 1$, we obtain (iii).

Additional parameters for ISDDP. For ISDDP, the maximal number of iterations allowed for Mosek LP solver to solve subproblems along the iterations of ISDDP is given in Table 2.