Journal of Optimization Theory and Applications (2021) 189:513–559
https://doi.org/10.1007/s10957-021-01842-x

# Stochastic Dynamic Cutting Plane for Multistage Stochastic Convex Programs

**Vincent Guigues[1] · Renato D. C. Monteiro[2]**

Received: 17 July 2020 / Accepted: 19 February 2021 / Published online: 25 March 2021


**Abstract**
We introduce Stochastic Dynamic Cutting Plane (StoDCuP), an extension of the Stochastic Dual Dynamic Programming (SDDP) algorithm to solve multistage stochastic convex optimization problems. At each iteration, the algorithm builds lower bounding affine functions not only for the cost-to-go functions, as SDDP does, but also for some or all nonlinear cost and constraint functions. We show the almost sure convergence of StoDCuP. We also introduce an inexact variant of StoDCuP where all subproblems are solved approximately (with bounded errors) and show the almost sure convergence of this variant for vanishing errors. Finally, numerical experiments are presented on nondifferentiable multistage stochastic programs where Inexact StoDCuP computes a good approximate policy quicker than StoDCuP while SDDP and the previous inexact variant of SDDP combined with Mosek library to solve subproblems were not able to solve the differentiable reformulation of the problem.

**Keywords** Stochastic programming · Inexact cuts for value functions · SDDP · Inexact SDDP

**Mathematics Subject Classification** 90C15 · 90C90

## 1 Introduction

Risk-neutral multistage stochastic programs (MSPs) aim at minimizing the expected value of the total cost over a given optimization period of $T$ stages while satisfying

Communicated by Lars Grüne.

✉ Vincent Guigues
vincent.guigues@fgv.br

Renato D. C. Monteiro
renato.monteiro@isye.gatech

1 School of Applied Mathematics, FGV, Praia de Botafogo, Rio de Janeiro, Brazil

2 Georgia Institute of Technology, Atlanta, GA 30332, USA

almost surely for every stage some constraints depending on an underlying stochastic process. These optimization problems are useful for many real-life applications but are challenging to solve, see for instance [33] and references therein for a thorough discussion on MSPs. Popular solution methods for MSPs are based on decomposition techniques such as Approximate Dynamic Programming [27], Lagrangian relaxation, or Stochastic Dual Dynamic Programming (SDDP) [23]. SDDP is a sampling-based extension of [3], itself a multistage extension of the L-shaped method [35]. The SDDP method builds linearizations of the convex cost-to-go functions at trial points computed on scenarios of the underlying stochastic process generated randomly along iterations. The use of such cutting plane models for the objective function in the context of deterministic convex optimization dates back to Kelley's cutting plane method [16] and has later been extended in many variants such as subgradient [17], bundle [18,20], and level [21] variants. Kelley's algorithm was also generalized by Benders to solve [2] mixed-variables programming problems. Recently, several enhancements of SDDP have been proposed, see for instance [12,19,24,32] for risk-averse variants, [5,6,26] for convergence analysis, [34] for the application of SDDP to periodic stochastic programs, and [8,22] to speed up the convergence of the method. In particular, in [8], Inexact SDDP was proposed, which incorporates inexact cuts in SDDP (for both linear and nonlinear programs). The idea of Inexact SDDP is to allow us to solve approximately some or all primal and dual subproblems in the forward and backward passes of SDDP. This extension and the study of Inexact SDDP was motivated by the following reasons:

(i) Solving to a very high accuracy nonlinear programs can take a significant amount of time or may even be impossible whereas linear programs (of similar sizes) can be solved exactly or to high accuracy quicker. Examples of convex but challenging to solve subproblems include semidefinite programs [36], quadratically constrained quadratic programs with degenerate quadratic forms (see the numerical experiments of Sect. 5), or some high-dimensional nondifferentiable problems. For subproblems where it is difficult or impossible to get optimal solutions, if we are able to provide a feasible primal-dual solution, we should be able to derive an extension of SDDP, i.e., cuts for the cost-to-go functions, from approximate subproblem primal-dual solutions. Therefore, one has to study how to extend the SDDP algorithm to still derive valid cuts and a converging Inexact SDDP or an Inexact SDDP with controlled accuracy when only approximate primal and dual solutions are computed for nonlinear MSPs.

(ii) As explained in [8], numerical experiments (see for instance [7,10,15]) indicate that very loose cuts are computed in the first iterations of SDDP and it may be useful to compute with less accuracy these cuts for the first iterations. Using this strategy, it was shown in [8] that for several instances of a portfolio problem, Inexact SDDP can converge (i.e., satisfy the stopping criterion) quicker than SDDP.

In this paper, we extend [8] in two ways:

- A natural way of taking advantage of observation (i) above in the context of SDDP applied to nonlinear problems, consists in linearizing some or all nonlinear objective and constraint functions of the subproblems solved along the iterations of the method at the optimal solutions of these subproblems. When all nonlinear functions are linearized, all subproblems solved in the iterations of SDDP are linear programs which allows us to avoid having to solve difficult problems that cannot be solved with high accuracy. However, to the best of our knowledge, this variant of SDDP that we term as StoDCuP (Stochastic Dynamic Cutting Plane) has not been proposed and studied so far in the literature (SDDP does build linearizations for the cost-to-go functions but not for some or all of the remaining nonlinear objective and constraint functions). In this context, the goal of this paper is to propose and study StoDCuP.

- As far as (ii) is concerned, it is interesting to notice that it is easy to incorporate inexact cuts in StoDCuP (i.e., to derive an inexact variant of StoDCuP), control the quality of these cuts (see Lemma 4.1), and show the convergence of this method (see Theorem 4.3). This comes from the fact that we can easily compute a cut for the value function of a linear program (and in StoDCuP all subproblems solved are linear programs) from any feasible primal-dual solution since the corresponding dual objective is linear, see Proposition 2.1 in [8]. On the contrary, deriving valid (inexact) cuts from approximate primal-dual solutions of the subproblems solved in SDDP applied to nonlinear problems and showing the convergence of the corresponding variant of Inexact SDDP is technical and the computation of inexact cuts may require solving additional subproblems, see [8] for details. Moreover, Inexact SDDP from [8] applies to differentiable multistage convex stochastic programs while both StoDCuP and Inexact StoDCuP apply to more general differentiable or nondifferentiable multistage convex stochastic programs.

The outline of the paper is the following. To ease the presentation and analysis of StoDCuP, we start in Sect. 2 with its deterministic counterpart, called DCuP (Dynamic Cutting Plane) which solves convex Dynamic Programming equations linearizing cost-to-go, constraint, and objective functions. Starting with the deterministic case allows us to focus on the differences between traditional Dual Dynamic Programming and its convergence analysis with DCuP and its convergence analysis. In Sect. 3, we introduce forward StoDCuP and prove the almost sure convergence of the method. In Sect. 4, we present Inexact StoDCuP, an inexact variant of StoDCuP which builds inexact cuts on the basis of approximate primal-dual solutions of the subproblems solved along the iterations of the method. We also prove the almost sure convergence of Inexact StoDCuP for vanishing noises. Our convergence proofs of DCuP and StoDCuP are based on the convergence analysis of SDDP for nonlinear problems in [6] but additional technical results are needed due to the linearizations of cost and constraint functions, see Lemmas 2.1-(c), (d), 2.2, 2.4-(b), 3.4-(c), 3.7, 3.8 and Theorems 2.6-(i), (ii), 3.9. Finally, numerical experiments are presented in Sect. 5 on nondifferentiable multistage stochastic programs. Two variants of Inexact StoDCuP are presented: with and without cut selection strategies. In all instances tested, at least one inexact variant computes a good approximate policy quicker than StoDCuP while SDDP and the previous inexact variant of SDDP from [8] combined with Mosek library to solve subproblems were not able to solve a differentiable reformulation of the problem (recall that such reformulation is necessary to use the inexact variant of SDDP from [8] which applies to differentiable stochastic programs).

We will use the following notation:

- For a real-valued convex function $f$, we denote by $\ell_f(\cdot; x_0)$ an arbitrary lower bounding linearization of $f$ at $x_0$, i.e., $\ell_f(\cdot; x_0) = f(x_0) + s_f(x_0)^\top(\cdot - x_0)$ where $s_f(x_0)$ is an arbitrary subgradient of $f$ at $x_0$.
- The domain of a point to set operator $T : A \rightrightarrows B$ is given by $\text{Dom}(T) = \{a \in A : T(a) \neq \emptyset\}$.
- For vectors $x, y \in \mathbb{R}^n$, $\langle x, y\rangle = x^\top y$ is the usual scalar product between $x$ and $y$.
- For $a \in \mathbb{R}^n$, $\bar{B}(a; \varepsilon) = \{x \in \mathbb{R}^n : \|x - a\|_2 \leq \varepsilon\}$.
- The domain of a convex function $f : X \to (-\infty, \infty]$ is $\text{dom}(f) = \{x \in X : f(x) < \infty\}$.
- The relative interior ri $X$ of a set $X$ is the set $\{x \in X : \exists \varepsilon > 0 : \bar{B}(x; \varepsilon) \cap \text{Aff}(X) \subset X\}$.
- The subdifferential of the convex function $f : X \to (-\infty, \infty]$ at $x$ is

$$\partial f(x) = \{s : f(y) \geq f(x) + \langle s, y - x\rangle \; \forall y \in X\}.$$

- The indicator function $\delta_X(\cdot)$ of the set $X$ is given by $\delta_X(x) = 0$ if $x \in X$ and $\delta_X(x) = \infty$ otherwise.
- A function $f : \mathbb{R}^n \to (-\infty, \infty]$ is proper if there is $x$ such that $f(x)$ is finite.
- **e** is a vector of ones whose dimension depends on the context.

## 2 The DCuP (Dynamic Cutting Plane) Algorithm

### 2.1 Problem Formulation and Assumptions

Given $x_0 \in \mathbb{R}^n$, consider the optimization problem

$$\begin{array}{l} \inf\limits_{x_1,\ldots,x_T \in \mathbb{R}^n} \displaystyle\sum_{t=1}^{T} f_t(x_t, x_{t-1}) \\ g_t(x_t, x_{t-1}) \leq 0, \quad A_t x_t + B_t x_{t-1} = b_t, \;\; t = 1, \ldots, T, \\ x_t \in \mathcal{X}_t, \;\; t = 1, \ldots, T, \end{array} \tag{2.1}$$

where $A_t$ and $B_t$ are matrices of appropriate dimensions, $f_t : \mathbb{R}^n \times \mathbb{R}^n \to (-\infty, \infty]$ and $g_t : \mathbb{R}^n \times \mathbb{R}^n \to (-\infty, \infty]^p$. In this problem, for each step $t$, we have nonlinear and linear coupling constraints, $g_t(x_t, x_{t-1}) \leq 0$ and $A_t x_t + B_t x_{t-1} = b_t$, respectively, and set constraints $x_t \in \mathcal{X}_t$. Without loss of generality, nonlinear noncoupling constraints $h_t(x_t) \leq 0$ can be dealt with by incorporating them into the constraint $g_t(x_t, x_{t-1}) \leq 0$. For convenience, we use the short notation

$$X_t(x_{t-1}) := \{x_t \in \mathcal{X}_t : \; g_t(x_t, x_{t-1}) \leq 0, \quad A_t x_t + B_t x_{t-1} = b_t\} \tag{2.2}$$

and

$$X_t^0(x_{t-1}) = X_t(x_{t-1}) \cap \text{ri}\, \mathcal{X}_t. \tag{2.3}$$

With this notation, the dynamic programming equations corresponding to problem (2.1) are

$$\mathcal{Q}_t(x_{t-1}) = \begin{cases} \inf_{x_t \in \mathbb{R}^n} \ F_t(x_t, x_{t-1}) := f_t(x_t, x_{t-1}) + \mathcal{Q}_{t+1}(x_t) \\ x_t \in X_t(x_{t-1}), \end{cases} \tag{2.4}$$

for $t = 1, \ldots, T$, and $\mathcal{Q}_{T+1} \equiv 0$. The cost-to-go function $\mathcal{Q}_{t+1}(x_t)$ represents the optimal total cost for time steps $t+1, \ldots, T$, starting from state $x_t$ at the beginning of step $t+1$. Clearly, it follows from the above definition that

$$\mathrm{Dom}(X_t^0) \subset \mathrm{Dom}(X_t) \quad \forall t = 1, \ldots, T. \tag{2.5}$$

Setting $\mathcal{X}_0 = \{x_0\}$, the following assumptions are made throughout this section.

**Assumption (H1):**

(1) For $t = 1, \ldots, T$:

- (a) $\mathcal{X}_t \subset \mathbb{R}^n$ is nonempty, convex, and compact;
- (b) $f_t$ is a proper lower-semicontinuous convex function such that $\mathcal{X}_t \times \mathcal{X}_{t-1} \subset \mathrm{int}\,(\mathrm{dom}(f_t))$;
- (c) each of the $p$ components $g_{ti}, i = 1, \ldots, p$, of $g_t$ is a proper lower-semicontinuous convex function such that $\mathcal{X}_t \times \mathcal{X}_{t-1} \subset \mathrm{int}\,(\mathrm{dom}(g_{ti}))$.

(2) $X_1(x_0) \neq \emptyset$ and $\mathcal{X}_{t-1} \subset \mathrm{int}\left[\mathrm{Dom}(X_t^0)\right]$ for every $t = 2, \ldots, T$.

The following simple lemma states a few consequences of the above assumptions.

**Lemma 2.1** *The following statements hold:*

(a) *for every $t = 1, \ldots, T$, $\mathcal{Q}_{t+1}$ is a convex function such that*

$$\mathcal{X}_t \subset \mathrm{int}\ (\mathrm{dom}(\mathcal{Q}_{t+1}))\,;$$

(b) *for every $t = 1, \ldots, T$, $\mathcal{Q}_{t+1}$ is Lipschitz continuous on $\mathcal{X}_t$;*

(c) *for every $t = 1, \ldots, T$, $i = 1, \ldots, p$, and $(x_t, x_{t-1}) \in \mathcal{X}_t \times \mathcal{X}_{t-1}$,*

$$\partial f_t(x_t, x_{t-1}) \neq \emptyset, \quad \partial g_{ti}(x_t, x_{t-1}) \neq \emptyset;$$

(d) *for every $t = 1, \ldots, T$, $i = 1, \ldots, p$, the sets*

$$\bigcup \{\partial f_t(x_t, x_{t-1}) : (x_t, x_{t-1}) \in \mathcal{X}_t \times \mathcal{X}_{t-1}\}\,,$$
$$\bigcup \{\partial g_{ti}(x_t, x_{t-1}) : (x_t, x_{t-1}) \in \mathcal{X}_t \times \mathcal{X}_{t-1}\}$$

*are bounded.*

***Proof*** (a) The proof is by backward induction on $t$. The result clearly holds for $t = T$ since $\mathcal{Q}_{T+1} \equiv 0$. Assume now that $\mathcal{Q}_{t+1}$ is a convex function such that $\mathcal{X}_t \subset \mathrm{int}\,(\mathrm{dom}(\mathcal{Q}_{t+1}))$ for some $2 \leq t \leq T$. Then, condition 1) of Assumption (H1) implies that the function $(x_t, x_{t-1}) \mapsto F_t(x_t, x_{t-1}) + \delta_{X_t(x_{t-1})}(x_t)$ is convex. This conclusion together with the definition of $\mathcal{Q}_t$ and the discussion following Theorem 5.7 of [28] then imply that $\mathcal{Q}_t$ is a convex function. Moreover, conditions 1)b) and 2) of Assumption (H1) and relation (2.5) imply that there exists $\varepsilon > 0$ such that for every $x_{t-1} \in \mathcal{X}_{t-1} + \bar{B}(0, \varepsilon)$,

$$\mathrm{dom}(f_t(\cdot, x_{t-1})) \supset \mathcal{X}_t, \quad X_t(x_{t-1}) \neq \emptyset.$$

The induction hypothesis, the latter observation, and relations (2.2) and (2.4), then imply that

$$\begin{aligned} X_t(x_{t-1}) \cap \mathrm{dom}(F_t(\cdot, x_{t-1})) &= X_t(x_{t-1}) \cap \mathrm{dom}(f_t(\cdot, x_{t-1})) \cap \mathrm{dom}(\mathcal{Q}_{t+1}) \\ &\supset X_t(x_{t-1}) \cap \mathcal{X}_t = X_t(x_{t-1}) \neq \emptyset \end{aligned}$$

for every $x_{t-1} \in \mathcal{X}_{t-1} + \bar{B}(0, \varepsilon)$. Since by (2.4),

$$\mathrm{dom}(\mathcal{Q}_t) = \{x_{t-1} \in \mathbb{R}^n : X_t(x_{t-1}) \cap \mathrm{dom}(F_t(\cdot, x_{t-1})) \neq \emptyset\},$$

we then conclude that $\mathcal{X}_{t-1} + \bar{B}(0, \varepsilon) \subset \mathrm{dom}(\mathcal{Q}_t)$, and hence that $\mathcal{X}_{t-1} \subset \mathrm{int}\,(\mathrm{dom}(\mathcal{Q}_t))$. We have thus proved that (a) holds.

(b) This statement follows from statement a) and Theorem 10.4 of [28].

(c–d) These two statements follow from conditions 1)a), 1)b) and 1)c) of Assumption (H1) together with Theorem 23.4 and 24.7 of [28]. □

### 2.2 Forward DCuP

Before formally describing the DCuP algorithm, we give some motivation for it. At iteration $k \geq 1$ and stage $t = 1, \ldots, T$, the algorithm uses the following approximation to the function $\mathcal{Q}_t(\cdot)$ defined in (2.4):

$$\underline{\mathcal{Q}}_t^{k-1}(x_{t-1}) = \min\left\{f_t^{k-1}(x_t, x_{t-1}) + \mathcal{Q}_{t+1}^{k-1}(x_t) : x_t \in X_t^{k-1}(x_{t-1})\right\} \tag{2.6}$$

where

$$X_t^{k-1}(x_{t-1}) = \{x_t \in \mathcal{X}_t : \; g_t^{k-1}(x_t, x_{t-1}) \leq 0, \; A_t x_t + B_t x_{t-1} = b_t\} \tag{2.7}$$

and $f_t^{k-1}, g_t^{k-1}$, and $\mathcal{Q}_{t+1}^{k-1}$ are polyhedral functions minorizing $f_t, g_t$ and $\mathcal{Q}_{t+1}$, respectively, i.e.,

$$f_t^{k-1} \leq f_t, \quad g_t^{k-1} \leq g_t, \quad \mathcal{Q}_{t+1}^{k-1} \leq \mathcal{Q}_{t+1}. \tag{2.8}$$

For $t = T + 1$, we actually assume that $\mathcal{Q}_{T+1}^{k-1} \equiv 0$, and hence that $\mathcal{Q}_{T+1}^{k} = \mathcal{Q}_{T+1}$. Moreover, we also assume that $\underline{\mathcal{Q}}_{T+1}^{k-1} \equiv 0$, and hence $\underline{\mathcal{Q}}_{T+1}^{k-1} = \mathcal{Q}_{T+1}$.

Observe that for every $k \geq 0$, $t = 1, \ldots, T$, and $x_{t-1} \in \mathcal{X}_{t-1}$, relations (2.7) and (2.8) imply that

$$X_t(x_{t-1}) \subset X_t^k(x_{t-1}) \subset \mathcal{X}_t \tag{2.9}$$

and

$$f_t^k(\cdot, x_{t-1}) + \underline{\mathcal{Q}}_{t+1}^k(\cdot) \leq f_t(\cdot, x_{t-1}) + \mathcal{Q}_{t+1}(\cdot),$$

and hence that

$$\underline{\mathcal{Q}}_t^k \leq \mathcal{Q}_t, \quad \forall\, t = 1, 2, \ldots, T, \;\; \forall\, k \geq 0. \tag{2.10}$$

At iteration $k$, feasible points $x_1^k, \ldots, x_T^k$ are computed recursively as follows: for $t = 1, \ldots, T$, $x_t^k$ is set to be an optimal solution of subproblem (2.6) with $x_{t-1} = x_{t-1}^k$ with the convention that $x_0^k = x_0$. These points in turn are used to compute new affine functions minorizing $f_t$, $g_t$ and $\mathcal{Q}_t$ which are then added to the bundle of affine functions describing $f_t^{k-1}$, $g_t^{k-1}$, and $\underline{\mathcal{Q}}_t^{k-1}$ to obtain new lower bounding approximations $f_t^k$, $g_t^k$, and $\underline{\mathcal{Q}}_t^k$ for $f_t$, $g_t$ and $\mathcal{Q}_t$, respectively.

The precise description of DCuP algorithm is as follows.

---

**DCuP (Dynamic Cutting Plane) with linearizations computed in a forward pass.**

---

**Step 0. Initialization.** For every $t = 1, \ldots, T$, let affine functions $f_t^0$ and $g_t^0$ such that $f_t^0 \leq f_t$ and $g_t^0 \leq g_t$, and a piecewise linear function $\underline{\mathcal{Q}}_t^0 : \mathcal{X}_{t-1} \to \mathbb{R}$ such that $\underline{\mathcal{Q}}_t^0 \leq \mathcal{Q}_t$ be given. We write $\underline{\mathcal{Q}}_t^0$ as $\underline{\mathcal{Q}}_t^0(x_{t-1}) = \theta_t^0 + \langle \beta_t^0, x_{t-1} \rangle$, set $\underline{\mathcal{Q}}_{T+1}^0 \equiv 0$, and $k = 1$.

**Step 1. Forward pass.** Set $\mathcal{C}_{T+1}^k = \underline{\mathcal{Q}}_{T+1}^k \equiv 0$ and $x_0^k = x_0$. For $t = 1, 2, \ldots, T$, do:

a) find an optimal solution $x_t^k$ of

$$\underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k) = \begin{cases} \inf\limits_{x_t \in \mathbb{R}^n} f_t^{k-1}(x_t, x_{t-1}^k) + \underline{\mathcal{Q}}_{t+1}^{k-1}(x_t) \\ x_t \in X_t^{k-1}(x_{t-1}^k), \end{cases} \tag{2.11}$$

where $X_t^k(\cdot)$ is as in (2.7);

b) compute function values and subgradients of $f_t$ and $g_{ti}, i = 1, \ldots, p$, at $(x_t^k, x_{t-1}^k)$, and let $\ell_{f_t}(\cdot; (x_t^k, x_{t-1}^k))$ and $\ell_{g_{ti}}(\cdot; (x_t^k, x_{t-1}^k))$ denote the corresponding linearizations;

c) set

$$f_t^k = \max\left(f_t^{k-1}, \ell_{f_t}\left((\cdot, \cdot); (x_t^k, x_{t-1}^k)\right)\right), \tag{2.12}$$

$$g_{ti}^k = \max\left(g_{ti}^{k-1}, \ell_{g_{ti}}\left((\cdot, \cdot); (x_t^k, x_{t-1}^k)\right)\right), \quad \forall i = 1, \ldots, p, \tag{2.13}$$

and define $g_t^k := (g_{t1}^k, \ldots, g_{tp}^k)$;

d) if $t \geq 2$, then compute $\beta_t^k \in \partial \underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k)$ and denote the corresponding linearization of $\underline{\mathcal{Q}}_t^{k-1}$ as

$$\mathcal{C}_t^k(\cdot) := \underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k) + \langle \beta_t^k, \cdot - x_{t-1}^k \rangle;$$

moreover, set

$$\underline{\mathcal{Q}}_t^k = \max\{\underline{\mathcal{Q}}_t^{k-1}, \mathcal{C}_t^k\}; \tag{2.14}$$

**Step 2.** Set $k \leftarrow k+1$ and go to Step 1.

---

We now make a few remarks about DCuP. First, Lemma 2.1(c) guarantees the existence of the subgradients, and hence the linearizations, of the functions $f_t$ and $g_{ti}$, $i = 1, \ldots, p$, at any point $(x_t, x_{t-1}) \in \mathcal{X}_t \times \mathcal{X}_{t-1}$, and hence that the functions $f_t^k$ and $g_t^k$ in Step 1 are well-defined. Second, in view of the definition of $x_t^k$ in Step a), we have that $x_t^k \in X_t^{k-1}(x_{t-1}^k) \subset \mathcal{X}_t$ for every $t = 1, \ldots, T$ and $k \geq 0$. Third, Lemma 2.2(b) and the previous remark guarantee the existence of the subgradient $\beta_t^k$ in Step d). Fourth, we dicuss in Sect. 2.3 ways of computing this subgradient.

In the remaining part of this subsection, we provide the convergence analysis of DCuP. The following result states some basic properties about the functions involved in DCuP.

**Lemma 2.2** *The following statements hold:*

(a) *for every $k \geq 1$ and $t = 1, \ldots, T$, we have*

$$f_t^k \leq f_t^{k+1} \leq f_t, \quad g_t^k \leq g_t^{k+1} \leq g_t, \tag{2.15}$$

$$X_t(x_{t-1}) \subset X_t^{k+1}(x_{t-1}) \subset X_t^k(x_{t-1}) \subset \mathcal{X}_{t-1} \quad \forall x_{t-1} \in \mathbb{R}^n, \tag{2.16}$$

$$\mathcal{Q}_{t+1}^k \leq \mathcal{Q}_{t+1}^{k+1} \leq \mathcal{Q}_{t+1}, \tag{2.17}$$

$$\underline{\mathcal{Q}}_t^k \leq \underline{\mathcal{Q}}_t^{k+1} \leq \mathcal{Q}_t. \tag{2.18}$$

(b) *For every $k \geq 1$ and $t = 2, \ldots, T$, function $\underline{\mathcal{Q}}_t^k$ is convex and* int $(\mathrm{dom}(\underline{\mathcal{Q}}_t^k)) \supset \mathcal{X}_{t-1}$*; as a consequence, $\partial \underline{\mathcal{Q}}_t^k(x_{t-1}) \neq \emptyset$ for every $x_{t-1} \in \mathcal{X}_{t-1}$.*

***Proof*** (a) Relations (2.15) and (2.16) follow immediately from the initialization of DCuP described in step 0, the recursive definitions of $f_t^k$ and $g_t^k$ in (2.12) and (2.13), respectively, the definition of $X_t^k(\cdot)$ in (2.7), and the fact that

$$\ell_{f_t}((\cdot,\cdot); (x_t^k, x_{t-1}^k)) \leq f_t(\cdot,\cdot), \quad \ell_{g_{ti}}((\cdot\cdot); (x_t^k, x_{t-1}^k)) \leq g_{ti}(\cdot\cdot).$$

Next note that the inequalities in (2.18) follow immediately from the respective ones in (2.15), (2.16) and (2.17), together with relations (2.4) and (2.11). It then remains to show that the inequalities in (2.17) hold. Indeed, the inequality $\mathcal{Q}_{t+1}^k \leq \mathcal{Q}_{t+1}^{k+1}$ follows immediately from (2.14) with $t = t + 1$. We will now show that inequalities $\mathcal{Q}_t^k \le \mathcal{Q}_t$ for every $t = 2, \dots, T + 1$ implies that $\mathcal{Q}_t^{k+1} \le \mathcal{Q}_t$ for every $t = 2, \dots, T + 1$, and hence that the second inequality in (2.17) follows from a simple inductive argument on $k$. Indeed, first observe that the inequality $\mathcal{Q}_{t+1}^k \le \mathcal{Q}_{t+1}$ implies that $\underline{\mathcal{Q}}_t^k \le \mathcal{Q}_t$. Next observe that the construction of $\mathcal{C}_t^{k+1}$ in Step d) of DCuP implies that $\mathcal{C}_t^{k+1} \le \underline{\mathcal{Q}}_t^k$, and hence that $\mathcal{C}_t^{k+1} \le \mathcal{Q}_t$. It then follows from (2.14) and the inequality $\mathcal{Q}_t^k \le \mathcal{Q}_t$ that $\mathcal{Q}_t^{k+1} \le \mathcal{Q}_t$. We have thus shown that $\mathcal{Q}_t^k \le \mathcal{Q}_t$ for every $t = 2, \dots, T + 1$ implies that $\mathcal{Q}_t^{k+1} \le \mathcal{Q}_t$ for every $t = 2, \dots, T$. Since the latter inequality for $t = T + 1$ is straightforward and $\mathcal{Q}_t^0 \le \mathcal{Q}_t$ for $t = 2, \dots, T$, (2.17) follows.

(b) The assertion that $\underline{\mathcal{Q}}_t^k$ is a convex function follows from the fact that $\mathcal{Q}_{t+1}^k$ is convex and the same arguments used in Lemma 2.1 to show that $\mathcal{Q}_t$ is convex. The assertion that $\mathrm{dom}(\underline{\mathcal{Q}}_t^k) \supset \mathcal{X}_{t-1}$ follows from the fact that by (2.18) we have $\underline{\mathcal{Q}}_t^k \le \mathcal{Q}_t$, and hence that

$$\mathrm{int}\left(\mathrm{dom}(\underline{\mathcal{Q}}_t^k)\right) \supset \mathrm{int}\ (\mathrm{dom}(\mathcal{Q}_t)) \supset \mathcal{X}_{t-1},$$

where the last inclusion is due to Lemma 2.1(a). □

The following technical result is useful to establish uniform Lipschitz continuity of convex functions.

**Lemma 2.3** *Assume that $\phi^-$ and $\phi^+$ are proper convex functions such that $\phi^- \le \phi^+$. Then, for any nonempty compact set $K \subset \mathrm{int}\,(\mathrm{dom}(\phi^+))$, there exists a scalar $L = L(K) \ge 0$ satisfying the following property: any convex function $\phi$ such that $\phi^- \le \phi \le \phi^+$ is L-Lipschitz continuous on $K$.*

***Proof*** Let $\phi$ be a convex function such that $\phi^- \le \phi \le \phi^+$ and let $K \subset \mathrm{int}\,(\mathrm{dom}(\phi^+))$ be a nonempty compact set. Since $\phi^-$ and $\phi^+$ are proper, it then follows that $\phi$ is proper and $\mathrm{dom}(\phi) \supset \mathrm{dom}(\phi^+)$, and hence that $\mathrm{int}\,(\mathrm{dom}(\phi^-)) \supset \mathrm{int}\,(\mathrm{dom}(\phi)) \supset \mathrm{int}\,(\mathrm{dom}(\phi^+)) \supset K$. Hence, in view of Theorem 23.4 of [28], we conclude that $\partial\phi(x) \neq \emptyset$ for every $x \in K$. We now claim that there exists $L$ such that $\|\beta\| \le L$ for every $\beta \in \partial\phi(x)$ and $x \in K$. This claim in turn can be easily seen to imply that the conclusion of the lemma holds. To show the claim, let $x \in K$ and $0 \neq \beta \in \partial\phi(x)$ be given. The inclusion $K \subset \mathrm{int}\,(\mathrm{dom}(\phi^+))$ implies the existence of $\varepsilon > 0$ such that $K_\varepsilon := K + \bar{B}(0; \varepsilon) \subset \mathrm{int}\,(\mathrm{dom}(\phi^+))$. Let

$$y_\varepsilon := x + \varepsilon \frac{\beta}{\|\beta\|}, \quad \theta^+ := \max_{y \in K_\varepsilon} \phi^+(y), \quad \theta^- := \min_{y \in K} \phi^-(y).$$

Clearly, $y_\varepsilon \in K_\varepsilon$ due to the definition of $K_\varepsilon$ and the facts that $x \in K$ and $\|y_\varepsilon - x\| \le \varepsilon$. Moreover, using the fact that every proper convex function is continuous in the interior of its domain, we then conclude that the proper convex functions $\phi^+$ and $\phi^-$ are continuous on $K_\varepsilon$ and $K$, respectively, since these two sets lie in the interior of their domains, respectively. Hence, it follows from Weierstrass' theorem that $\theta^+$ and $\theta^-$ are both finite due to the compactness of $K$ and $K_\varepsilon$, respectively. Using the facts that

$x \in K$, $y_\varepsilon \in K_\varepsilon$, $\beta \in \partial\phi(x)$ and $\phi^+ \geq \phi$, the definitions of $\theta^+$ and $\theta^-$, and the definition of subgradient, it then follows that

$$\theta^+ \geq \phi^+(y_\varepsilon) \geq \phi(y_\varepsilon) \geq \phi(x) + \langle \beta, y_\varepsilon - x\rangle = \phi(x) + \varepsilon\|\beta\| \geq \theta^- + \varepsilon\|\beta\|$$

and hence that the claim holds with $L = (\theta^+ - \theta^-)/\varepsilon$. □

**Lemma 2.4** *The following statements hold:*

(a) *For each $t = 2, \ldots, T$, there exist $L_t \geq 0$ such that the functions $\mathcal{Q}_t^k$ and $\underline{\mathcal{Q}}_t^k$ are $L_t$-Lipschitz continuous on $\mathcal{X}_{t-1}$ for every $k \geq 1$;*
(b) *For each $t = 1, \ldots, T$, there exist $\hat{L}_t \geq 0$ such that the functions $f_t^k$ and $g_{ti}^k$ are $\hat{L}_t$-Lipschitz continuous functions on $\mathcal{X}_t \times \mathcal{X}_{t-1}$ for every $k \geq 1$ and $i = 1, \ldots, p$.*

***Proof*** Let $t \in \{2, \ldots, T\}$ be given. The existence of $L_t$ satisfying (a) follows from Lemmas 2.1 and 2.2, and applying Lemma 2.3 twice, the first time with $K = \mathcal{X}_{t-1}$, $\phi^+ = \mathcal{Q}_t$ and $\phi^- = \mathcal{Q}_t^0$, and the second time with $K = \mathcal{X}_{t-1}$, $\phi^+ = \mathcal{Q}_t$ and $\phi^- = \underline{\mathcal{Q}}_t^0$. Moreover, the existence of $\hat{L}_t$ satisfying (b) follows from Lemma 2.2, and applying Lemma 2.3 twice, the first time with $K = \mathcal{X}_t \times \mathcal{X}_{t-1}$, $\phi^+ = f_t$ and $\phi^- = f_t^0$, and the second time with $K = \mathcal{X}_t \times \mathcal{X}_{t-1}$, $\phi^+ = g_{ti}$ and $\phi^- = g_{ti}^0$ for $i = 1, \ldots, p$. □

We now state a result whose proof is given in Lemma 5.2 of [5]. Even though the latter result assumes convexity of the functions involved in its statement, its proof does not make use of this assumption. For this reason, we state the result here in a slightly more general way than it is stated in Lemma 5.2 of [5].

**Lemma 2.5** *Lemma 5.2 in* [5]*. Assume that $Y \subset \mathbb{R}^n$ is a compact set, $f : \mathbb{R}^n \to (-\infty, \infty]$ is a function and $\{f_k : \mathbb{R}^n \to (\infty, \infty]\}_{k=1}^\infty$ is a sequence of functions such that, for some integer $k_0 > 0$ and scalar $L > 0$, we have:*

(a) *$f^{k-k_0}(y) \leq f^k(y) \leq f(y) < \infty$ for every $k \geq k_0 + 1$ and $y \in Y$;*
(b) *$f^k$ is $L$-Lipschitz continuous on $Y$ for every $k \geq 1$.*

*Then, for any infinite sequence $\{y^k\}_{k=1}^\infty \subset Y$, we have*

$$\lim_{k\to+\infty} [f(y^k) - f^k(y^k)] = 0 \iff \lim_{k\to+\infty} [f(y^k) - f^{k-k_0}(y^k)] = 0.$$

We are now ready to provide the main result of this subsection, i.e., the convergence analysis of DCuP.

**Theorem 2.6** *Let Assumption (H1) hold. Define*

$$\mathcal{H}(t) \begin{cases} \text{(i)} & \overline{\lim_{k\to+\infty}}\, g_{ti}(x_t^k, x_{t-1}^k) \leq 0, \quad i = 1, \ldots, p, \\ \text{(ii)} & \lim_{k\to+\infty} \mathcal{Q}_t(x_{t-1}^k) - \underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k) = \lim_{k\to+\infty} \mathcal{Q}_t(x_{t-1}^k) - \underline{\mathcal{Q}}_t^k(x_{t-1}^k) = 0, \\ \text{(iii)} & \lim_{k\to+\infty} \mathcal{Q}_t(x_{t-1}^k) - \sum_{\tau=t}^T f_\tau(x_\tau^k, x_{\tau-1}^k) = 0, \\ \text{(iv)} & \lim_{k\to+\infty} \mathcal{Q}_t(x_{t-1}^k) - \mathcal{Q}_t^k(x_{t-1}^k) = 0. \end{cases}$$

*Then $\mathcal{H}(t)$-(i) holds for $t = 1, \dots, T$, $\mathcal{H}(t)$-(ii), (iii) hold for $t = 1, \dots, T+1$, and $\mathcal{H}(t)$-(iv) holds for $t = 2, \dots, T+1$. In particular, the limit of the sequence of upper bounds $(\sum_{t=1}^{T} f_t(x_t^k, x_{t-1}^k))_{k \geq 1}$ and of lower bounds $(\underline{\mathcal{Q}}_1^{k-1}(x_0))_{k \geq 1}$ is the optimal value $\mathcal{Q}_1(x_0)$ of (2.1) and any accumulation point of the sequence $(x_1^k, \dots, x_T^k)$ is an optimal solution to (2.1).*

***Proof*** We first prove $\mathcal{H}(t)$-(i) for $t = 1, \dots, T$. Let $t \in \{1, \dots, T\}$ be given and define the sequence $\{y_t^k\}$ as $y_t^k = (x_t^k, x_{t-1}^k)$ for every $k \geq 1$. In view of Lemma 2.2, we have $g_{ti}(y_t^k) \geq g_{ti}^k(y_t^k) \geq \ell_{g_{ti}}(y_t^k; y_t^k) = g_{ti}(y_t^k)$, and hence

$$g_t^k(y_t^k) = g_t(y_t^k), \ \forall\, k \geq 1. \tag{2.19}$$

Due to Lemma 2.4-(b), functions $g_{ti}^k$ are convex $\hat{L}_t$-Lipschitz continuous on $\mathcal{X}_t \times \mathcal{X}_{t-1}$. Therefore, recalling (2.19), we can apply Lemma 2.5 to $f = g_{ti}$, $f^k = g_{ti}^k$, $y^k = y_t^k$, $Y = \mathcal{X}_t \times \mathcal{X}_{t-1}$ for $i = 1, \dots, p$, to obtain

$$\lim_{k \to +\infty} g_t(x_t^k, x_{t-1}^k) - g_t^{k-1}(x_t^k, x_{t-1}^k) = 0. \tag{2.20}$$

The latter conclusion together with the fact that $x_t^k \in X_t^{k-1}(x_{t-1}^k)$, and hence $g_t^{k-1}(x_t^k, x_{t-1}^k) \leq 0$, for every $k \geq 1$, then implies that $\mathcal{H}(t)$-(i) holds.

Let us now show $\mathcal{H}(1)$-(ii), (iii) and $\mathcal{H}(t)$-(ii)-(iii), (iv) for $t = 2, \dots, T+1$ by backward induction on $t$. $\mathcal{H}(T+1)$-(ii), (iii), (iv) trivially holds. Now, fix $t \in \{1, \dots, T\}$ and assume that $\mathcal{H}(t+1)$-(ii), (iii), (iv) holds. We will show that $\mathcal{H}(t)$-(ii), (iii) holds and that $\mathcal{H}(t)$-(iv) holds if $t \geq 2$. Indeed, since $f_t \geq f_t^k \geq \ell_{f_t}(\cdot; y_t^k)$ and $f_t(y_t^k) = \ell_{f_t}(y_t^k; y_t^k)$, we conclude that $f_t^k(y_t^k) = f_t(y_t^k)$ for every $k \geq 1$, and hence that $\lim_{k \to +\infty} f_t(y_t^k) - f_t^k(y_t^k) = 0$. Recalling by Lemma 2.4-(b) that $f_t^k$ is $\hat{L}_t$-Lipschitz continuous on $\mathcal{X}_t \times \mathcal{X}_{t-1}$ and using Lemma 2.5 with $f = f_t$, $f^k = f_t^k$, $(y^k) = (y_t^k)$, and $Y = \mathcal{X}_t \times \mathcal{X}_{t-1}$, we conclude that

$$\lim_{k \to +\infty} f_t(x_t^k, x_{t-1}^k) - f_t^{k-1}(x_t^k, x_{t-1}^k) = 0. \tag{2.21}$$

Moreover, by the induction hypothesis $\mathcal{H}(t+1)$-(iv), we have $\lim_{k \to +\infty} \mathcal{Q}_{t+1}^k(x_t^k) - \mathcal{Q}_{t+1}(x_t^k) = 0$. Recalling by Lemma 2.4-(a) that functions $\mathcal{Q}_t^k$ are $L_t$-Lipschitz continuous on $\mathcal{X}_{t-1}$, we can use Lemma 2.5 with $k_0 = 1$, $f = \mathcal{Q}_{t+1}$, $f^k = \mathcal{Q}_{t+1}^k$, $y^k = x_t^k$ and $Y = \mathcal{X}_t$, to obtain

$$\lim_{k \to +\infty} \mathcal{Q}_{t+1}^{k-1}(x_t^k) - \mathcal{Q}_{t+1}(x_t^k) = 0. \tag{2.22}$$

Now, using Lemma 2.2, we easily see that the objective function $f_t^{k-1}(\cdot, x_{t-1}^k) + \mathcal{Q}_{t+1}^{k-1}(\cdot)$ and feasible region $X_t^{k-1}(x_{t-1}^k)$ of (2.11) satisfies $f_t^{k-1}(\cdot, x_{t-1}^k) + \mathcal{Q}_{t+1}^{k-1}(\cdot) \leq F_t(\cdot, x_{t-1}^k)$ and $X_t^{k-1}(x_{t-1}^k) \supseteq X_t(x_{t-1}^k)$. Since $x_t^k$ is an optimal solution of (2.11) and $\mathcal{Q}_t(x_{t-1}^k)$ is the optimal value of $\min\{F_t(x_t, x_{t-1}^k) : x_t \in X_t(x_{t-1}^k)\}$ due to (2.4), we then conclude that $f_t^{k-1}(x_t^k, x_{t-1}^k) + \mathcal{Q}_{t+1}^{k-1}(x_t^k) \leq \mathcal{Q}_t(x_{t-1}^k)$. Hence, we conclude that

$$\begin{aligned} 0 &\geq \overline{\lim}_{k\to+\infty} f_t^{k-1}(x_t^k, x_{t-1}^k) + \mathcal{Q}_{t+1}^{k-1}(x_t^k) - \mathcal{Q}_t(x_{t-1}^k) \\ &= \overline{\lim}_{k\to+\infty} f_t(x_t^k, x_{t-1}^k) + \mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_t(x_{t-1}^k) \end{aligned}$$

where the equality is due to (2.21) and (2.22). We now claim that

$$\lim_{k\to+\infty} f_t(x_t^k, x_{t-1}^k) + \mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_t(x_{t-1}^k) = 0. \tag{2.23}$$

Indeed, assume by contradiction that the above claim does not hold. Then, it follows from the last conclusion before the claim that

$$\underline{\lim}_{k\to+\infty} f_t(x_t^k, x_{t-1}^k) + \mathcal{Q}_{t+1}(x_t^k) - \mathcal{Q}_t(x_{t-1}^k) < 0. \tag{2.24}$$

Since $\{(x_t^k, x_{t-1}^k)\}$ is a sequence lying in the compact set $\mathcal{X}_t \times \mathcal{X}_{t-1}$, it has a subsequence $\{(x_t^k, x_{t-1}^k)\}_{k\in K}$ converging to some $(x_t^*, x_{t-1}^*) \in \mathcal{X}_t \times \mathcal{X}_{t-1}$. Hence, in view of $\mathcal{H}(t)$-(i), (2.24), and the fact that $f_t$ and $g_t$ are lower semi-continuous on $\mathcal{X}_t \times \mathcal{X}_{t-1}$ and $\mathcal{Q}_t$ (resp. $\mathcal{Q}_{t+1}$) is lower semi-continuous on $\mathcal{X}_{t-1}$ (resp. $\mathcal{X}_t$), we conclude that

$$g_t(x_t^*, x_{t-1}^*) \leq 0, \quad f_t(x_t^*, x_{t-1}^*) + \mathcal{Q}_{t+1}(x_t^*) - \mathcal{Q}_t(x_{t-1}^*) < 0$$

and hence that $x_t^* \in X_t(x_{t-1}^*)$ (recall that $\mathcal{X}_t$ is closed) and $F_t(x_t^*, x_{t-1}^*) < \mathcal{Q}_t(x_{t-1}^*)$ due to the definition of $X_t$ and $F_t$ in (2.2) and (2.4), respectively. Since this contradicts the definition of $\mathcal{Q}_t$ in (2.4), the above claim follows. Combining

$$\begin{aligned} 0 \leq \mathcal{Q}_t(x_{t-1}^k) - \underline{\mathcal{Q}}_t^k(x_{t-1}^k) &\leq \mathcal{Q}_t(x_{t-1}^k) - \underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k), \\ &= \mathcal{Q}_t(x_{t-1}^k) - f_t^{k-1}(x_t^k, x_{t-1}^k) \\ &\quad - \mathcal{Q}_{t+1}^{k-1}(x_t^k) \text{ [by definition of } x_t^k] \end{aligned}$$

with relations (2.21), (2.22), (2.23) we obtain $\lim_{k\to+\infty} \mathcal{Q}_t(x_{t-1}^k) - \underline{\mathcal{Q}}_t^k(x_{t-1}^k) = 0$. Also observe that

$$\begin{aligned} &\lim_{k\to+\infty} \mathcal{Q}_t(x_{t-1}^k) - \sum_{\tau=t}^{T} f_\tau(x_\tau^k, x_{\tau-1}^k) \\ &\quad = \underbrace{\lim_{k\to+\infty} \mathcal{Q}_t(x_{t-1}^k) - f_t(x_t^k, x_{t-1}^k) - \mathcal{Q}_{t+1}(x_t^k)}_{=0 \text{ by } (2.23)} \\ &\qquad + \underbrace{\lim_{k\to+\infty} \mathcal{Q}_{t+1}(x_t^k) - \sum_{\tau=t+1}^{T} f_\tau(x_\tau^k, x_{\tau-1}^k),}_{=0 \text{ using } \mathcal{H}(t+1)-(iii)} \\ &\quad = 0, \end{aligned}$$

and we have shown $\mathcal{H}(t)$-(ii), (iii).

Finally, if $t \geq 2$, $\mathcal{H}(t)$-(iv) follows from

$$\begin{aligned} 0 \leq \mathcal{Q}_t(x_{t-1}^k) - \underline{\mathcal{Q}}_t^k(x_{t-1}^k) &\leq \mathcal{Q}_t(x_{t-1}^k) - \mathcal{C}_t^k(x_{t-1}^k) \text{ since } \underline{\mathcal{Q}}_t^k \geq \mathcal{C}_t^k, \\ &= \mathcal{Q}_t(x_{t-1}^k) - f_t^{k-1}(x_t^k, x_{t-1}^k) \\ &\quad - \underline{\mathcal{Q}}_{t+1}^{k-1}(x_t^k) \text{ [by definition of } x_t^k] \end{aligned}$$

combined with relations (2.21), (2.22), (2.23). □

### 2.3 Computation of the Subgradient in Step d) of DCuP

This subsection explains how to compute a subgradient $\beta_t^k$ of $\underline{\mathcal{Q}}_t^{k-1}(\cdot)$ at $x_{t-1}^k$ in Step d) of DCuP.

Observe that we can express $\underline{\mathcal{Q}}_t^{k-1}$ as

$$\underline{\mathcal{Q}}_t^{k-1}(x_{t-1}) = \begin{cases} \min\limits_{x_t \in \mathbb{R}^n, f, \theta \in \mathbb{R}} f + \theta \\ x_t \in \mathcal{X}_t, \\ f \geq \ell_{f_t}(x_t, x_{t-1}, (x_t^j, x_{t-1}^j)), \; j = 1, \ldots, k-1, \\ \theta \geq \underline{\mathcal{Q}}_{t+1}^{i-1}(x_t^i) + \langle \beta_{t+1}^i, x_t - x_t^i \rangle, \; i = 1, \ldots, k-1, \\ \ell_{g_{ti}}(x_t, x_{t-1}, (x_t^j, x_{t-1}^j)) \leq 0, \; j = 1, \ldots, k-1, i = 1, \ldots, p, \\ A_t x_t + B_t x_{t-1} = b_t. \end{cases} \tag{2.25}$$

Due to Assumption (H1)-2), for every $x_{t-1} \in \mathcal{X}_{t-1}$, there exists $x_t \in \text{ri}(\mathcal{X}_t)$ such that $A_t x_t + B_t x_{t-1} = b_t$ and $g_t(x_t, x_{t-1}) \leq 0$, which implies that for every $i = 1, \ldots, p$, and $j = 1, \ldots, k-1$, we have

$$\ell_{g_{ti}}(x_t, x_{t-1}, (x_t^j, x_{t-1}^j)) \leq g_{ti}(x_t, x_{t-1}) \leq 0$$

and therefore Slater constraint qualification holds for problem (2.25) for every $x_{t-1} \in \mathcal{X}_{t-1}$. Next observe that due to the compactness of $\mathcal{X}_t$, the objective function of (2.25) bounded from below on the feasible set. It follows that the optimal value of (2.25) is finite and by the duality theorem, we can write problem (2.25) as the optimal value of the corresponding dual problem. To write this dual, it is convenient to rewrite $\underline{\mathcal{Q}}_t^{k-1}$ on $\mathcal{X}_{t-1}$ as

$$\underline{\mathcal{Q}}_t^{k-1}(x_{t-1}) = \begin{cases} \min\limits_{x_t \in \mathbb{R}^n, f, \theta \in \mathbb{R}} f + \theta \\ x_t \in \mathcal{X}_t, \\ f\mathbf{e} \geq A_t^{k-1} x_t + B_t^{k-1} x_{t-1} + C_t^{k-1}, \\ \theta\mathbf{e} \geq \theta_{t+1}^{0:k-1} + \beta_{t+1}^{0:k-1} x_t, \\ D_t^{k-1} x_t + E_t^{k-1} x_{t-1} + H_t^{k-1} \leq 0, \\ A_t x_t + B_t x_{t-1} = b_t, \end{cases} \tag{2.26}$$

where $\mathbf{e}$ is a vector of ones of dimension $k-1$ and $A_t^{k-1}$, $B_t^{k-1}$, $D_t^{k-1}$, $E_t^{k-1}$, $\beta_{t+1}^{1:k-1}$ (resp. $C_t^{k-1}$, $H_t^{k-1}$, $\theta_{t+1}^{1:k-1}$) are matrices (resp. vectors) of appropriate dimensions. In particular, $\beta_{t+1}^{0:k-1}$ is a matrix with $k$ rows with $(i+1)$th row equal to $(\beta_{t+1}^{i})^\top$ and $\theta_{t+1}^{0:k-1}$ is a vector of size $k$ with first component equal to $\theta_{t+1}^{0}$ and for $i \geq 2$ component $i$ given by $\theta_{t+1}^{i-1} = \underline{\mathcal{Q}}_{t+1}^{i-2}(x_t^{i-1}) - \langle \beta_{t+1}^{i-1}, x_t^{i-1}\rangle$.

We now write the dual of (2.26) as

$$\underline{\mathcal{Q}}_t^{k-1}(x_{t-1}) = \begin{cases} \max\limits_{\alpha,\mu,\delta,\lambda} h_{t,x_{t-1}}(\alpha,\lambda,\mu,\delta) \\ \alpha \geq 0, \mu \geq 0, \delta \geq 0, \lambda, \end{cases} \tag{2.27}$$

where dual function $h_{t,x_{t-1}}$ is given by

$$h_{t,x_{t-1}}(\alpha,\lambda,\mu,\delta) = \begin{cases} \min\limits_{x_t\in\mathbb{R}^n, f,\theta\in\mathbb{R}} L_{t,x_{t-1}}(x_t, f, \theta; \alpha,\lambda,\mu,\delta) \\ x_t \in \mathcal{X}_t, \end{cases} \tag{2.28}$$

with Lagrangian $L_{t,x_{t-1}}(x_t, f, \theta; \alpha,\lambda,\mu,\delta)$ given by

$$\begin{aligned} L_{t,x_{t-1}}(x_t, f, \theta; \alpha,\lambda,\mu,\delta) = f + \theta &+ \langle \alpha, A_t^{k-1}x_t + B_t^{k-1}x_{t-1} + C_t^{k-1} - f\mathbf{e}\rangle \\ &+ \langle \lambda, A_t x_t + B_t x_{t-1} - b_t\rangle \\ &+ \langle \mu, D_t^{k-1}x_t + E_t^{k-1}x_{t-1} + H_t^{k-1}\rangle \\ &+ \langle \delta, \theta_{t+1}^{1:k-1} + \beta_{t+1}^{1:k-1}x_t - \theta\mathbf{e}\rangle. \end{aligned}$$

With this notation, we have the following characterization of $\partial\underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k)$:

**Lemma 2.7** *Let Assumption (H1) hold. Then the subdifferential of $\underline{\mathcal{Q}}_t^{k-1}$ at $x_{t-1}^k$ is the set of points of form*

$$B_t^\top\lambda + (B_t^{k-1})^\top\alpha + (E_t^{k-1})^\top\mu \tag{2.29}$$

*where $(\alpha,\lambda,\mu)$ is such that there is $\delta$ satisfying $(\alpha,\lambda,\mu,\delta)$ is an optimal solution of dual problem (2.27) written for $x_{t-1} = x_{t-1}^k$.*

***Proof*** Defining

$$\mathcal{S}_k = \mathcal{X}_t\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}^n \cap C_k \cap D,$$

where

$$C_k = \left\{ (x_t, f, \theta, x_{t-1}) : \begin{cases} A_t^{k-1}x_t + B_t^{k-1}x_{t-1} + C_t^{k-1} \leq f\mathbf{e}, \\ \theta_{t+1}^{0:k-1} + \beta_{t+1}^{0:k-1}x_t \leq \theta\mathbf{e}, \\ D_t^{k-1}x_t + E_t^{k-1}x_{t-1} + H_t^{k-1} \leq 0 \end{cases} \right\}$$
$$D = \{(x_t, f, \theta, x_{t-1}) : A_t x_t + B_t x_{t-1} = b_t\},$$

we have

$$\underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k) = \begin{cases} \inf\ f + \theta + \mathbb{I}_{\mathcal{S}_k}(x_t, f, \theta, x_{t-1}^k) \\ x_t \in \mathbb{R}^n, f, \theta \in \mathbb{R}. \end{cases} \tag{2.30}$$

Using Theorem 24(a) in Rockafellar [29], we have

$$\begin{aligned}\beta_t^k \in \partial \underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k) &\Leftrightarrow (0,0,0,\beta_t^k) \in \partial(f+\theta+\mathbb{I}_{\mathcal{S}_k})(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k) \\ &\Leftrightarrow (0,0,0,\beta_t^k) \in [0;1;1;0] + \mathcal{N}_{\mathcal{S}_k}(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k),\end{aligned} \quad \begin{matrix}(2.31)\\(a)\end{matrix}$$

where $f_{tk}$ and $\theta_{tk}$ are the optimal values of, respectively, $f$ and $\theta$ in (2.26) written for $x_{t-1} = x_{t-1}^k$. For equivalence (2.31)-(a), we have used the fact that $(x_t, f, \theta, x_{t-1}) \to f+\theta$ and $\mathbb{I}_{\mathcal{S}_k}$ are proper, finite at $(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k)$, and the intersection of the relative interior of the domain of these functions, i.e., set $\text{ri}(\mathcal{S}_k)$, is nonempty. Next,

$$\begin{aligned}\mathcal{N}_{\mathcal{S}_k}(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k) &= \mathcal{N}_{C_k}(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k) + \mathcal{N}_D(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k) \\ &\quad + \mathcal{N}_{\mathcal{X}_t\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}^n}(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k),\end{aligned} \tag{2.32}$$

and standard calculus on normal cones gives

$$\begin{aligned}\mathcal{N}_{\mathcal{X}_t\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}^n}(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k) &= \mathcal{N}_{\mathcal{X}_t}(x_t^k)\times\{0\}\times\{0\}\times\{0\}, \\ \mathcal{N}_D(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k) &= \left\{[A_t^\top; 0; 0; B_t^\top]\lambda \ : \ \lambda \in \mathbb{R}^q\right\},\end{aligned} \tag{2.33}$$

and $\mathcal{N}_{C_k}(x_t^k, f_{tk}, \theta_{tk}, x_{t-1}^k)$ is the set of points of form

$$\begin{pmatrix}(A_t^{k-1})^\top\alpha + (\beta_{t+1}^{0:k-1})^\top\delta + (D_t^{k-1})^\top\mu \\ -\mathbf{e}^\top\alpha \\ -\mathbf{e}^\top\delta \\ (B_t^{k-1})^\top\alpha + (E_t^{k-1})^\top\mu\end{pmatrix} \tag{2.34}$$

where $\alpha, \delta, \mu$ satisfy

$$\begin{aligned}&\alpha, \delta, \mu \geq 0, \\ &\begin{pmatrix}\alpha\\ \delta\\ \mu\end{pmatrix}^\top \begin{pmatrix}A_t^{k-1}x_t^k + B_t^{k-1}x_{t-1}^k + C_t^{k-1} - f_{tk}\mathbf{e} \\ \theta_{t+1}^{0:k-1} + \beta_{t+1}^{0:k-1}x_t^k - \theta_{tk}\mathbf{e} \\ D_t^{k-1}x_t^k + E_t^{k-1}x_{t-1}^k + H_t^{k-1}\end{pmatrix} = 0.\end{aligned} \tag{2.35}$$

Combining (2.31), (2.32), (2.33), (2.34), we see that $\beta_t^k \in \partial \underline{\mathcal{Q}}_t^k(x_{t-1}^k)$ if and only if $\beta_t^k$ is of form (2.29) where $\alpha, \lambda, \mu$ satisfies (2.35) and

$$\begin{aligned}&0 \in \mathcal{N}_{\mathcal{X}_t}(x_t^k) + A_t^\top\lambda + (A_t^{k-1})^\top\alpha + (\beta_{t+1}^{0:k-1})\delta + (D_t^{k-1})^\top\mu, \\ &0 = 1 - \mathbf{e}^\top\alpha, \\ &0 = 1 - \mathbf{e}^\top\delta.\end{aligned} \tag{2.36}$$

Finally, it suffices to observe that $\alpha, \lambda, \mu$ satisfies (2.35) and (2.36) if and only if $\alpha, \lambda, \mu, \delta$ is an optimal solution of dual problem (2.27). □

Using the previous lemma and denoting by $(\alpha_t^k, \lambda_t^k, \mu_t^k, \delta_t^k)$ an optimal solution of (2.27) written for $x_{t-1} = x_{t-1}^k$, we have that

$$\beta_t^k = (B_t^{k-1})^\top \alpha_t^k + B_t^\top \lambda_t^k + (E_t^{k-1})^\top \mu_t^k \in \partial \underline{\mathcal{Q}}_t^{k-1}(x_{t-1}^k). \tag{2.37}$$

***Remark*** When $\mathcal{X}_t$ is polyhedral, formula (2.37) follows from Duality for linear programming. For a more general convex set $\mathcal{X}_t$, formula (2.37) directly follows from applying to value function $\underline{\mathcal{Q}}_t^{k-1}$ Lemma 2.1 in [6] or Proposition 3.2 in [9] which, respectively, provide a characterization of the subdifferential and subgradients for value functions of general convex optimization problems (whose argument is in the objective function and in linear and nonlinear coupling constraints of the corresponding optimization problem). The proof of Lemma 2.7 is a proof of relation (2.37) specializing to the particular case of value function $\underline{\mathcal{Q}}_t^{k-1}$ the proof of Lemma 2.1 in [6].

## 3 The StoDCuP (Stochastic Dynamic Cutting Plane) Algorithm

### 3.1 Problem Formulation and Assumptions

We consider multistage stochastic nonlinear optimization problems of the form

$$\begin{array}{c} \displaystyle\min_{x_1 \in X_1(x_0,\xi_1)} f_1(x_1, x_0, \xi_1) + \mathbb{E}\left[ \min_{x_2 \in X_2(x_1,\xi_2)} f_2(x_2, x_1, \xi_2) + \mathbb{E} \right. \\ \left. \left[ \ldots + \mathbb{E}\left[ \min_{x_T \in X_T(x_{T-1},\xi_T)} f_T(x_T, x_{T-1}, \xi_T) \right]\right]\right], \end{array} \tag{3.1}$$

where $x_0$ is given, $(\xi_t)_{t=2}^T$ is a stochastic process, $\xi_1$ is deterministic, and

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

In the constraint set above, $\mathcal{X}_t$ is polyhedral and $\xi_t$ contains in particular the random elements in matrices $A_t$, $B_t$, and vector $b_t$.

We make the following assumption on $(\xi_t)$:

(H0) $(\xi_t)$ is interstage independent and for $t = 2, \ldots, T$, $\xi_t$ is a random vector taking values in $\mathbb{R}^K$ with a discrete distribution and a finite support $\Theta_t = \{\xi_{t1}, \ldots, \xi_{tM_t}\}$ with $p_{ti} = \mathbb{P}(\xi_t = \xi_{ti})$, $i = 1, \ldots, M_t$, while $\xi_1$ is deterministic.

For this problem, we can write Dynamic Programming equations: the first stage problem is

$$\mathcal{Q}_1(x_0) = \begin{cases} \min_{x_1 \in \mathbb{R}^n} f_1(x_1, x_0, \xi_1) + \mathcal{Q}_2(x_1) \\ x_1 \in X_1(x_0, \xi_1) \end{cases} \tag{3.2}$$

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

$$\mathfrak{Q}_t(x_{t-1}, \xi_t) = \begin{cases} \min_{x_t \in \mathbb{R}^n} f_t(x_t, x_{t-1}, \xi_t) + \mathcal{Q}_{t+1}(x_t) \\ x_t \in X_t(x_{t-1}, \xi_t), \end{cases} \tag{3.3}$$

with the convention that $\mathcal{Q}_{T+1}$ is identically zero.

We set $\mathcal{X}_0 = \{x_0\}$ and make the following assumptions (H1)-Sto on the problem data:

(H1)-Sto: for $t = 1, \ldots, T$,

(1) $\mathcal{X}_t$ is a nonempty, compact, and polyhedral set.
(2) For every $j = 1, \ldots, M_t$, the function $f_t(\cdot, \cdot, \xi_{tj})$ is convex, proper, lower semicontinuous on $\mathcal{X}_t \times \mathcal{X}_{t-1}$ and $\mathcal{X}_t \times \mathcal{X}_{t-1} \subset \text{int}\,(\text{dom}(f_t(\cdot, \cdot, \xi_{tj})))$.
(3) For every $j = 1, \ldots, M_t$, each component $g_{ti}(\cdot, \cdot, \xi_{tj}), i = 1, \ldots, p$, of function $g_t(\cdot, \cdot, \xi_{tj})$ is convex, proper, lower semicontinuous such that $\mathcal{X}_t \times \mathcal{X}_{t-1} \subset \text{int}\,(\text{dom}(g_{ti}(\cdot, \cdot, \xi_{tj})))$.
(4) $X_1(x_0, \xi_1) \neq \emptyset$ and for every $t = 2, \ldots, T$, for every $j = 1, \ldots, M_t$, $\mathcal{X}_{t-1} \subset \text{int}\,(\text{dom}(X_t(\cdot, \xi_{tj})))$.

***Remark 3.1*** Nonlinear constraints of form $h_{ti}(x_t, \xi_t) \le 0$ or $h_{ti}(x_t) \le 0$ at stage $t$ can be handled, adding the corresponding component functions $h_{ti}$ in $g_t$, as long as (H1)-Sto is satisfied. In particular, convexity of $h_{ti}(\cdot, \xi_{tj})$ is required for $j = 1, \ldots, M_t$.

It is easy to show that under Assumption (H1)-Sto, functions $\mathcal{Q}_t$ are convex and Lipschitz continuous on $\mathcal{X}_{t-1}$:

**Lemma 3.2** *Let Assumption (H1)-Sto hold. Then $\mathcal{Q}_t$ is convex Lipschitz continuous on $\mathcal{X}_{t-1}$ for $t = 2, \ldots, T+1$.*

***Proof*** The proof is analogue to the proof of Lemma 2.1. □

### 3.2 Forward StoDCuP

The algorithm to be presented in this section for solving (3.1) is an extension of the DCuP algorithm to the stochastic case. All inequalities and equalities between random variables in the rest of the paper hold almost surely with respect to the sampling of the algorithm.

Due to Assumption (H0), the $\prod_{t=2}^{T} M_t$ realizations of $(\xi_t)_{t=1}^{T}$ form a scenario tree of depth $T+1$ where the root node $n_0$ associated with a stage 0 (with decision $x_0$ taken at that node) has one child node $n_1$ associated with the first stage (with $\xi_1$ deterministic).

We denote by $\mathcal{N}$ the set of nodes, by $\texttt{Nodes}(t)$ the set of nodes for stage $t$ 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 $n$;

- $\xi_n$: the realization of process $(\xi_t)$ at node $n$[1]: for a node $n$ of stage $t$, this realization $\xi_n$ contains in particular the realizations $b_n$ of $b_t$, $A_n$ of $A_t$, and $B_n$ of $B_t$.
- $\xi_{[n]}$: the history of the realizations of process $(\xi_t)$ from the first stage node $n_1$ to node $n$: for a node $n$ of stage $t$, the $i$th component of $\xi_{[n]}$ is $\xi_{\mathcal{P}^{t-i}(n)}$ for $i = 1, \ldots, t$, where $\mathcal{P} : \mathcal{N} \to \mathcal{N}$ is the function associating with a node its parent node (the empty set for the root node).

At each iteration of the algorithm, trial points are computed on a sampled scenario and lower bounding affine functions, called cuts in the sequel, are built for convex functions $\mathcal{Q}_t, t = 2, \ldots, T+1$, at these trial points. More precisely, at iteration $k$ denoting by $x_{t-1}^k$ the trial point for stage $t-1$, the cut

$$\mathcal{C}_t^k(x_{t-1}) = \theta_t^k + \langle \beta_t^k, x_{t-1} \rangle \tag{3.4}$$

is built for $\mathcal{Q}_t$ with the convention that $\mathcal{C}_{T+1}^k$ is the null function (see below for the computation of $\theta_t^k, \beta_t^k$). As in SDDP, we end up in iteration $k$ with an approximation $\mathcal{Q}_t^k$ of $\mathcal{Q}_t$ which is a maximum of $k+1$ affine functions: $\mathcal{Q}_t^k(x_{t-1}) = \max_{0 \le j \le k} \mathcal{C}_t^j(x_{t-1})$.

Additionally, the variant we propose builds cutting plane approximations of convex functions $f_t(\cdot,\cdot,\xi_{tj})$ and $g_{ti}(\cdot,\cdot,\xi_{tj})$, $t = 1, \ldots, T$, $i = 1, \ldots, p$, $j = 1, \ldots, M_t$, computing linearizations of these functions. At the end of iteration $k$, these approximations will be denoted by $f_{tj}^k$ and $g_{tij}^k$ for $f_t(\cdot,\cdot,\xi_{tj})$ and $g_{ti}(\cdot,\cdot,\xi_{tj})$, respectively, and take the form of a maximum of $k+1$ affine functions. We use the notation

$$\begin{aligned} f_{tj}^k(x_t, x_{t-1}) &= \max_{\ell=0,\ldots,k} \; a_{tj}^\ell x_t + b_{tj}^\ell x_{t-1} + c_{tj}^\ell, \\ g_{tij}^k(x_t, x_{t-1}) &= \max_{\ell=0,\ldots,k} \; d_{tij}^\ell x_t + e_{tij}^\ell x_{t-1} + h_{tij}^\ell, \end{aligned}$$

where $a_{tj}^\ell, b_{tj}^\ell, d_{tij}^\ell$, and $e_{tij}^\ell$ are $n$-dimensional row vectors. The trial points of iteration $k$ are computed before updating these functions, therefore using approximations $f_{tj}^{k-1}$, $g_{tij}^{k-1}$, and $\mathcal{Q}_{t+1}^{k-1}$ of $f_t(\cdot,\cdot,\xi_{tj})$, $g_{ti}(\cdot,\cdot,\xi_{tj})$, and $\mathcal{Q}_{t+1}$ available at the end of iteration $k-1$. These trial points are decisions computed at nodes $(n_1^k, n_2^k, \ldots, n_T^k)$ using these approximations, knowing that $n_1^k = n_1$, and for $t \ge 2$, $n_t^k$ is a node of stage $t$, child of node $n_{t-1}^k$, i.e., these nodes correspond to a sample $(\tilde{\xi}_1^k, \tilde{\xi}_2^k, \ldots, \tilde{\xi}_T^k)$ of $(\xi_1, \xi_2, \ldots, \xi_T)$. At iteration $k$, the linearizations for $f_t(\cdot,\cdot,\xi_{tj})$, $g_{ti}(\cdot,\cdot,\xi_{tj})$ (resp. $\mathcal{Q}_t$) are computed at $(x_m^k, x_n^k)$ (resp. $x_n^k$) where $n = n_{t-1}^k$, and $m$ is the child node of node $n$ such that $\xi_m = \xi_{tj}$. For convenience, for any node $m$ of stage $t$, we will denote by $j_t(m)$ the unique index $j_t(m)$ such that $\xi_m = \xi_{tj_t(m)}$. Before detailing the steps of StoDCuP, we need more notation: for all $k \ge 1, t = 1, \ldots, T$, $j = 1, \ldots, M_t$, let $X_{tj}^k : \mathcal{X}_{t-1} \rightrightarrows \mathcal{X}_t$ be the multifunction given by

$$X_{tj}^k(x_{t-1}) = \{x_t \in \mathcal{X}_t : \; g_{tij}^k(x_t, x_{t-1}) \le 0, i = 1, \ldots, p, \; A_{tj}x_t + B_{tj}x_{t-1} = b_{tj}\}, \tag{3.5}$$

[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 $(\xi_t)$ 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.

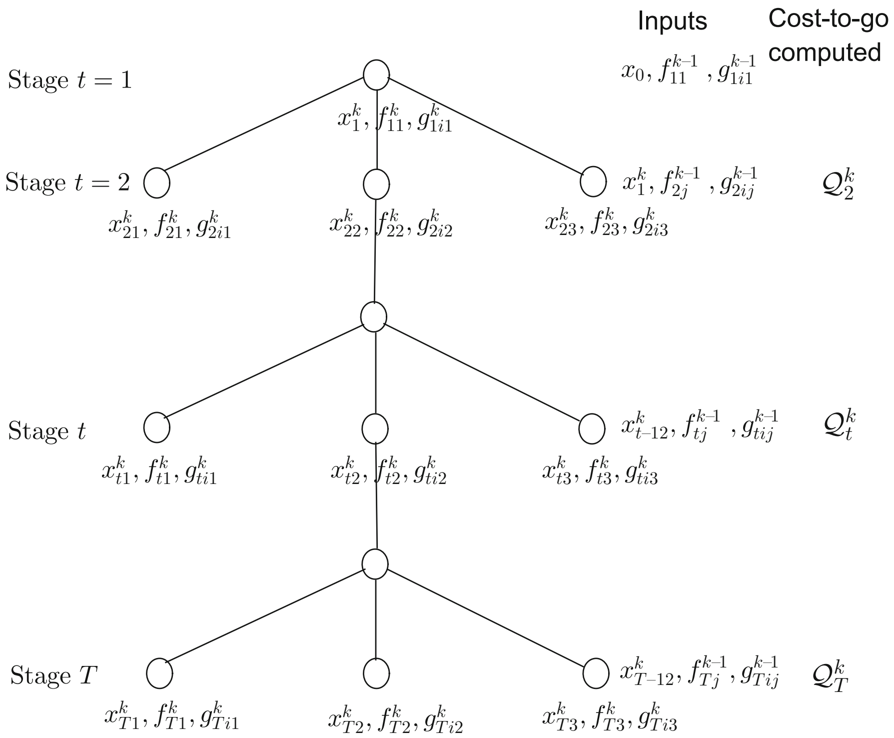


**Fig. 1** Variables updated in iteration $k$ of StoDCuP. In this representation, for simplicity, every node had 3 child nodes and the sampled scenario is $(\xi_1, \xi_{22}, \ldots, \xi_{t2}, \ldots, \xi_{T2})$ with corresponding decisions $(x_1^k, x_{22}^k, \ldots, x_{t2}^k, \ldots, x_{T2}^k)$. The decisions computed for the nodes of stage $t$ on this scenario are denoted on this figure by $x_{t1}^k, x_{t2}^k, x_{t3}^k$ for nodes with realization of $\xi_t$ given by respectively $\xi_{t1}, \xi_{t2}, \xi_{t3}$. For a given stage $t$, the inputs are $x_{t-12}^k$ (trial point), $f_{tj}^{k-1}, g_{tij}^{k-1}$ (for all $i, j$) while the outputs are $\mathcal{Q}_t^k$ and for node with realization $\xi_{tj}$ of $\xi_t$ decision $x_{tj}^k$ and functions $f_{tj}^k, g_{tij}^k$

where $A_{tj}, B_{tj}, b_{tj}$ are respectively the realizations of $A_t$, $B_t$, and $b_t$ in $\xi_{tj}$ and let $\underline{\mathfrak{Q}}_{tj}^k : \mathcal{X}_{t-1} \to \mathbb{R}$ be the function

$$\underline{\mathfrak{Q}}_{tj}^k(x_{t-1}) = \begin{cases} \min_{x_t} \;\; f_{tj}^k(x_t, x_{t-1}) + \mathcal{Q}_{t+1}^k(x_t) \\ x_t \in X_{tj}^k(x_{t-1}). \end{cases} \tag{3.6}$$

The detailed steps of the algorithm are described below (see the correspondence with DCuP). We refer to Fig. 1 for the representation of the variables updated in iteration $k$ of StoDCuP.

**Forward StoDCuP (Stochastic Dynamic Cutting Plane) with linearizations computed in a forward pass.**

Step 0) **Initialization.** For $t = 1, \ldots, T$, $j = 1, \ldots, M_t$, $i = 1, \ldots, p$, take $f_{tj}^0, g_{tij}^0 : \mathcal{X}_t \times \mathcal{X}_{t-1} \to \mathbb{R}$ affine functions satisfying $f_{tj}^0 \leq f_t(\cdot, \cdot, \xi_{tj})$, $g_{tij}^0 \leq g_{ti}(\cdot, \cdot, \xi_{tj})$, and for $t = 2, \ldots, T$, $\mathcal{Q}_t^0 : \mathcal{X}_{t-1} \to \mathbb{R}$ is an affine function satisfying $\mathcal{Q}_t^0 \leq \mathcal{Q}_t$. Set $x_{n_0} = x_0$, set the iteration count $k$ to 1, and $\mathcal{Q}_{T+1}^0 \equiv 0$.

Step 1) **Forward pass.** Set $\mathcal{C}_{T+1}^k = \mathcal{Q}_{T+1}^k \equiv 0$ and $x_0^k = x_0$.
Generate a sample $(\tilde{\xi}_1^k, \tilde{\xi}_2^k, \ldots, \tilde{\xi}_T^k)$ of $(\xi_1, \xi_2, \ldots, \xi_T)$ corresponding to a set of nodes $(n_1^k, n_2^k, \ldots, n_T^k)$ where $n_1^k = n_1$, and for $t \geq 2$, $n_t^k$ is a node of stage $t$, child of node $n_{t-1}^k$. Set $n_0^k = n_0$.
**For** $t = 1, \ldots, T$, do:
Let $n = n_{t-1}^k$.
**For** every $m \in C(n)$,

a) compute an optimal solution $x_m^k$ of

$$\underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k) = \begin{cases} \min\limits_{x_m} \; f_{tj_t(m)}^{k-1}(x_m, x_n^k) + \mathcal{Q}_{t+1}^{k-1}(x_m) \\ x_m \in X_{tj_t(m)}^{k-1}(x_n^k) \end{cases} \tag{3.7}$$

where we recall that

$$\begin{aligned} f_{tj_t(m)}^{k-1}(x_m, x_n^k) &= \max_{\ell=0,\ldots,k-1} \; a_{tj_t(m)}^{\ell} x_m + b_{tj_t(m)}^{\ell} x_n^k + c_{tj_t(m)}^{\ell}, \\ \mathcal{Q}_{t+1}^{k-1}(x_m) &= \max_{0 \leq \ell \leq k-1} \; \mathcal{C}_{t+1}^{\ell}(x_m). \end{aligned}$$

b) Compute function values and subgradients of convex functions $f_t(\cdot, \cdot, \xi_m)$ and $g_{ti}(\cdot, \cdot, \xi_m)$ at $(x_m^k, x_n^k)$ and let $\ell_{f_t(\cdot,\cdot,\xi_m)}((\cdot, \cdot); (x_m^k, x_n^k))$ and $\ell_{g_{ti}(\cdot,\cdot,\xi_m)}((\cdot, \cdot); (x_m^k, x_n^k))$ denote the corresponding linearizations.
c) Set

$$\begin{aligned} f_{tj_t(m)}^k &= \max\Big(f_{tj_t(m)}^{k-1}\,,\; \ell_{f_t(\cdot,\cdot,\xi_m)}((\cdot, \cdot); (x_m^k, x_n^k))\Big), \\ g_{ti}^k &= \max\Big(g_{ti}^{k-1}\,,\; \ell_{g_{ti}(\cdot,\cdot,\xi_m)}((\cdot, \cdot); (x_m^k, x_n^k))\Big), \quad \forall i = 1, \ldots, p. \end{aligned}$$

d) If $t \geq 2$ then compute $\beta_{tm}^k \in \partial \underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k)$.

**End For**
**If** $t \geq 2$ compute:

$$\beta_t^k = \sum_{m \in C(n)} p_m \beta_{tm}^k,$$
$$\theta_t^k = \sum_{m \in C(n)} p_m \Big[ \underline{\mathfrak{Q}}_{t j_t(m)}^{k-1}(x_n^k) - \langle \beta_{tm}^k, x_n^k \rangle \Big], \tag{3.8}$$

yielding the new cut $\mathcal{C}_t^k(x_{t-1}) = \theta_t^k + \langle \beta_t^k, x_{t-1} \rangle$ and $\mathcal{Q}_t^k = \max\{\mathcal{Q}_t^{k-1}, \mathcal{C}_t^k\}$.

**End If**
**End For**
Step 2) Do $k \leftarrow k+1$ and go to Step 2).

---

The following assumption will be made on the sampling process in StoDCuP:
(H2) The samples of $(\xi_t)$ generated in StoDCuP 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 $\xi^k$, $k \geq 1$, are independent.

Recall that there are $\prod_{t=2}^T M_t$ possible scenarios (realizations) for $(\xi_2, \ldots, \xi_T)$. Moreover, by (H2), for every such scenario $s_j$, $j = 1, \ldots, \prod_{t=2}^T M_t$, the events $E_n = \{\xi^n = s_j\}, n \geq 1$, are independent and have a positive probability that only depends on $j$. This gives $\sum_{n \geq 1} \mathbb{P}(E_n) = \infty$ and by the Borel–Cantelli lemma, this implies that $\mathbb{P}(\overline{\lim}_{n \to \infty} E_n) = 1$. In what follows, several relations hold almost surely. In this case, the corresponding event of probability 1 is $\overline{\lim}_{n \to \infty} E_n$ corresponding to those realizations of StoDCuP where every scenario $s_j$ is sampled an infinite number of times.

***Remark 3.3*** As a consequence of the previous observation, for every realization of StoDCuP, and every node $n$ of the scenario tree, an infinite number of scenarios sampled in StoDCuP pass through that node $n$.

We have for StoDCuP the following analogue of Lemma 2.4 for DCuP (the proof is similar to the proof of Lemma 2.4):

**Lemma 3.4** *Let Assumptions (H0) and (H1)-Sto hold. Then, the following statements hold for StoDCuP:*

(a) *For $t = 2, \ldots, T$, the sequence $\{\beta_t^k\}_{k=1}^\infty$ is almost surely bounded.*
(b) *There exists $L \geq 0$ such that for each $t = 2, \ldots, T$, $\mathcal{Q}_t^k$ is $L$-Lipschitz continuous on $\mathcal{X}_{t-1}$ for every $k \geq 1$.*
(c) *There exists $\hat{L} \geq 0$ such that for each $t = 1, \ldots, T$, $j = 1, \ldots, M_t$, functions $f_{tj}^k$ and $g_{tij}^k$ are $\hat{L}$-Lipschitz continuous on $\mathcal{X}_t \times \mathcal{X}_{t-1}$ for every $k \geq 1$ and $i = 1, \ldots, p$.*

***Remark 3.5*** (*On the cuts and linearizations computed*) Assumption (H0) is fundamental for StoDCuP, due to the following claim:

(C) StoDCuP builds a cut for $\mathcal{Q}_t, t = 2, \ldots, T$, on any sampled scenario and a single cut for each of the functions $f_t(\cdot, \cdot, \xi_{tj}), g_{ti}(\cdot, \cdot, \xi_{tj}), t = 1, \ldots, T, j = 1, \ldots, M_t, i = 1, \ldots, p$, at each iteration.

The validity of the formulas of the cuts for $\mathcal{Q}_t$ will be checked in Lemma 3.8. The fact that a single cut is built for functions $f_t(\cdot, \cdot, \xi_{tj}), g_{ti}(\cdot, \cdot, \xi_{tj}), i = 1, \ldots, p$, $t = 1, \ldots, T, j = 1, \ldots, M_t$, comes from the fact that at iteration $k$ and stage $t$ a cut is built for each of functions $f_t(\cdot, \cdot, \xi_m), g_{ti}(\cdot, \cdot, \xi_m), i = 1, \ldots, p, m \in C(n)$, where $n = n_{t-1}^k$, and due to Assumption (H0), to each $m \in C(n)$, corresponds one and only one index $j = j_t(m)$ such that $\xi_m = \xi_{tj} = \xi_{tj_t(m)}$.

***Remark 3.6*** The algorithm can be extended to solve risk-averse problems. It was shown in [12] that dynamic programming equations can be written and that SDDP can be applied for multistage stochastic linear optimization problems which minimize some extended polyhedral risk measure of the cost. As a special case, spectral risk measures are considered in [13] where analytic formulas for some cut coefficients computed by SDDP are available. Similarly, StoDCuP can be extended to solve multistage nonlinear optimization problems with objective and constraint functions as in (3.1) if instead of minimizing the expected cost we minimize an extended polyhedral risk measure of the cost, as long as Assumptions (H0) and (H1)-Sto are satisfied. It is also possible to apply StoDCuP to solve risk-averse dynamic programming equations with nested conditional risk measures (see [30,31] for details on conditional risk mappings) and objective and constraint functions as in (3.1), again, as long as Assumptions (H0) and (H1)-Sto are satisfied. Using SDDP in this risk-averse setting was proposed in [32].

We can simulate the policy obtained after $k-1$ iterations of StoDCuP and define decisions $x_n^k$ at each node $n$ of the scenario tree as follows:

**Simulation of StoDCuP after $k-1$ iterations.**

Set $x_{n_0}^k = x_0$.
**For** $t = 1, \ldots, T$,
  **For** every node $n \in \texttt{Nodes}(t-1)$,
    **For** every $m \in C(n)$,
      compute an optimal solution $x_m^k$ of

$$\underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k) = \begin{cases} \min\limits_{x_m} \ f_{tj_t(m)}^{k-1}(x_m, x_n^k) + \mathcal{Q}_{t+1}^{k-1}(x_m) \\ x_m \in X_{tj_t(m)}^{k-1}(x_n^k). \end{cases} \tag{3.9}$$

    **End For**
  **End For**
**End For**

We close this section providing in Lemma 3.7 simple relations involving the linearizations of the objective and constraint functions that will be used for the convergence analysis of StoDCuP.

**Lemma 3.7** *Let Assumption (H1)-Sto hold. For every $t = 1, \ldots, T$, $j = 1, \ldots, M_t$, $i = 1, \ldots, p$, we have almost surely*

$$f_t(x_t, x_{t-1}, \xi_{tj}) \geq f^k_{tj}(x_t, x_{t-1}), \, g_{ti}(x_t, x_{t-1}, \xi_{tj}) \geq g^k_{tij}(x_t, x_{t-1}),$$
$$\forall k \geq 0, \forall x_t \in \mathcal{X}_t, \forall x_{t-1} \in \mathcal{X}_{t-1}, \tag{3.10}$$

*and for every $k \geq 1$,*

$$X_t(x_{t-1}, \xi_{tj}) \subset X^k_{tj}(x_{t-1}), \;\; \forall \, x_{t-1} \in \mathcal{X}_{t-1}. \tag{3.11}$$

*For all $t = 1, \ldots, T$, $i = 1, \ldots, p$, for all $n \in \texttt{Nodes}(t-1)$, for all $k \in \mathcal{S}_n$, we have for all $m \in C(n)$:*

$$f_t(x^k_m, x^k_n, \xi_m) = f^k_{tj_t(m)}(x^k_m, x^k_n) \text{ and } g_{ti}(x^k_m, x^k_n, \xi_m) = g^k_{tij_t(m)}(x^k_m, x^k_n), \;\; \textit{almost surely} \tag{3.12}$$

*For all $t = 1, \ldots, T$, $i = 1, \ldots, p$, for all $n \in \texttt{Nodes}(t-1)$, for all $k \geq 1$, for all $m \in C(n)$, we have*

$$g^{k-1}_{tij_t(m)}(x^k_m, x^k_n) \leq 0, \;\; \textit{almost surely}, \tag{3.13}$$
$$0 \leq \max(g_{ti}(x^k_m, x^k_n, \xi_m), 0) \leq g_{ti}(x^k_m, x^k_n, \xi_m) - g^{k-1}_{tij_t(m)}(x^k_m, x^k_n), \;\; \textit{almost surely} \tag{3.14}$$

***Proof*** Let us show (3.10). The relation holds for $k = 0$. Now let us fix $t \in \{1, \ldots, T\}$, $j \in \{1, \ldots, M_t\}$, $k \geq 1$ and $\ell \in \{1, \ldots, k\}$. At iteration $\ell$, setting $n = n^\ell_{t-1}$, there exists one and only one node $m$ in the set $C(n)$ such that $\xi_m = \xi_{tj}$ with $j = j_t(m)$ and by the subgradient inequality for every $x_t \in \mathcal{X}_t$, for every $x_{t-1} \in \mathcal{X}_{t-1}$, we have

$$\begin{aligned} f_t(x_t, x_{t-1}, \xi_{tj}) &= f_t(x_t, x_{t-1}, \xi_m) \geq \ell_{f_t(\cdot,\cdot,\xi_m)}(x_t, x_{t-1}; (x^\ell_m, x^\ell_n)) \\ g_{ti}(x_t, x_{t-1}, \xi_{tj}) &= g_{ti}(x_t, x_{t-1}, \xi_m) \geq \ell_{g_{ti}(\cdot,\cdot,\xi_m)}(x_t, x_{t-1}; (x^\ell_m, x^\ell_n)), \end{aligned} \tag{3.15}$$

which, by Step c) of StoDCuP, immediately implies (3.10) and clearly inclusion (3.11) is a consequence of (3.10).

Take $t \in \{1, \ldots, T\}$, $i \in \{1, \ldots, p\}$, take a node $n \in \texttt{Nodes}(t-1)$ and $k \in \mathcal{S}_n$. Then for any $m \in C(n)$, a linearization is built for $f_t(\cdot, \cdot, \xi_m)$ and $g_{ti}(\cdot, \cdot, \xi_m)$ at $(x^k_m, x^k_n)$. Therefore,

$$\begin{aligned} f_t(x^k_m, x^k_n, \xi_m) &\overset{(3.10)}{\geq} f^k_{tj_t(m)}(x^k_m, x^k_n) \\ &\geq \ell_{f_t(\cdot,\cdot,\xi_m)}(x^k_m, x^k_n; (x^k_m, x^k_n)) = f_t(x^k_m, x^k_n, \xi_m) \text{ since } n^k_{t-1} = n, \\ g_{ti}(x^k_m, x^k_n, \xi_m) &\overset{(3.10)}{\geq} g^k_{tij_t(m)}(x^k_m, x^k_n) \\ &\geq \ell_{g_{ti}(\cdot,\cdot,\xi_m)}(x^k_m, x^k_n; (x^k_m, x^k_n)) = g_{ti}(x^k_m, x^k_n, \xi_m), \text{ since } n^k_{t-1} = n, \end{aligned}$$

and (3.12) follows.

Relation (3.13) comes from the fact that $x_m^k \in X_{tj_t(m)}^{k-1}(x_n^k)$ by definition of $x_m^k$ (see the simulation of StoDCuP).

Finally take a realization $\omega$ of StoDCuP. We show that

$$\begin{aligned} 0 &\leq \max(g_{ti}(x_m^k(\omega), x_n^k(\omega), \xi_m), 0) \\ &\leq g_{ti}(x_m^k(\omega), x_n^k(\omega), \xi_m) - g_{tij_t(m)}^{k-1}(\omega)(x_m^k(\omega), x_n^k(\omega)). \end{aligned} \tag{3.16}$$

If $g_{ti}(x_m^k(\omega), x_n^k(\omega), \xi_m) \leq 0$ then (3.16) holds because $g_{ti}(\cdot, \cdot, \xi_m) \geq g_{tij_t(m)}^{k-1}(\omega)$ and if $g_{ti}(x_m^k(\omega), x_n^k(\omega), \xi_m) > 0$ then (3.16) holds too because of inequality (3.13). Therefore, (3.16) holds. □

### 3.3 Implementation Details for Steps b) and d) of StoDCuP

In this section, we explain how to compute variables $a_{tj}^\ell$, $b_{tj}^\ell$, $c_{tj}^\ell$, $d_{tij}^\ell$, $e_{tij}^\ell$, $h_{tij}^\ell$, as well as cut coefficients $\beta_{tm}^k$ in StoDCuP.

In Step b) of StoDCuP, we compute an arbitrary subgradient $[s_1; s_2]$ of convex function $f_t(\cdot, \cdot, \xi_m)$ at $(x_m^k, x_n^k)$ where $s_1, s_2 \in \mathbb{R}^n$ and set $a_{tj_t(m)}^k = s_1^\top$ and $b_{tj_t(m)}^k = s_2^\top$. For $i = 1, \ldots, p$, we also compute an arbitrary subgradient $[s_{1i}; s_{2i}]$ of convex function $g_{ti}(\cdot, \cdot, \xi_m)$ at $(x_m^k, x_n^k)$ where $s_{1i}, s_{2i} \in \mathbb{R}^n$; we set $d_{tij_t(m)}^k = s_{1i}^\top$, $e_{tij_t(m)}^k = s_{2i}^\top$, and compute

$$\begin{aligned} c_{tj_t(m)}^k &= f_t(x_m^k, x_n^k, \xi_m) - a_{tj_t(m)}^k x_m^k - b_{tj_t(m)}^k x_n^k, \\ h_{tij_t(m)}^k &= g_{ti}(x_m^k, x_n^k, \xi_m) - d_{tij_t(m)}^k x_m^k - e_{tij_t(m)}^k x_n^k. \end{aligned}$$

For the computation of $\beta_{tm}^k$, it is convenient to introduce $k \times n$ matrices

$$A_{tj}^k = \begin{bmatrix} a_{tj}^0 \\ a_{tj}^1 \\ \vdots \\ a_{tj}^k \end{bmatrix}, \; B_{tj}^k = \begin{bmatrix} b_{tj}^0 \\ b_{tj}^1 \\ \vdots \\ b_{tj}^k \end{bmatrix}, \; D_{tij}^k = \begin{bmatrix} d_{tij}^0 \\ d_{tij}^1 \\ \vdots \\ d_{tij}^k \end{bmatrix}, E_{tij}^k = \begin{bmatrix} e_{tij}^0 \\ e_{tij}^1 \\ \vdots \\ e_{tij}^k \end{bmatrix}, \; \beta_t^{0:k} = \begin{bmatrix} (\beta_t^0)^\top \\ (\beta_t^1)^\top \\ \vdots \\ (\beta_t^k)^\top \end{bmatrix}, \tag{3.17}$$

$k$-dimensional vectors,

$$C_{tj}^k = \begin{bmatrix} c_{tj}^0 \\ c_{tj}^1 \\ \vdots \\ c_{tj}^k \end{bmatrix}, \; H_{tij}^k = \begin{bmatrix} h_{tij}^0 \\ h_{tij}^1 \\ \vdots \\ h_{tij}^k \end{bmatrix}, \text{ and } \theta_t^{0:k} = \begin{bmatrix} \theta_t^0 \\ \theta_t^1 \\ \vdots \\ \theta_t^k \end{bmatrix}, \tag{3.18}$$

and matrices and vectors

$$D_{tj}^k = \begin{bmatrix} D_{t1j}^k \\ D_{t2j}^k \\ \vdots \\ D_{tpj}^k \end{bmatrix}, \; E_{tj}^k = \begin{bmatrix} E_{t1j}^k \\ E_{t2j}^k \\ \vdots \\ E_{tpj}^k \end{bmatrix}, \; H_{tj}^k = \begin{bmatrix} H_{t1j}^k \\ H_{t2j}^k \\ \vdots \\ H_{tpj}^k \end{bmatrix}. \tag{3.19}$$

If $\mathcal{X}_t = \{x_t : \mathbb{X}_t x_t \geq \bar{x}_t\}$, we can write problem (3.6) as

$$\underline{\mathfrak{Q}}_{tj}^k(x_{t-1}) = \begin{cases} \min_{x_t, f, \theta} \ f + \theta \\ f\mathbf{e} \geq A_{tj}^k x_t + B_{tj}^k x_{t-1} + C_{tj}^k, \\ A_{tj} x_t + B_{tj} x_{t-1} = b_{tj}, \\ D_{tj}^k x_t + E_{tj}^k x_{t-1} + H_{tj}^k \leq 0, \\ \theta \mathbf{e} \geq \theta_{t+1}^{0:k} + \beta_{t+1}^{0:k} x_t, \ \mathbb{X}_t x_t \geq \bar{x}_t. \end{cases} \tag{3.20}$$

Due to Assumption (H1)-Sto-4), for every $x_{t-1} \in \mathcal{X}_{t-1}$ and $j = 1, \ldots, M_t$, there exists $x_t \in \mathcal{X}_t$ such that $A_{tj} x_t + B_{tj} x_{t-1} = b_{tj}$, and $g_{ti}(x_t, x_{t-1}, \xi_{tj}) \leq 0$, $i = 1, \ldots, p$, which implies $g_{tij}^k(x_t, x_{t-1}) \leq 0, i = 1, \ldots, p$, $D_{tj}^k x_t + E_{tj}^k x_{t-1} + H_{tj}^k \leq 0$ and therefore the above problem (3.20) is feasible. Recalling (H1)-Sto-1), this linear program has a finite optimal value. Therefore, this optimal value is the optimal value of the dual problem and can be expressed as:

$$\underline{\mathfrak{Q}}_{tj}^k(x_{t-1}) = \begin{cases} \max\limits_{\alpha,\mu,\delta,\nu,\lambda} \ \alpha^\top (B_{tj}^k x_{t-1} + C_{tj}^k) + \mu^\top (E_{tj}^k x_{t-1} + H_{tj}^k) + \delta^\top \theta_{t+1}^{0:k} \\ \quad + \lambda^\top (b_{tj} - B_{tj} x_{t-1}) + \nu^\top \bar{x}_t \\ (A_{tj}^k)^\top \alpha + (D_{tj}^k)^\top \mu + (\beta_{t+1}^{0:k})^\top \delta - \mathbb{X}_t^\top \nu - (A_{tj})^\top \lambda = 0, \\ \mathbf{e}^\top \alpha = 1, \ \mathbf{e}^\top \delta = 1, \alpha, \mu, \delta, \nu \geq 0. \end{cases}$$

The above representation of $\underline{\mathfrak{Q}}_{tj}^k$ allows us to obtain the formulas for $\beta_{tm}^k, \beta_t^k, \theta_t^k$. More precisely, consider iteration $k$ and stage $t \geq 2$ of the forward pass of StoDCuP. Setting $n = n_{t-1}^k$ and $m \in C(n)$, let $(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k)$ be an optimal solution of the dual problem

$$\begin{array}{l} \max\limits_{\alpha,\mu,\delta,\nu,\lambda} \ \alpha^\top (B_{tj_t(m)}^{k-1} x_n^k + C_{tj_t(m)}^{k-1}) + \mu^\top (E_{tj_t(m)}^{k-1} x_n^k + H_{tj_t(m)}^{k-1}) + \delta^\top \theta_{t+1}^{0:k-1} + \lambda^\top (b_{tj_t(m)} - B_{tj_t(m)} x_n^k) + \nu^\top \bar{x}_t \\ (A_{tj_t(m)}^{k-1})^\top \alpha + (D_{tj_t(m)}^{k-1})^\top \mu + (\beta_{t+1}^{0:k-1})^\top \delta - \mathbb{X}_t^\top \nu - (A_{tj_t(m)})^\top \lambda = 0, \\ \mathbf{e}^\top \alpha = 1, \ \mathbf{e}^\top \delta = 1, \alpha, \mu, \delta, \nu \geq 0. \end{array} \tag{3.21}$$

By the discussion above, the optimal value of (3.21) is $\underline{\mathfrak{Q}}_{tj}^k(x_n^k)$. We now show in Lemma 3.8 that we can choose in StoDCuP,

$$\begin{aligned} \beta_{tm}^k &= \sum_{m \in C(n)} (B_{tj_t(m)}^{k-1})^\top \alpha_m^k + (E_{tj_t(m)}^{k-1})^\top \mu_m^k - B_{tj_t(m)}^\top \lambda_m^k, \\ \beta_t^k &= \sum_{m \in C(n)} p_m \Big[ (B_{tj_t(m)}^{k-1})^\top \alpha_m^k + (E_{tj_t(m)}^{k-1})^\top \mu_m^k - B_{tj_t(m)}^\top \lambda_m^k \Big], \\ \theta_t^k &= \sum_{m \in C(n)} p_m \Big[ \langle \alpha_m^k, C_{tj_t(m)}^{k-1} \rangle + \langle \mu_m^k, H_{tj_t(m)}^{k-1} \rangle + \langle \delta_m^k, \theta_{t+1}^{0:k-1} \rangle \\ &\qquad + \langle \lambda_m^k, b_{tj_t(m)} \rangle + \langle \nu_m^k, \bar{x}_t \rangle \Big]. \end{aligned} \tag{3.22}$$

More precisely, we show in Lemma 3.8 that computations (3.22) provide valid cuts (lower bounding functions $\mathcal{C}_t^k$) for $\mathcal{Q}_t$, in particular that $\beta_{tm}^k \in \partial \underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k)$, as required by Step d) of StoDCuP, and $\beta_t^k \in \partial \mathcal{Q}_t(x_n^k)$:

**Lemma 3.8** *Let Assumptions (H0) and (H1)-Sto hold. For every $t = 2, \ldots, T+1$, for every $k \geq 1$, we have almost surely*

$$\mathcal{Q}_t(x_{t-1}) \geq \mathcal{C}_t^k(x_{t-1}) \text{ and } \mathcal{Q}_t(x_{t-1}) \geq \underline{\mathcal{Q}}_t^k(x_{t-1}), \ \forall x_{t-1} \in \mathcal{X}_{t-1}. \tag{3.23}$$

*For all $t = 1, \ldots, T$, $j = 1, \ldots, M_t$, for every $k \geq 1$, we have almost surely*

$$\underline{\mathfrak{Q}}_{tj}^k(x_{t-1}) \leq \mathfrak{Q}_t(x_{t-1}, \xi_{tj}) \text{ for all } x_{t-1} \in \mathcal{X}_{t-1}. \tag{3.24}$$

*For all $t = 2, \ldots, T$, for every $k \geq 1$, defining $\underline{\mathcal{Q}}_t^{k-1}(x_n^k) = \sum_{j=1}^{M_t} p_{tj} \underline{\mathfrak{Q}}_{tj}^{k-1}(x_n^k)$, we have for every $n \in \texttt{Nodes}(t-1)$ and for all $k \in \mathcal{S}_n$:*

$$\underline{\mathcal{Q}}_t^{k-1}(x_n^k) = \mathcal{C}_t^k(x_n^k), \ \text{almost surely} \tag{3.25}$$

***Proof*** Let us show (3.23)–(3.24) by backward induction on $t$. Relation (3.23) clearly holds for $t = T+1$. Now assume that for some $t \in \{1, \ldots, T\}$, we have $\mathcal{Q}_{t+1}(x_t) \geq \underline{\mathcal{Q}}_{t+1}^k(x_t)$ for all $x_t \in \mathcal{X}_t$ and all $k \geq 1$. Using Lemma 3.7, we have for all $k \geq 1$, for all $j = 1, \ldots, M_t$, for all $x_t \in \mathcal{X}_t$, $x_{t-1} \in \mathcal{X}_{t-1}$, that $f_{tj}^k(x_t, x_{t-1}) \leq f_t(x_t, x_{t-1}, \xi_{tj})$ and $X_t(x_{t-1}, \xi_{tj}) \subset X_{tj}^k(x_{t-1})$, which, together with the induction hypothesis $\underline{\mathcal{Q}}_{t+1}^k \leq \mathcal{Q}_{t+1}$, implies

$$\underline{\mathfrak{Q}}_{tj}^k(x_{t-1}) \leq \mathfrak{Q}_t(x_{t-1}, \xi_{tj}) \text{ for all } x_{t-1} \in \mathcal{X}_{t-1}, \tag{3.26}$$

i.e., (3.24). Now observe that due to Assumption (H1)-Sto, for every $x_{t-1} \in \mathcal{X}_{t-1}$, the optimization problem

$$\underline{\mathfrak{Q}}_{tj}^{k-1}(x_{t-1}) = \begin{cases} \min_{x_t} \ f_{tj}^{k-1}(x_t, x_{t-1}) + \underline{\mathcal{Q}}_{t+1}^{k-1}(x_t) \\ x_t \in X_{tj}^{k-1}(x_{t-1}), \end{cases}$$

is a linear program with feasible set that is bounded (since $\mathcal{X}_t$ is compact) and nonempty (it contains the nonempty set $X_t(x_{t-1})$). Therefore, it has a finite optimal value which is also the optimal value of the dual problem given by

$$\underline{\mathfrak{Q}}_{tj}^{k-1}(x_{t-1}) = \begin{cases} \max_{\alpha,\mu,\delta,\nu,\lambda} \ \mathcal{D}_{tj}^{k-1}(\alpha, \mu, \delta, \nu, \lambda; x_{t-1}) \\ (A_{tj}^{k-1})^\top \alpha + (D_{tj}^{k-1})^\top \mu + (\beta_{t+1}^{0:k-1})^\top \delta - \mathbb{X}_t^\top \nu - (A_{tj})^\top \lambda = 0, \\ \mathbf{e}^\top \alpha = 1, \ \mathbf{e}^\top \delta = 1, \alpha, \mu, \delta, \nu \geq 0, \end{cases} \tag{3.27}$$

where

$$\begin{aligned} \mathcal{D}_{tj}^{k-1}(\alpha, \mu, \delta, \nu, \lambda; x_{t-1}) = \alpha^\top (B_{tj}^{k-1} x_{t-1} + C_{tj}^{k-1}) + \mu^\top (E_{tj}^{k-1} x_{t-1} + H_{tj}^{k-1}) \\ + \delta^\top \theta_{t+1}^{0:k-1} + \lambda^\top (b_{tj} - B_{tj} x_{t-1}) + \nu^\top \bar{x}_t. \end{aligned}$$

Now assume that $t \geq 2$. Let us take $m \in C(n_{t-1}^k)$. Recall that $j_t(m)$ is the unique index $j$ such that $\xi_{tj} = \xi_m$. Clearly $(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k)$ is feasible for dual problem (3.27) written for $j = j_t(m)$ and therefore for any $x_{t-1} \in \mathcal{X}_{t-1}$ we have

$$\underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_{t-1}) \geq \mathcal{D}_{tj_t(m)}^{k-1}(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k; x_{t-1}), \tag{3.28}$$

which gives

$$\begin{aligned}
\mathcal{Q}_t(x_{t-1}) &= \sum_{j=1}^{M_t} p_{tj}\mathfrak{Q}_t(x_{t-1}, \xi_{tj}) \\
&\overset{(H0)}{=} \sum_{m\in C(n_{t-1}^k)} p_m\mathfrak{Q}_t(x_{t-1}, \xi_m) \\
&= \sum_{m\in C(n_{t-1}^k)} p_m\mathfrak{Q}_t(x_{t-1}, \xi_{tj_t(m)}) \\
&\overset{(3.26)}{\geq} \sum_{m\in C(n_{t-1}^k)} p_m\underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_{t-1}) \\
&\overset{(3.28)}{\geq} \sum_{m\in C(n_{t-1}^k)} p_m\mathcal{D}_{tj_t(m)}^{k-1}(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k; x_{t-1}) \\
&= \mathcal{C}_t^k(x_{t-1}),
\end{aligned}$$

for every $x_{t-1} \in \mathcal{X}_{t-1}$, where for the last equality, we have used (3.4) and (3.22). Therefore, we have shown (3.23).

Now take $n \in$ `Nodes`$(t-1)$ and $k \in \mathcal{S}_n$. Then by definition of $(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k)$ and of $\mathcal{C}_t^k$, we get for any $m \in C(n)$:

$$\underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k) = \mathcal{D}_{tj_t(m)}^{k-1}(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k; x_n^k) \tag{3.29}$$

and

$$\begin{aligned}
\mathcal{C}_t^k(x_n^k) &= \sum_{m\in C(n)} p_m\mathcal{D}_{tj_t(m)}^{k-1}(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k; x_n^k) \\
&= \sum_{m\in C(n)} p_m\underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k) = \underline{\mathcal{Q}}_t^{k-1}(x_n^k).
\end{aligned} \tag{3.30}$$

□

### 3.4 Convergence Analysis

In what follows, if the stage associated with node $n$ is $\tau(n)$, we use the notation

$$\mathcal{S}_n = \{k \in \mathbb{N}^* : n_{\tau(n)}^k = n\}. \tag{3.31}$$

In other words, $\mathcal{S}_n$ the set of iterations $k$ where the sampled scenario passes through node $n$.

**Theorem 3.9** (Convergence of StoDCuP) *Let Assumption (H0), (H1)-Sto, and (H2) hold. Then*

(i) *for every* $t = 1, \ldots, T$*,* $i = 1, \ldots, p$*, almost surely*

$$\lim_{k \to +\infty} \max(g_{ti}(x_m^k, x_n^k, \xi_m), 0) = 0, \;\; \forall m \in \texttt{Nodes}(t), n = \mathcal{P}(m). \quad (3.32)$$

*For all* $t = 2, \ldots, T+1$*, for all node* $n \in \texttt{Nodes}(t-1)$*, we have almost surely*

$$\mathcal{H}(t): \lim_{k \to +\infty} \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) = 0. \quad (3.33)$$

(ii) *The limit of the sequence of first stage problems optimal values* $(f_{11}^{k-1}(x_{n_1}^k, x_0) + \mathcal{Q}_2^{k-1}(x_{n_1}^k))_{k \geq 1}$ *is the optimal value* $\mathcal{Q}_1(x_0)$ *of* (3.2) *and any accumulation point of the sequence* $(x_{n_1}^k)$ *is an optimal solution to the first stage problem* (3.2)*.*

***Proof*** We first show (3.32). Let us fix $t \in \{1, \ldots, T\}$, $i \in \{1, \ldots, p\}$, $m \in \texttt{Nodes}(t)$, $n = \mathcal{P}(m)$. Recall from Lemma 3.7 that

$$0 \leq \max(g_{ti}(x_m^k, x_n^k, \xi_m), 0) \leq g_{ti}(x_m^k, x_n^k, \xi_m) - g_{tij_t(m)}^{k-1}(x_m^k, x_n^k). \quad (3.34)$$

We now show that

$$\lim_{k \to +\infty} g_{ti}(x_m^k, x_n^k, \xi_m) - g_{tij_t(m)}^{k-1}(x_m^k, x_n^k) = 0, \quad (3.35)$$

which will show (3.32) due to relation (3.34).

Recalling that set $\mathcal{S}_n$ is infinite (see Remark 3.3), we denote by $k(1), k(2), \ldots,$ the iterations in $\mathcal{S}_n$ with $k(i) < k(i+1)$: $\mathcal{S}_n = \{k(1), k(2), k(3), \ldots\}$. Let us first show that we have

$$\lim_{k \to +\infty, k \in \mathcal{S}_n} \max(g_{ti}(x_m^k, x_n^k, \xi_m), 0) = 0. \quad (3.36)$$

For all $\ell \geq 1$, relation (3.12) gives

$$g_{ti}(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) = g_{tij_t(m)}^{k(\ell)}(x_m^{k(\ell)}, x_n^{k(\ell)}). \quad (3.37)$$

Let us now apply Lemma 2.5 to $y^\ell = (x_m^{k(\ell)}, x_n^{k(\ell)})$, sequence $f^\ell = g_{tij_t(m)}^{k(\ell)}$, and $f = g_{ti}(\cdot, \cdot, \xi_m)$ (observe that the assumptions of the lemma are satisfied with $k_0 = 1$). Since

$$\lim_{\ell \to +\infty} f(y^\ell) - f^\ell(y^\ell) = 0,$$

we deduce that

$$\lim_{\ell \to +\infty} f(y^\ell) - f^{\ell-1}(y^\ell) = \lim_{\ell \to +\infty} g_{ti}(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) - g_{tij_t(m)}^{k(\ell-1)}(x_m^{k(\ell)}, x_n^{k(\ell)}) = 0. \tag{3.38}$$

Since $k(\ell) \geq 1 + k(\ell - 1)$, we have $0 \leq g_{ti}(\cdot, \cdot, \xi_m) - g_{tij_t(m)}^{k(\ell)-1}(\cdot, \cdot) \leq g_{ti}(\cdot, \cdot, \xi_m) - g_{tij_t(m)}^{k(\ell-1)}(\cdot, \cdot)$ and therefore (3.38) implies

$$\begin{aligned}\lim_{\ell \to +\infty} & g_{ti}(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) - g_{tij_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) \\ &= \lim_{k \to +\infty, k \in \mathcal{S}_n} g_{ti}(x_m^k, x_n^k, \xi_m) - g_{tij_t(m)}^{k-1}(x_m^k, x_n^k) = 0.\end{aligned} \tag{3.39}$$

Finally, we show in "Appendix" that

$$\lim_{k \to +\infty, k \notin \mathcal{S}_n} g_{ti}(x_m^k, x_n^k) - g_{tij_t(m)}^{k-1}(x_m^k, x_n^k) = 0, \tag{3.40}$$

which achieves the proof of (3.35) and therefore of (3.32).

Let us now show $\mathcal{H}(t)$ by backward induction on $t$. $\mathcal{H}(T+1)$ holds since $\mathcal{Q}_{T+1} = \mathcal{Q}_{T+1}^k$. Assume now that $\mathcal{H}(t+1)$ holds for some $t \in \{2, \ldots, T\}$ and let us show that $\mathcal{H}(t)$ holds. Take a node $n \in \texttt{Nodes}(t-1)$ and let us denote again by $k(1), k(2), \ldots,$ the iterations in $\mathcal{S}_n$ with $k(i) < k(i+1)$: $\mathcal{S}_n = \{k(1), k(2), k(3), \ldots\}$. Let us first show that

$$\lim_{k \to +\infty, k \in \mathcal{S}_n} \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) = \lim_{\ell \to +\infty} \mathcal{Q}_t(x_n^{k(\ell)}) - \mathcal{Q}_t^{k(\ell)}(x_n^{k(\ell)}) = 0. \tag{3.41}$$

By definition of $\mathcal{Q}_t^{k(\ell)}$, we have $\mathcal{Q}_t^{k(\ell)}(x_n^{k(\ell)}) \geq \mathcal{C}_t^{k(\ell)}(x_n^{k(\ell)})$ and therefore for all $\ell \geq 1$ we get:

$$\begin{aligned}0 \leq \mathcal{Q}_t(x_n^{k(\ell)}) - \mathcal{Q}_t^{k(\ell)}(x_n^{k(\ell)}) &\leq \mathcal{Q}_t(x_n^{k(\ell)}) - \mathcal{C}_t^{k(\ell)}(x_n^{k(\ell)}) \\ &= \mathcal{Q}_t(x_n^{k(\ell)}) - \underline{\mathcal{Q}}_t^{k(\ell)-1}(x_n^{k(\ell)}), \\ &= \sum_{m \in C(n_{t-1}^{k(\ell)})} p_m \Big[ \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m) - \underline{\mathfrak{Q}}_{tj_t(m)}^{k(\ell)-1}(x_n^{k(\ell)}) \Big].\end{aligned} \tag{3.42}$$

By definition of $x_m^k$, we have

$$\underline{\mathfrak{Q}}_{tj_t(m)}^{k(\ell)-1}(x_n^{k(\ell)}) = f_{tj_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) + \mathcal{Q}_{t+1}^{k(\ell)-1}(x_m^{k(\ell)}), \tag{3.43}$$

which, plugged into (3.42), gives

$$\begin{aligned}0 &\leq \mathcal{Q}_t(x_n^{k(\ell)}) - \mathcal{Q}_t^{k(\ell)}(x_n^{k(\ell)}) \\ &\leq \sum_{m \in C(n_{t-1}^{k(\ell)})} p_m \Big[ \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m) - f_{tj_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) - \mathcal{Q}_{t+1}^{k(\ell)-1}(x_m^{k(\ell)}) \Big].\end{aligned} \tag{3.44}$$

Let us apply Lemma 2.5 to $y^\ell = (x_m^{k(\ell)}, x_n^{k(\ell)})$, sequence $f^\ell = f_{tj_t(m)}^{k(\ell)}$, and $f = f_t(\cdot, \cdot, \xi_m)$ (observe that the assumptions of the lemma are satisfied). Due to (3.12), we have

$$\lim_{\ell \to +\infty} f(y^\ell) - f^\ell(y^\ell) = 0$$

and therefore

$$\lim_{\ell \to +\infty} f(y^\ell) - f^{\ell-1}(y^\ell) = \lim_{\ell \to +\infty} f_t(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) - f_{tj_t(m)}^{k(\ell-1)}(x_m^{k(\ell)}, x_n^{k(\ell)}) = 0. \tag{3.45}$$

Since $k(\ell) \geq k(\ell-1)+1$, we have $0 \leq f_t(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) - f_{tj_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) \leq f_t(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) - f_{tj_t(m)}^{k(\ell-1)}(x_m^{k(\ell)}, x_n^{k(\ell)})$ which combined with (3.45) gives

$$\lim_{\ell \to +\infty} f_t(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) - f_{tj_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) = 0. \tag{3.46}$$

Using (3.43) and (3.24), we get

$$f_{tj_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) + \mathcal{Q}_{t+1}^{k(\ell)-1}(x_m^{k(\ell)}) = \underline{\mathfrak{Q}}_{tj_t(m)}^{k(\ell)-1}(x_n^{k(\ell)}) \leq \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m).$$

Therefore, the sequence $(f_{tj_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) + \mathcal{Q}_{t+1}^{k(\ell)-1}(x_m^{k(\ell)}) - \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m))_{\ell \geq 1}$ is bounded and has a finite limit superior which satisfies

$$\overline{\lim_{\ell \to +\infty}} f_{tj_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) + \mathcal{Q}_{t+1}^{k(\ell)-1}(x_m^{k(\ell)}) - \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m) \leq 0. \tag{3.47}$$

Applying Lemma 2.5 to $y^\ell = x_m^{k(\ell)}$, sequence $f^\ell = \mathcal{Q}_{t+1}^{k(\ell)}$, and $f = \mathcal{Q}_{t+1}$ (observe that the assumptions of the lemma are satisfied), since from the induction hypothesis we know that

$$\lim_{\ell \to +\infty} f(y^\ell) - f^\ell(y^\ell) = 0$$

we deduce that

$$\lim_{\ell \to +\infty} f(y^\ell) - f^{\ell-1}(y^\ell) = \lim_{\ell \to +\infty} \mathcal{Q}_{t+1}(x_m^{k(\ell)}) - \mathcal{Q}_{t+1}^{k(\ell-1)}(x_m^{k(\ell)}) = 0. \tag{3.48}$$

Since $k(\ell) \geq k(\ell-1)+1$, we have $0 \leq \mathcal{Q}_{t+1}(x_m^{k(\ell)}) - \mathcal{Q}_{t+1}^{k(\ell)-1}(x_m^{k(\ell)}) \leq \mathcal{Q}_{t+1}(x_m^{k(\ell)}) - \mathcal{Q}_{t+1}^{k(\ell-1)}(x_m^{k(\ell)})$, which combines with (3.48) to give

$$\lim_{\ell \to +\infty} \mathcal{Q}_{t+1}(x_m^{k(\ell)}) - \mathcal{Q}_{t+1}^{k(\ell)-1}(x_m^{k(\ell)}) = 0. \tag{3.49}$$

Combining (3.46), (3.47), and (3.49), we obtain

$$\overline{\lim}_{\ell \to +\infty} f_t(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) + \mathcal{Q}_{t+1}(x_m^{k(\ell)}) - \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m) \leq 0. \tag{3.50}$$

Let us now show by contradiction that

$$\underline{\lim}_{k \to +\infty} f_t(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) + \mathcal{Q}_{t+1}(x_n^{k(\ell)}) - \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m) \geq 0. \tag{3.51}$$

Assume that (3.51) does not hold. Using the fact that sequence $(x_m^k, x_n^k)_{k \in \mathcal{S}_n}$ belongs to the compact set $\mathcal{X}_t \times \mathcal{X}_{t-1}$, and the lower semicontinuity of $f_t(\cdot, \cdot, \xi_m)$, $g_t(\cdot, \cdot, \xi_m)$, $\mathcal{Q}_t$, $\mathfrak{Q}_t(\cdot, \xi_m)$, there is a subsequence $(x_m^k, x_n^k)_{k \in K}$ with $K \subset \mathcal{S}_n$ converging to some $(\bar{x}_m, \bar{x}_n) \in \mathcal{X}_t \times \mathcal{X}_{t-1}$ such that

$$f_t(\bar{x}_m, \bar{x}_n, \xi_m) + \mathcal{Q}_{t+1}(\bar{x}_n) - \mathfrak{Q}_t(\bar{x}_n, \xi_m) < 0$$

and $\bar{x}_m \in X_t(\bar{x}_n, \xi_m)$. This is in contradiction with the definition of $\mathfrak{Q}_t$. Therefore, we must have

$$\begin{aligned} 0 &= \lim_{\ell \to +\infty} f_t(x_m^{k(\ell)}, x_n^{k(\ell)}, \xi_m) + \mathcal{Q}_{t+1}(x_m^{k(\ell)}) - \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m) \\ &= \lim_{\ell \to +\infty} f_{tj_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) + \mathcal{Q}_{t+1}^{k(\ell)-1}(x_m^{k(\ell)}) - \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m) \end{aligned}$$

which, plugged into (3.44) gives

$$\lim_{k \to +\infty, k \in \mathcal{S}_n} \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) = 0. \tag{3.52}$$

Finally, we show in "Appendix" that

$$\lim_{k \to +\infty, k \notin \mathcal{S}_n} \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) = 0, \tag{3.53}$$

which achieves the proof of $\mathcal{H}(t)$.

(ii) The proof of (ii) can easily be obtained from (i), see Theorem 4.1-(ii) in [6] for details. □

***Remark 3.10*** (*Stopping criterion*) The stopping criterion is similar to SDDP. We can stop the algorithm when the gap $\frac{Ub-Lb}{Ub}$ is less than a threshold, for instance 5%, where $Ub$ and $Lb$ are upper and lower bounds, respectively, defined as follows. Due to Lemma 3.8, we can take as a lower bound on the optimal value of problem (3.1) the value $Lb = \underline{\mathfrak{Q}}_{11}^{k-1}(x_0)$. The upper bound $Ub$ corresponds to the upper end of a 100(1-$\alpha$)%-one-sided confidence interval (with for instance $\alpha = 0.05$) on the optimal value for $N$ policy realizations (using the costs of decisions taken on $N$ independent sampled scenarios).

### 3.5 Other Variants

It is easy to adapt several recent enhancements of SDDP to the forward StoDCuP method we have just presented. More precisely, we can extend forward StoDCuP to forward–backward StoDCuP which builds the trial points and cuts for the objective and constraint functions corresponding to the sampled scenario in the forward pass and to build cuts for the cost-to-go functions $\mathcal{Q}_t$ in a backward pass. In this case, the backward pass also builds cuts for all functions $f_t(\cdot, \cdot, \xi_{tj})$, $g_{ti}(\cdot, \cdot, \xi_{tj})$ $t = 1, \ldots, T$, $j = 1, \ldots, M_t$, $i = 1, \ldots, p$. It is also easy to incorporate in StoDCuP regularization as in [15], to apply multicut variants as in [3,10], and cut selection strategies for the bundles of cuts of $\mathcal{Q}_t$, for instance along the lines of [7,10,25]. Observe, however, that all linearizations for $f_t(\cdot, \cdot, \xi_{tj})$ and $g_{ti}(\cdot, \cdot, \xi_{tj})$ are tight and therefore no cut selection is needed for these linearizations.

## 4 Inexact Cuts in StoDCuP

In this section, we present an extension of StoDCuP to solve problem (3.1). Since all subproblems of forward StoDCuP presented in Sect. 3 are linear programs, it is easy to derive an inexact variant of StoDCuP that computes $\varepsilon_t^k$-optimal solutions (instead of optimal solutions in StoDCuP) of the subproblems solved for iteration $k$ and stage $t$. We show in Lemma 4.1 that the cuts computed by this variant are still valid and that the distance between the cuts and $\underline{\mathcal{Q}}_t^{k-1}(\cdot) = \sum_{j=1}^{M_t} p_{tj} \underline{\mathfrak{Q}}_{tj}^{k-1}(\cdot)$ at the trial point $x_n^k$ for stage $t$ and iteration $k$ is at most $\varepsilon_t^k$. This variant of StoDCuP, called inexact StoDCuP, is given below and the convergence of the method is proved in Theorem 4.3:

---

**Inexact StoDCuP.**

---

Step 1) **Initialization.** For $t = 1, \ldots, T$, take $f_{tj}^0, g_{tij}^0 : \mathcal{X}_t \times \mathcal{X}_{t-1} \to \mathbb{R}$ affine functions satisfying $f_{tj}^0 \leq f_t(\cdot, \cdot, \xi_{tj})$, $g_{tij}^0 \leq g_{ti}(\cdot, \cdot, \xi_{tj})$, and for $t = 2, \ldots, T$, $\mathcal{Q}_t^0 : \mathcal{X}_{t-1} \to \mathbb{R}$ is an affine function satisfying $\mathcal{Q}_t^0 \leq \mathcal{Q}_t$. Set $x_{n_0} = x_0$, set the iteration count $k$ to 1, and $\mathcal{Q}_{T+1}^0 \equiv 0$.

Step 2) Generate a sample $(\tilde{\xi}_1^k, \tilde{\xi}_2^k, \ldots, \tilde{\xi}_T^k)$ of $(\xi_1, \xi_2, \ldots, \xi_T)$ corresponding to a set of nodes $(n_1^k, n_2^k, \ldots, n_T^k)$ where $n_1^k = n_1$, and for $t \geq 2$, $n_t^k$ is a node of stage $t$, child of node $n_{t-1}^k$. Set $n_0^k = n_0$.
Do $\theta_{T+1}^k = 0$ and $\beta_{T+1}^k = 0$.
**For** $t = 1, \ldots, T$,
Let $n = n_{t-1}^k$.
**For** every $m \in C(n)$,

a) compute an $\varepsilon_t^k$-optimal feasible solution $x_m^k$ of

$$\underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k) = \begin{cases} \min_{x_m} \ f_{tj_t(m)}^{k-1}(x_m, x_n^k) + \mathcal{Q}_{t+1}^{k-1}(x_m) \\ x_m \in X_{tj_t(m)}^{k-1}(x_n^k). \end{cases} \tag{4.1}$$

b) Compute function values and subgradients of convex functions $f_t(\cdot,\cdot,\xi_m)$ and $g_{ti}(\cdot,\cdot,\xi_m)$ at $(x_m^k, x_n^k)$ and let $\ell_{f_t(\cdot,\cdot,\xi_m)}((\cdot,\cdot);(x_m^k,x_n^k))$ and $\ell_{g_{ti}(\cdot,\cdot,\xi_m)}((\cdot,\cdot);(x_m^k,x_n^k))$ denote the corresponding linearizations.

c) Set

$$f_{tj_t(m)}^k = \max\left(f_{tj_t(m)}^{k-1}\,,\ \ell_{f_t(\cdot,\cdot,\xi_m)}((\cdot,\cdot);(x_m^k,x_n^k))\right),$$
$$g_{ti}^k = \max\left(g_{ti}^{k-1}\,,\ \ell_{g_{ti}(\cdot,\cdot,\xi_m)}((\cdot,\cdot);(x_m^k,x_n^k))\right), \quad \forall i=1,\ldots,p.$$

d) Using the notation of Section 3.3, in particular (3.17), (3.18), and (3.18), if $t \geq 2$ compute an $\varepsilon_t^k$-optimal feasible solution $(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k)$ of the dual problem

$$\begin{array}{l}\max\limits_{\alpha,\mu,\delta,\nu,\lambda}\ \alpha^\top(B_{tj_t(m)}^{k-1}x_n^k + C_{tj_t(m)}^{k-1}) + \mu^\top(E_{tj_t(m)}^{k-1}x_n^k + H_{tj_t(m)}^{k-1}) + \delta^\top\theta_{t+1}^{0:k-1}\\ \qquad\qquad +\lambda^\top(b_{tj_t(m)} - B_{tj_t(m)}x_n^k) + \nu^\top\bar{x}_t\\ (A_{tj_t(m)}^{k-1})^\top\alpha + (D_{tj_t(m)}^{k-1})^\top\mu + (\beta_{t+1}^{0:k-1})^\top\delta - \mathbb{X}_t^\top\nu - (A_{tj_t(m)})^\top\lambda = 0,\\ \mathbf{e}^\top\alpha = 1,\ \mathbf{e}^\top\delta = 1, \alpha,\mu,\delta,\nu \geq 0.\end{array}$$

**End For**

**If** $t \geq 2$ compute:

$$\begin{array}{l}\beta_t^k = \sum\limits_{m\in C(n)} p_m\Big[(B_{tj_t(m)}^k)^\top\alpha_m^{k-1} + (E_{tj_t(m)}^{k-1})^\top\mu_m^k - B_{tj_t(m)}^\top\lambda_m^k\Big],\\ \theta_t^k = \sum\limits_{m\in C(n)} p_m\Big[\langle\alpha_m^k, C_{tj_t(m)}^{k-1}\rangle + \langle\mu_m^k, H_{tj_t(m)}^{k-1}\rangle + \langle\delta_m^k, \theta_{t+1}^{0:k-1}\rangle + \langle\lambda_m^k, b_{tj_t(m)}\rangle + \langle\nu_m^k, \bar{x}_t\rangle\Big].\end{array} \tag{4.2}$$

**End If**

**End For**

Step 4) Do $k \leftarrow k+1$ and go to Step 2).

---

Clearly Lemma 3.7 still holds for Inexact StoDCuP. The quality of the cuts computed for $\mathcal{Q}_t$ by Inexact StoDCuP is given in Lemma 4.1:

**Lemma 4.1** (Validity and quality of cuts computed by Inexact StoDCuP) *Let Assumptions (H0) and (H1)-Sto hold. For every $t = 2,\ldots,T+1$, for every $k \geq 1$, we have*

$$\mathcal{Q}_t(x_{t-1}) \geq \mathcal{C}_t^k(x_{t-1}) \text{ and } \mathcal{Q}_t(x_{t-1}) \geq \mathcal{Q}_t^k(x_{t-1}),\ \forall x_{t-1}\in\mathcal{X}_{t-1}. \tag{4.3}$$

*For all $t=1,\ldots,T$, $j=1,\ldots,M_t$, for every $k\geq 1$, we have*

$$\underline{\mathfrak{Q}}_{tj}^k(x_{t-1}) \leq \mathfrak{Q}_t(x_{t-1},\xi_{tj}) \text{ for all } x_{t-1}\in\mathcal{X}_{t-1}. \tag{4.4}$$

*For all $t = 2, \ldots, T$, for every $k \geq 1$, defining $\underline{\mathcal{Q}}_t^{k-1}(x_n^k) = \sum_{j=1}^{M_t} p_{tj} \underline{\mathfrak{Q}}_{tj}^{k-1}(x_n^k)$, we have for every $n \in \mathtt{Nodes}(t-1)$ and for all $k \in \mathcal{S}_n$:*

$$0 \leq \underline{\mathcal{Q}}_t^{k-1}(x_n^k) - \mathcal{C}_t^k(x_n^k) \leq \varepsilon_t^k. \tag{4.5}$$

***Proof*** The proofs of (3.23) and (3.24) in Lemma 3.8 can be used to prove (4.3) and (4.4) for Inexact StoDCuP, observing that only feasibility and not optimality of the primal and dual solutions computed as well as Lemma 3.7 (which, as we have already observed, holds) are needed in these proofs.

Now take $n \in \mathtt{Nodes}(t-1)$ and $k \in \mathcal{S}_n$. Then recalling that

$$\begin{aligned}\mathcal{D}_{tj}^{k-1}(\alpha, \mu, \delta, \nu, \lambda; x_{t-1}) = \alpha^\top (B_{tj}^{k-1} x_{t-1} + C_{tj}^{k-1}) + \mu^\top (E_{tj}^{k-1} x_{t-1} + H_{tj}^{k-1}) \\ + \delta^\top \theta_{t+1}^{0:k-1} + \lambda^\top (b_{tj} - B_{tj} x_{t-1}) + \nu^\top \bar{x}_t,\end{aligned}$$

by definition of $(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k)$ and of $\mathcal{C}_t^k$, we get

$$\underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k) - \varepsilon_t^k \leq \mathcal{D}_{tj_t(m)}^{k-1}(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k; x_n^k) \leq \underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k) \tag{4.6}$$

and

$$\mathcal{C}_t^k(x_n^k) = \sum_{m \in C(n)} p_m \mathcal{D}_{tj_t(m)}^{k-1}(\alpha_m^k, \mu_m^k, \delta_m^k, \nu_m^k, \lambda_m^k; x_n^k). \tag{4.7}$$

Since $\underline{\mathcal{Q}}_t^{k-1}(x_n^k) = \sum_{m \in C(n_{t-1}^k)} p_m \underline{\mathfrak{Q}}_{tj_t(m)}^{k-1}(x_n^k)$, $p_m \geq 0$, and $\sum_{m \in C(n)} p_m = 1$, relations (4.6) and (4.7) imply (4.5). □

Lemma 4.2 is the analogue of Lemma 3.4:

**Lemma 4.2** *Let Assumptions (H0) and (H1)-Sto hold and assume that sequences $\varepsilon_t^k$ are bounded: $|\varepsilon_t^k| \leq \hat{\varepsilon}$ for all $t, k$, for some $0 \leq \hat{\varepsilon} < +\infty$. Then, the following statements hold for Inexact StoDCuP:*

(a) *For $t = 2, \ldots, T$, the sequences $\{\theta_t^k\}_{k=1}^\infty$ and $\{\beta_t^k\}_{k=1}^\infty$ are almost surely bounded.*
(b) *There exists $L \geq 0$ such that for each $t = 2, \ldots, T$, $\mathcal{Q}_t^k$ is $L$-Lipschitz continuous on $\mathcal{X}_{t-1}$ for every $k \geq 1$.*
(c) *There exists $\hat{L} \geq 0$ such that for each $t = 1, \ldots, T$, $j = 1, \ldots, M_t$, functions $f_{tj}^k$ and $g_{tij}^k$ are $\hat{L}$-Lipschitz continuous on $\mathcal{X}_t \times \mathcal{X}_{t-1}$ for every $k \geq 1$ and $i = 1, \ldots, p$.*

***Proof*** (a) Using (H1)-Sto, there is $\varepsilon > 0$ such that for every $t \in \{2, \ldots, T\}$, every $x_{t-1} \in \mathcal{X}_{t-1} + \bar{B}(0; \varepsilon)$, and every $j = 1, \ldots, M_t$, the set $X_{tj}^0(x_{t-1})$ is nonempty and $f_{tj}^0(\cdot, x_{t-1}) + \mathcal{Q}_{t+1}^0(\cdot)$ is continuous on this set. Therefore, $\underline{\mathfrak{Q}}_{tj}^0$ is convex and finite on $\mathcal{X}_{t-1} + \bar{B}(0; \varepsilon)$, implying that $\underline{\mathfrak{Q}}_{tj}^0$ is Lipschitz continuous on $\mathcal{X}_{t-1}$. It follows that $\underline{\mathcal{Q}}_t^0$ is also Lipschitz continuous on $\mathcal{X}_{t-1}$ and we can define $\min_{x_{t-1} \in \mathcal{X}_{t-1}} \underline{\mathcal{Q}}_t^0(x_{t-1}) \in \mathbb{R}$. Similarly to DCuP, due to (H1)-Sto, we can also choose

$\varepsilon > 0$ in such a way that $\mathcal{Q}_t$ is Lipschitz continuous on $\mathcal{X}_{t-1} + \bar{B}(0;\varepsilon)$, implying that we can define $\max_{x_{t-1}\in\mathcal{X}_{t-1}+\bar{B}(0;\varepsilon)} \mathcal{Q}_t(x_{t-1}) < +\infty$. We can now easily extend the proof of Lemma 3.4: for every $x_{t-1} \in \mathcal{X}_{t-1} + \bar{B}(0;\varepsilon)$, denoting $n = n^k_{t-1}$, we have for $k \geq 2$:

$$\begin{aligned}\max_{x_{t-1}\in\mathcal{X}_{t-1}+\bar{B}(0;\varepsilon)} \mathcal{Q}_t(x_{t-1}) \geq \mathcal{Q}_t(x_{t-1}) &\overset{(4.3)}{\geq} \mathcal{C}^k_t(x_{t-1}) \\ &= \mathcal{C}^k_t(x^k_n) + \langle \beta^k_t, x_{t-1} - x^k_n \rangle \quad [\mathcal{C}^k_t \text{ is affine}], \\ &\overset{(4.5)}{\geq} \underline{\mathcal{Q}}^{k-1}_t(x^k_n) - \varepsilon^k_t + \langle \beta^k_t, x_{t-1} - x^k_n \rangle, \\ &\geq \min_{x_{t-1}\in\mathcal{X}_{t-1}} \underline{\mathcal{Q}}^0_t(x_{t-1}) - \hat{\varepsilon} + \langle \beta^k_t, x_{t-1} - x^k_n \rangle.\end{aligned}$$

For $\beta^k_t \neq 0$, take $x_{t-1} = x^k_n + \frac{\varepsilon}{2}\frac{\beta^k_t}{\|\beta^k_t\|}$ to obtain

$$\|\beta^k_t\| \leq L := \frac{2}{\varepsilon}\left(\hat{\varepsilon} + \max_{x_{t-1}\in\mathcal{X}_{t-1}+\bar{B}(0;\varepsilon)} \mathcal{Q}_t(x_{t-1}) - \min_{x_{t-1}\in\mathcal{X}_{t-1}} \underline{\mathcal{Q}}^0_t(x_{t-1})\right).$$

Using (4.5), we also have for $n = n^k_{t-1}$:

$$-\hat{\varepsilon} + \min_{x_{t-1}\in\mathcal{X}_{t-1}} \underline{\mathcal{Q}}^0_t(x_{t-1}) \leq \theta^k_t = \mathcal{C}^k_t(x^k_n) \leq \max_{x_{t-1}\in\mathcal{X}_{t-1}} \mathcal{Q}_t(x_{t-1}).$$

(b) immediately follows from (a) and (c) from (H1)-Sto. □

**Theorem 4.3** (Convergence of Inexact StoDCuP) *Let Assumptions (H0), (H1)-Sto, and (H2) hold and assume that* $\lim_{k\to+\infty} \varepsilon^k_t = 0$ *for* $t = 1,\ldots,T$. *Then the conclusions of Theorem 3.9 hold: for every* $t = 1,\ldots,T, i = 1,\ldots,p$, *almost surely* (3.32) *and* (3.33) *hold and the limit of the sequence of first stage problems optimal values* $(f^{k-1}_{11}(x^k_{n_1}, x_0) + \mathcal{Q}^{k-1}_2(x^k_{n_1}))_{k\geq 1}$ *is the optimal value* $\mathcal{Q}_1(x_0)$ *of* (3.2) *and any accumulation point of the sequence* $(x^k_{n_1})$ *is an optimal solution to the first stage problem* (3.2).

***Proof*** The proof is an adaptation of the proof of Theorem 3.9 and uses Lemmas 3.7, 4.1, and 4.2. We highlight these adaptations below.

Using Lemma 4.1, for Inexact StoDCuP relation (3.42) becomes

$$\begin{aligned}0 \leq \mathcal{Q}_t(x^{k(\ell)}_n) - \mathcal{Q}^{k(\ell)}_t(x^{k(\ell)}_n) \leq \mathcal{Q}_t(x^{k(\ell)}_n) - \mathcal{C}^{k(\ell)}_t(x^{k(\ell)}_n) \\ &\overset{(3.25)}{\leq} \varepsilon^{k(\ell)}_t + \mathcal{Q}_t(x^{k(\ell)}_n) - \underline{\mathcal{Q}}^{k(\ell)-1}_t(x^{k(\ell)}_n), \\ &= \varepsilon^{k(\ell)}_t + \sum_{m\in C(n^{k(\ell)}_{t-1})} p_m\Big[\mathfrak{Q}_t(x^{k(\ell)}_n, \xi_m) - \underline{\mathfrak{Q}}^{k(\ell)-1}_{tj_t(m)}(x^{k(\ell)}_n)\Big].\end{aligned} \tag{4.8}$$

Also, by definition of $x^k_m$, we now have

$$\underline{\mathfrak{Q}}^{k(\ell)-1}_{tj_t(m)}(x^{k(\ell)}_n) \leq f^{k(\ell)-1}_{tj_t(m)}(x^{k(\ell)}_m, x^{k(\ell)}_n) + \mathcal{Q}^{k(\ell)-1}_{t+1}(x^{k(\ell)}_m) \leq \underline{\mathfrak{Q}}^{k(\ell)-1}_{tj_t(m)}(x^{k(\ell)}_n) + \varepsilon^{k(\ell)}_t, \tag{4.9}$$

**Table 1** Number of variables and constraints of the deterministic equivalents of the six instances

| Instance $T, n, M$ | Variables $(n+2)(1+M^{T-1})$ | Linear constraints $(2n+1)(1+M^{T-1})$ | Quadratic constraints $4(1+M^{T-1})$ |
|---|---|---|---|
| 3, 10, 2 | 60 | 105 | 20 |
| 5, 10, 10 | 120,012 | 210,021 | 40,004 |
| 5, 10, 20 | 1.92e6 | 3.36e6 | 6.4e5 |
| 10, 200, 10 | 2.02e11 | 4.01e11 | 4e9 |
| 10, 200, 20 | 1.0342e14 | 2.0531e14 | 2.0480e12 |

**Table 2** Relative error of the subproblem solutions in IStoDCuP along iterations (Mosek parameter MSK_DPAR_INTPNT_TOL_REL_GAP)

| Iteration | 1–10 | 11–20 | 21–40 | 41–140 | 141–240 | 241–350 | $> 350$ |
|---|---|---|---|---|---|---|---|
| MSK_DPAR_INTPNT_TOL_REL_GAP | 10 | 5 | 3 | 1 | 0.5 | 0.1 | e-6 |

which, plugged into (4.8) gives

$$\begin{aligned} 0 &\leq \mathcal{Q}_t(x_n^{k(\ell)}) - \mathcal{Q}_t^{k(\ell)}(x_n^{k(\ell)}) \\ &\leq 2\varepsilon_t^{k(\ell)} + \sum_{m \in C(n_{t-1}^{k(\ell)})} p_m \Big[ \mathfrak{Q}_t(x_n^{k(\ell)}, \xi_m) - f_{tj_t(m)}^{k(\ell)-1}(x_m^{k(\ell)}, x_n^{k(\ell)}) - \mathcal{Q}_{t+1}^{k(\ell)-1}(x_m^{k(\ell)}) \Big]. \end{aligned} \tag{4.10}$$

The remaining relations and arguments used in the convergence proof of StoDCuP apply to prove the theorem. □

## 5 Numerical Experiments

We consider the multistage nondifferentiable nonlinear stochastic program given by the following DP equations: the Bellman function for stage $t = 1, \ldots, T$, is $\mathcal{Q}_t(x_{t-1}) = \mathbb{E}_{\xi_t, \Psi_t, U_t}[\mathfrak{Q}_t(x_{t-1}, \xi_t, \Psi_t, U_t)]$ and for $t = 1, \ldots, T$, $\mathfrak{Q}_t(x_{t-1}, \xi_t, \Psi_t, U_t)$ is given by

$$\begin{array}{l} \min \ f_t(x_t, x_{t-1}, \xi_t, U_t) + \mathcal{Q}_{t+1}(x_t) \\ -100\,\mathbf{e} \leq x_t \leq 100\,\mathbf{e}, \\ \max(4(x_t - \mathbf{e})^T(x_t - \mathbf{e}), x_t^T \xi_t \xi_t^T x_t + x_t^T \xi_t + 1) \leq \Psi_t, \end{array} \tag{5.1}$$

where $x_t \in \mathbb{R}^n$, $f_t(x_t, x_{t-1}, \xi_t, U_t) = \max((x_t - x_{t-1})^T \xi_t \xi_t^T (x_t - x_{t-1}) + x_t^T \xi_t + 1, x_t^T \xi_t \xi_t^T x_t + x_t^T \mathbf{e} + U_t)$, $\mathbf{e}$ is a vector of size $n$ of ones, and $\mathcal{Q}_{T+1}$ is the null function. In these equations, $\xi_t$ is a discretization of a Gaussian random vector with mean vector $m_t$ having entries 1 or $-1$ and covariance matrix $\Sigma_t = A_t A_t^T + 0.5I$ where $A_t$ has entries in $[-0.5, 0.5]$; $U_t$ is a discrete random variable taking values $+10$, $-10$, and $\Psi_t$ has discrete distribution with support contained in $[10^4, 10^5]$. The number of realizations

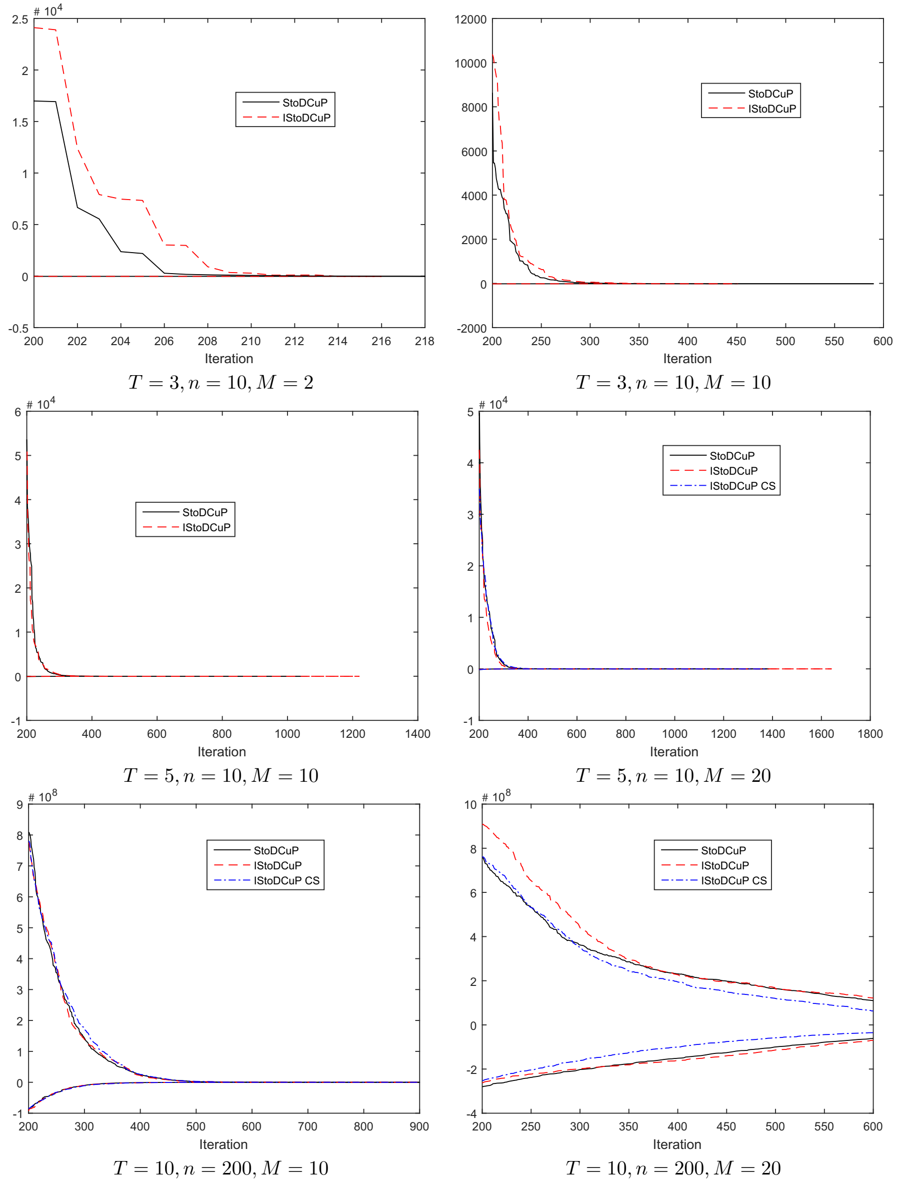


**Fig. 2** Upper and lower bounds computed by StoCuP, IStoDCuP, and IStoDCuP CS along the iterations to solve the instances

$M_t$ for $(\xi_t, \Psi_t, U_t)$ is fixed to $M_t = M$ for each stage. We assume that $(\xi_1, \Psi_1, U_1)$ is known and $(\xi_2, \Psi_2, U_2), \ldots, (\xi_T, \Psi_T, U_T)$ are independent.

We generate six instances of this problem with parameters $T, n, M$ given by $(T, n, M) = (3, 10, 2)$, $(3, 10, 10)$, $(5, 10, 10)$, $(5, 10, 20)$, $(10, 200, 10)$, and $(10, 200, 20)$. The instances are chosen taking realizations $\Psi_{tj}$ of $\Psi_t$ sufficiently large,

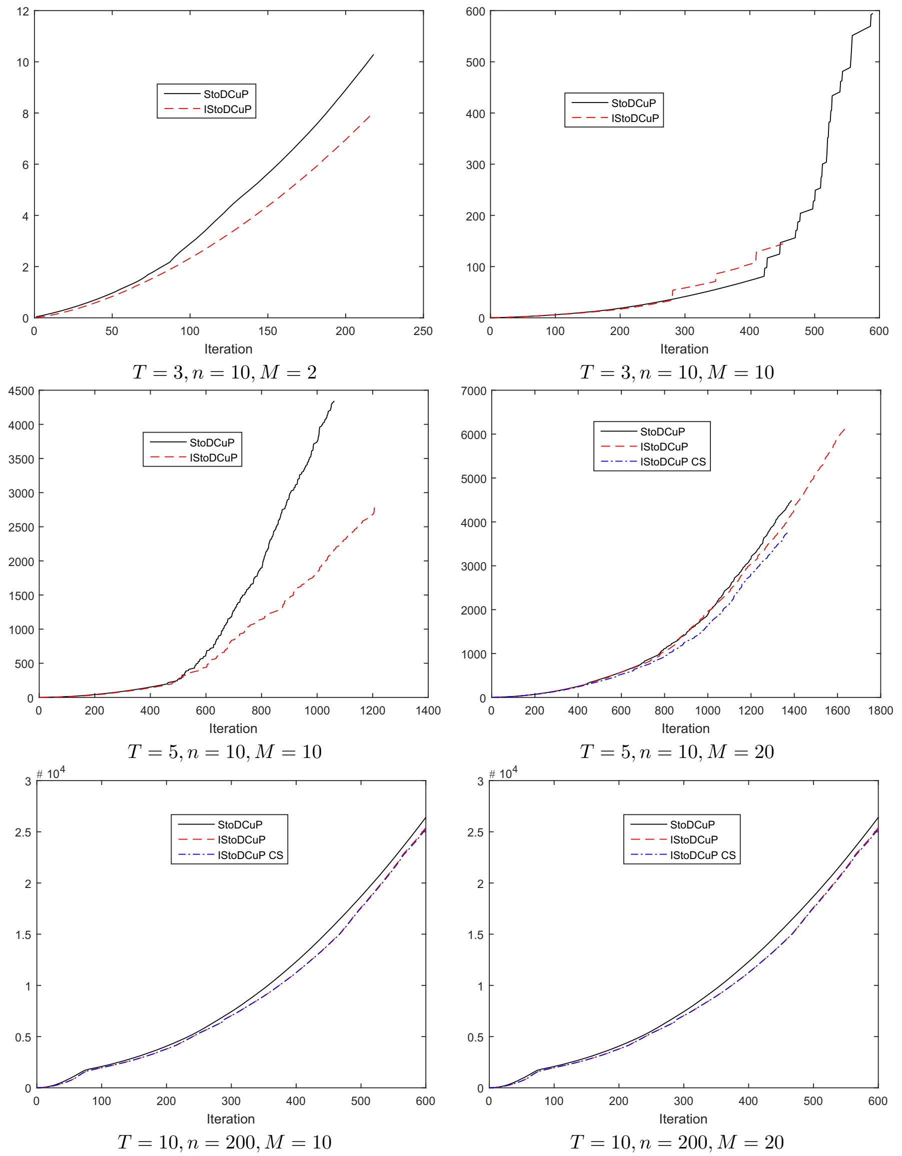


**Fig. 3** Cumulated CPU time in seconds along the iterations of StoCuP, IStoDCuP, and IStoDCuP CS to solve the instances

in such a way that Assumption (H1)-Sto-4) holds.[2] It is easy to check that the remaining assumptions (H1)-Sto and (H0) are satisfied and therefore StoDCuP and Inexact StoDCuP (IStoDCuP) can both be applied to solve the problem. Since the problem is nondifferentiable, SDDP and Inexact SDDP from [8] cannot be applied directly.

[2] We checked that the instances generate nontrivial nondifferentiable problems in the sense that no function in the max dominates the other on the set $\mathcal{X}_t := \{x_t \in \mathbb{R}^n : -100\,\mathbf{e} \le x_t \le 100\,\mathbf{e}\}$.

**Table 3** Number of iterations and CPU time (in seconds) for each instance and method. For Inexact StoDCuP, we report the quickest, among IStoDCuP and IStoDCuP CS

| $(T, n, M)$ | Iterations StoDCuP | Iterations Inexact StoDCuP | CPU time StoDCuP | CPU Time Inexact StoDCuP |
|---|---|---|---|---|
| 3, 10, 2 | 216 | 216 | 10.13 | 7.63 |
| 3, 10, 10 | 586 | 451 | 597.2 | 148.4 |
| 5, 10, 10 | 1061 | 1221 | 4345 | 2825 |
| 5, 10, 20 | 1387 | 1376 | 4493 | 3784 |
| 10, 200, 10 | 900 | 900 | 62,536 | 55,061 |
| 10, 200, 20 | 600 | 600 | 26,414 | 25,276 |

However, it is possible to reformulate the problem as a differentiable problem replacing in (5.1) each max with 2 quadratic constraints. The number of variables and of linear and quadratic constraints of the deterministic equivalent corresponding to this reformulation is given in Table 1 for all instances.

Using this reformulation, we implemented ISDDP given in [8] and SDDP, using Mosek [1] to solve the subproblems. Unfortunately, none of the six instances could be solved by these implementations because essentially all suproblems to be solved within SDDP and ISDDP cannot be solved by Mosek due to the fact that all the matrices of the quadratic forms are ill-conditioned, yielding an error in the convexity check performed by Mosek (even if of course in theory all subproblems are convex) which is done using Cholesky factorizations of those matrices. Rather than a flaw of Mosek which is an efficient solver for conic problems, the problem comes from the subproblems under consideration which are difficult to solve because of the degeneracy of the quadratic forms.[3] In this condition, StoDCup and IStoDCuP (considering the variants which linearize all nonlinear functions at all iterations for all subproblems) which only have to solve linear subproblems are possible solution methods to solve the original problem. The corresponding MATLAB implementation can be found at https://github.com/vguigues/StoDCuP.[4] Both StoDCuP and IStoDCuP were warm-started constructing 20 linearizations of each function $f_t(\cdot, \cdot, \xi_{tj})$ and $g_{ti}(\cdot, \cdot, \xi_{tj})$ at points randomly selected in the set $\mathcal{X}_t := \{x_t \in \mathbb{R}^n : -100\,\mathbf{e} \leq x_t \leq 100\,\mathbf{e}\}$.

For IStoDCuP to be well defined, we also need to set the level of accuracy of the computed solutions along the iterations of the method. It makes sense to increase the accuracy (or equivalently to decrease the relative error) of the solutions as the algorithm progresses and eventually for a given iteration to increase the accuracy with the stage. In our experiments the relative error of the subproblem solutions (Mosek parameter MSK_DPAR_INTPNT_TOL_REL_GAP whose range is any value $\geq 10^{-14}$ and

[3] We also implemented ISDDP using the inexact cuts from Section 2 of [11] and such variant could not solve our instances neither, again because Mosek failed to solve all quadratic subproblems of the corresponding ISDDP.

[4] The tests were run in file `TestStoDCuP.m` and the functions implementing StoDCup and IStoDCuP are `inexact_stodcup_quadratic.m` and `inexact_stodcup_quadratic_cut_selection.m`, this latter being a variant with cut selection, denoted IStoDCuP CS in this section.

**Table 4** Cumulated CPU time (Time) in seconds and upper (UB) and lower (LB) bounds computed by StoDCuP, IStoDCuP, and IStoDCuP CS for some iterations and the six instances

| Iteration | UB IStoDCuP | UB StoDCuP | LB IStoDCuP | LB StoDCuP | Time IStoDCuP | Time StoDCuP |
|---|---|---|---|---|---|---|
| $T = 3, n = 10, M = 2$ | | | | | | |
| 10 | – | – | – 36,424 | – 15,037 | 0.09 | 0.16 |
| 200 | 24 109 | 16 990 | – 29.3120 | – 29.3123 | 6.98 | 8.91 |
| 210 | 282.4 | 73.9 | – 29.3120 | – 29.3123 | 7.57 | 9.67 |
| 216 | – 26.99 | – 27.34 | – 29.3120 | – 29.3123 | 7.63 | 10.13 |
| $T = 3, n = 10, M = 10$ | | | | | | |
| 10 | - | – | – 98,584 | – 82,872 | 0.29 | 0.35 |
| 210 | 6361.8 | 3889.3 | – 19.023 | – 17.44 | 20.6 | 18.5 |
| 451 | – 14.97 | – 13.58 | – 16.27 | – 16.25 | 148.4 | 144.6 |
| $T = 5, n = 10, M = 10$ | | | | | | |
| 10 | - | - | – 440,000 | – 352,310 | 0.65 | 0.72 |
| 400 | 9.68 | 13.84 | – 12.09 | – 12.58 | 140.9 | 151.7 |
| 800 | – 6.01 | – 8.69 | – 10.83 | – 10.85 | 1144.6 | 1897.2 |

default value is $10^{-8}$) is given in Table 2; see also Remark 2 in [8] for other choices of sequences of noises $\varepsilon_t^k$. For StoDCuP, this parameter was set to $10^{-10}$ for all iterations.

Same as SDDP, methods StoDCuP and IStoDCuP compute at each iteration a lower bound on the optimal value which is the optimal value of the first stage problem solved in the forward pass and upper bounds computed as SDDP by Monte–Carlo simulations, from iterations 200 on, using the last 200 forward scenarios. We also run the methods with the smoothed upper bounds used in [4,14] which consists in using all previous forward passes to compute the upper bound but this implementation needed many more iterations to satisfy the stopping criterion for the large instances and the corresponding results will not be reported. We should also recall (see [8]) that for both IStoDCuP and StoDCuP the first stage problems are solved with high accuracy to get valid lower bounds from the optimal values of the first stage forward subproblems. The algorithms stopped when a relative gap of at most 0.1 was achieved for the first four instances while for the last two instances, the algorithms were run for 900 and 600 iterations, respectively.

As mentioned in Sect. 3.5, the cut selection methods proposed in [7,10,25] for SDDP can be directly applied to StoDCuP. The convergence of DDP, single cut SDDP, and multicut SDDP combined with these cut selection methods was proved in [7, 10]. For the three largest instances, we tested another cut selection strategy for the inexact variant IStoDCuP of StoDCuP, denoted by IStoDCuP CS, which consists, in the backward passes, from a given iteration $I$ and for the next $L - 1$ iterations, to simultaneously add a new cut (computed at the trial points computed in the forward pass) for each cost-to-go function and to eliminate the oldest cut. As long as $L$ is not too large, we only eliminate, progressively, the cuts computed with loose accuracy (the cuts computed for the first $L$ iterations). Therefore, with this method, in the end of iterations $I, I + 1, \ldots, I + L - 1$, the number of cuts for each cost-to-go

**Table 4** continued

| Iteration | UB IStoDCuP | UB StoDCuP | LB IStoDCuP | LB StoDCuP | Time IStoDCuP | Time StoDCuP |
|---|---|---|---|---|---|---|
| 1061 | – 7.84 | – 9.78 | – 10.72 | – 10.72 | 2166 | 4345 |
| 1221 | – 9.78 | – | – 10.69 | – | 2825 | – |

| Iteration | 600 | 1376 | 1387 | 1642 |
|---|---|---|---|---|
| $T = 5, n = 10, M = 20$ | | | | |
| UB IStoDCuP CS | – 0.586 | – 4.5343 | – | – |
| UB IStoDCuP | – 1.6317 | – 3.7448 | – 3.7993 | – 4.5153 |
| UB StoDCuP | – 0.0327 | – 3.9773 | – 4.5648 | – |
| LB IStoDCuP CS | – 5.5886 | – 4.9595 | – | – |
| LB IStoDCuP | – 5.6431 | – 4.9584 | – 4.9552 | – 4.9078 |
| LB StoDCuP | – 5.7420 | – 4.9623 | – 4.9591 | – |
| Time IStoDCuP CS | 525 | 3784 | – | – |
| Time IStoDCuP | 575 | 4106 | 4180 | 6178 |
| Time StoDCuP | 579 | 4424 | 4493 | – |

| Iteration | 400 | 600 | 900 |
|---|---|---|---|
| $T = 10, n = 200, M = 10$ | | | |
| UB IStoDCuP CS | 2.5343e7 | 3.6465e5 | 143.2 |
| UB IStoDCuP | 2.1689e7 | 4.6095e5 | 338.7 |
| UB StoDCuP | 2.3785e7 | 3.0292e5 | – 50.4 |
| LB IStoDCuP CS | – 1.3343e6 | – 4.0214e4 | – 444.4 |
| LB IStoDCuP | – 1.4643e6 | – 5.0292e4 | – 436.4 |
| LB StoDCuP | – 0.9529e6 | – 2.0954e4 | – 428.9 |
| Time IStoDCuP CS | 8 534.6 | 21,166 | 56,082 |

function is constant, equal to $I$, and then from iteration $I + L$ on, one cut is added for each cost-to-go function at each iteration as in IStoDCuP if we choose one sampled scenario per forward pass. In our experiments, this cut selection strategy was run taking $I = L = 350$.

The evolution of the upper and lower bounds along the iterations of StoDCuP, IStoDCup, and IStoDCuP CS to solve the six instances is given in Fig. 2 while the cumulated CPU time is given in Fig. 3. All methods were implemented in MATLAB and run on an Intel Core i7, 1.8GHz, processor with 12,0 Go of RAM. More precisely, the number of iterations and CPU time required to solve all instances is given in Table 3 and the bounds and cumulated CPU time for some iterations are given in Table 4.

We observe that the sequences of upper bounds tend to decrease, the sequences of lower bounds are increasing, and all these sequences converge to the same values for a given instance, which illustrates the validity of StoDCuP and IStoDCuP to solve a multistage stochastic nondifferentiable convex problem and is a good indication that both methods have been well implemented.

**Table 4** continued

| Iteration | 400 | 600 | 900 |
|---|---|---|---|
| Time IStoDCuP | 7 946.7 | 21,557 | 55,061 |
| Time StoDCuP | 8 364.2 | 24,015 | 62,536 |

| Iteration | 400 | 500 | 600 |
|---|---|---|---|
| $T = 10, n = 200, M = 20$ | | | |
| UB IStoDCuP CS | 1.943e8 | 1.1955e8 | 0.6321e8 |
| UB IStoDCuP | 2.2722e8 | 1.6320e8 | 1.2129e8 |
| UB StoDCuP | 2.3129e8 | 1.6563e8 | 1.0990e8 |
| LB IStoDCuP CS | − 1.0060e8 | − 0.5826e8 | − 0.3522e8 |
| LB IStoDCuP | − 1.6151e8 | − 1.1376e8 | − 0.6979e8 |
| LB StoDCuP | − 1.5124e8 | − 0.9974e8 | − 0.6059e8 |
| Time IStoDCuP CS | 1.1254e4 | 1.7554e4 | 2.5276e4 |
| Time IStoDCuP | 1.1254e4 | 1.7618e4 | 2.5418e4 |
| Time StoDCuP | 1.2320e4 | 1.8689e4 | 2.6414e4 |

In all instances, at least one of the inexact variants of StoDCuP was quicker than StoDCuP and provided policies of similar quality. A general behavior we expect for IStoDCuP is to have quicker iterations but to need more iterations, as for instance $T = 5, n = 10, M = 10$, or a similar number of iterations, as for instances $T = 3, n = 10, M = 2$ and $T = 5, n = 10, M = 20$. (In this latter the number of iterations before getting a gap smaller than 0.1 is 1376, 1387, and 1642 for, respectively, IStoDCuP CS, StoDCuP, and IStoDCuP (see Table 4).) However, it may happen that StoDCuP requires more iterations as for instance $T = 3, n = 10, M = 10$. The inexact variant with cut selection tested on the three largest instances allowed us to decrease the gap with respect to IStoDCuP while still being quicker than StoDCuP. It is also interesting to see that on the largest instance this inexact variant also yielded a much smaller gap than StoDCuP after completing the 600 iterations (see Table 4, Fig. 2).

## 6 Conclusion

We introduced StoDCuP, a variant of SDDP which builds linearizations of some or all nonlinear constraint and objective functions along the iterations of the method, as well as an inexact variant of StoDCuP which is able to cope with approximate primal-dual solutions of the subproblems solved along the iterations. We have shown the convergence of StoDCuP and of Inexact StoDCuP for vanishing error terms $\varepsilon_t^k$.

Our numerical experiments have illustrated on a difficult nonlinear nondifferentiable multistage stochastic program that StoDCuP can be an alternative solution method to SDDP and that its inexact variant can converge quicker than StoDCuP. An interesting feature of the inexact variant is its flexibility, able to cope with any approximate primal-dual solution to the subproblems, allowing to further study the impact of the calibration of error terms $\varepsilon_t^k$ on the performance of Inexact StoDCuP. For DCuP, the calibration seems simpler, see for instance Remark 2 in [8] on the calibration of the error terms for Inexact DDP which also applies to Inexact DCuP.

**Acknowledgements** The research of the first author was partially supported by an FGV grant, CNPq grants 401371/2014-0, 311289/2016-9, 204872/2018-9, and FAPERJ grant E-26/201.599/2014. Research of the second author was partially supported by CNPq grant 401371/2014-0.

**Data Availability** All (simulated) data generated or analyzed during this study can be obtained following the steps given in Sect. 5 of this article.

## Appendix

To prove (3.40) and (3.53), we will need the following lemma (the proof of (ii) of this lemma was given in [5] for a more general sampling scheme and the proof of (i), that we detail, is similar to the proof of (ii)):

**Lemma A.1** *Assume that Assumptions (H0), (H1)-Sto, and (H2) hold for StoDCuP. Define random variables* $y_n^k = 1(k \in \mathcal{S}_n)$.

(i) *Let* $\varepsilon > 0$, $t \in \{1, \ldots, T\}$, $n \in \texttt{Nodes}(t-1)$, $m \in C(n)$, $i \in \{1, \ldots, p\}$ *and set*

$$K_{\varepsilon,m,i} = \left\{k \geq 1 : g_{ti}(x_m^k, x_n^k, \xi_m) - g_{tij_t(m)}^{k-1}(x_m^k, x_n^k) \geq \varepsilon \right\}.$$

*Let*

$$\Omega_0(\varepsilon) = \{\omega \in \Omega : |K_{\varepsilon,m,i}(\omega)| \textit{ is infinite}\}$$

*and assume that* $\Omega_0(\varepsilon) \neq \emptyset$. *Define on the sample space* $\Omega_0(\varepsilon)$ *the random variables* $\mathcal{I}_{\varepsilon,m,i}(j)$, $j \geq 1$, *where* $\mathcal{I}_{\varepsilon,m,i}(1) = \min\{k \geq 1 : k \in K_{\varepsilon,m,i}(\omega)\}$ *and for* $j \geq 2$

$$\mathcal{I}_{\varepsilon,m,i}(j) = \min\{k > \mathcal{I}_{\varepsilon,m,i}(j-1) : k \in K_{\varepsilon,m,i}(\omega)\},$$

*i.e.,* $\mathcal{I}_{\varepsilon,m,i}(j)(\omega)$ *is the index of* $j$*th iteration* $k$ *such that* $g_{ti}(x_m^k, x_n^k, \xi_m) - g_{tij_t(m)}^{k-1}(x_m^k, x_n^k) \geq \varepsilon$. *Then random variables* $(y_n^{\mathcal{I}_{\varepsilon,m,i}(j)})_{j\geq 1}$ *defined on sample space* $\Omega_0(\varepsilon)$ *are independent, have the distribution of* $y_n^1$ *and therefore by the Strong Law of Large numbers we have*

$$\mathbb{P}\left(\lim_{N\to+\infty} \frac{1}{N} \sum_{j=1}^{N} y_n^{\mathcal{I}_{\varepsilon,m,i}(j)} = \mathbb{E}[y_n^1]\right) = 1. \tag{A.1}$$

(ii) *Let* $\varepsilon > 0$, $t \in \{1, \ldots, T\}$, $n \in \texttt{Nodes}(t-1)$, *and set*

$$K_{\varepsilon,n} = \left\{k \geq 1 : \mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) \geq \varepsilon \right\}.$$

*Let*

$$\Omega_1(\varepsilon) = \{\omega \in \Omega : |K_{\varepsilon,n}(\omega)| \text{ is infinite}\}$$

*and assume that* $\Omega_1(\varepsilon) \neq \emptyset$. *Define on the sample space* $\Omega_1(\varepsilon)$ *the random variables* $\mathcal{I}_{\varepsilon,n}(j)$, $j \geq 1$, *where* $\mathcal{I}_{\varepsilon,n}(1) = \min\{k \geq 1 : k \in K_{\varepsilon,n}(\omega)\}$ *and for* $j \geq 2$

$$\mathcal{I}_{\varepsilon,n}(j) = \min\{k > \mathcal{I}_{\varepsilon,n}(j-1) : k \in K_{\varepsilon,n}(\omega)\},$$

*i.e.,* $\mathcal{I}_{\varepsilon,n}(j)(\omega)$ *is the index of* $j$*th iteration* $k$ *such that* $\mathcal{Q}_t(x_n^k) - \mathcal{Q}_t^k(x_n^k) \geq \varepsilon$. *Then random variables* $(y_n^{\mathcal{I}_{\varepsilon,n}(j)})_{j\geq 1}$ *defined on sample space* $\Omega_1(\varepsilon)$ *are independent, have the distribution of* $y_n^1$ *and therefore by the Strong Law of Large numbers we have*

$$\mathbb{P}\left(\lim_{N\to+\infty} \frac{1}{N}\sum_{j=1}^{N} y_n^{\mathcal{I}_{\varepsilon,n}(j)} = \mathbb{E}[y_n^1]\right) = 1. \tag{A.2}$$

***Proof*** (i) Define on the sample space $\Omega_0(\varepsilon)$ the random variables $(w_{\varepsilon,m,i}^k)_k$ by

$$w_{\varepsilon,m,i}^k(\omega) = \begin{cases} 1 & \text{if } k \in K_{\varepsilon,m,i}(\omega) \\ 0 & \text{otherwise.} \end{cases}$$

To alleviate notation ($\varepsilon, m, n, i$ being fixed), let us put $w^k := w_{\varepsilon,m,i}^k$, $\mathcal{I}(j) := \mathcal{I}_{\varepsilon,m,i}(j)$, For $\overline{y}_j \in \{0, 1\}$, we have

$$\mathbb{P}\Big(y_n^{\mathcal{I}(j)} = \overline{y}_j\Big) = \sum_{\overline{\mathcal{I}}_j=1}^{\infty} \mathbb{P}\Big(y_n^{\overline{\mathcal{I}}_j} = \overline{y}_j; \mathcal{I}(j) = \overline{\mathcal{I}}_j\Big). \tag{A.3}$$

Observe that the event $\mathcal{I}(j) = \overline{\mathcal{I}}_j$ can be written as the union $\bigcup_{1\leq\overline{\mathcal{I}}_1<\overline{\mathcal{I}}_2<\ldots<\overline{\mathcal{I}}_j} E(\overline{\mathcal{I}}_1, \ldots, \overline{\mathcal{I}}_j)$ of events

$$E(\overline{\mathcal{I}}_1, \ldots, \overline{\mathcal{I}}_j) := \left\{ \begin{array}{l} w^{\overline{\mathcal{I}}_1} = \ldots = w^{\overline{\mathcal{I}}_j} = 1, \\ w^{\ell} = 0, 1 \leq \ell < \overline{\mathcal{I}}_j, \ell \notin \{\overline{\mathcal{I}}_1, \ldots, \overline{\mathcal{I}}_j\} \end{array} \right\}.$$

Due to Assumption (H2) observe that random variable $y_n^{\overline{\mathcal{I}}_j}$ is independent of random variables $w^i$, $i = 1, \ldots, \overline{\mathcal{I}}_j$, and therefore events $\{y_n^{\overline{\mathcal{I}}_j} = \overline{y}_j\}$ and $\{\mathcal{I}(j) = \overline{\mathcal{I}}_j\}$ are independent which gives

$$\begin{aligned}\mathbb{P}\Big(y_n^{\mathcal{I}(j)} = \overline{y}_j\Big) &= \sum_{\overline{\mathcal{I}}_j=1}^{\infty} \mathbb{P}\Big(y_n^{\overline{\mathcal{I}}_j} = \overline{y}_j; \mathcal{I}(j) = \overline{\mathcal{I}}_j\Big) \\ &= \sum_{\overline{\mathcal{I}}_j=1}^{\infty} \mathbb{P}\Big(y_n^{\overline{\mathcal{I}}_j} = \overline{y}_j\Big)\mathbb{P}\Big(\mathcal{I}(j) = \overline{\mathcal{I}}_j\Big) \\ &= \mathbb{P}\Big(y_n^1 = \overline{y}_j\Big) \sum_{\overline{\mathcal{I}}_j=1}^{\infty} \mathbb{P}\Big(\mathcal{I}(j) = \overline{\mathcal{I}}_j\Big) = \mathbb{P}\Big(y_n^1 = \overline{y}_j\Big)\end{aligned} \tag{A.4}$$

where we have used the fact that $y_n^1$ and $y_n^{\overline{\mathcal{I}}_j}$ have the same distribution (from (H2)).
Next for $\overline{y}_1, \ldots, \overline{y}_p \in \{0, 1\}$, we have

$$\begin{aligned}&\mathbb{P}\Big(y_n^{\mathcal{I}(1)} = \overline{y}_1, \ldots, y_n^{\mathcal{I}(p)} = \overline{y}_p\Big) \\ &= \sum_{1\leq\overline{\mathcal{I}}_1<\overline{\mathcal{I}}_2<\ldots<\overline{\mathcal{I}}_p}^{\infty} \mathbb{P}\Big(y_n^{\overline{\mathcal{I}}_1} = \overline{y}_1; \ldots; y_n^{\overline{\mathcal{I}}_p} = \overline{y}_p; \mathcal{I}(1) = \overline{\mathcal{I}}_1; \ldots; \mathcal{I}(p) = \overline{\mathcal{I}}_p\Big).\end{aligned}$$

By the same reasoning as above, the event

$$\Big\{y_n^{\overline{\mathcal{I}}_1} = \overline{y}_1; \ldots; y_n^{\overline{\mathcal{I}}_{p-1}} = \overline{y}_{p-1}; \mathcal{I}(1) = \overline{\mathcal{I}}_1; \ldots; \mathcal{I}(p) = \overline{\mathcal{I}}_p\Big\}$$

can be expressed in terms of random variables $y_n^{\overline{\mathcal{I}}_1}, \ldots, y_n^{\overline{\mathcal{I}}_{p-1}}, w_n^{\overline{\mathcal{I}}_1}, \ldots, w_n^{\overline{\mathcal{I}}_p}$, and is therefore independent of event $\{y_n^{\overline{\mathcal{I}}_p} = \overline{y}_p\}$. It follows that

$$\begin{aligned}&\mathbb{P}\Big(y_n^{\mathcal{I}(j)} = \overline{y}_j, 1 \leq j \leq p\Big) \\ &= \sum_{1\leq\overline{\mathcal{I}}_1<\overline{\mathcal{I}}_2<\ldots<\overline{\mathcal{I}}_p}^{\infty} \mathbb{P}\Big(y_n^{\overline{\mathcal{I}}_p} = \overline{y}_p\Big)\mathbb{P}\Big(y_n^{\overline{\mathcal{I}}_j} = \overline{y}_j, 1 \leq j \leq p-1; \mathcal{I}(j) = \overline{\mathcal{I}}_j, 1 \leq j \leq p\Big) \\ &= \mathbb{P}\Big(y_n^1 = \overline{y}_p\Big) \sum_{1\leq\overline{\mathcal{I}}_1<\overline{\mathcal{I}}_2<\ldots<\overline{\mathcal{I}}_p}^{\infty} \mathbb{P}\Big(y_n^{\overline{\mathcal{I}}_j} = \overline{y}_j, 1 \leq j \leq p-1; \mathcal{I}(j) = \overline{\mathcal{I}}_j, 1 \leq j \leq p\Big) \\ &= \mathbb{P}\Big(y_n^1 = \overline{y}_p\Big) \sum_{1\leq\overline{\mathcal{I}}_1<\overline{\mathcal{I}}_2<\ldots<\overline{\mathcal{I}}_{p-1}}^{\infty} \mathbb{P}\Big(y_n^{\overline{\mathcal{I}}_j} = \overline{y}_j, 1 \leq j \leq p-1; \mathcal{I}(j) = \overline{\mathcal{I}}_j, 1 \leq j \leq p-1\Big) \\ &= \mathbb{P}\Big(y_n^1 = \overline{y}_p\Big)\mathbb{P}\Big(y_n^{\mathcal{I}(j)} = \overline{y}_j, 1 \leq j \leq p-1\Big).\end{aligned} \tag{A.5}$$

By induction this implies

$$\mathbb{P}\Big(y_n^{\mathcal{I}(j)} = \overline{y}_j, 1 \le j \le p\Big) = \prod_{j=1}^{p} \mathbb{P}\Big(y_n^1 = \overline{y}_j\Big) \overset{(A.4)}{=} \prod_{j=1}^{p} \mathbb{P}\Big(y_n^{\mathcal{I}(j)} = \overline{y}_j\Big) \quad (A.6)$$

which shows that random variables $(y_n^{\mathcal{I}(j)})_{j\ge 1}$ are independent.

The proof of (ii) is similar to the proof of (i). □

**Proof of** (3.40) **and** (3.53). As in [5], we can now use the previous lemma to prove (3.40) and (3.53). Let us prove (3.40). By contradiction, assume that (3.40) does not hold. Then, there is $\varepsilon > 0$ such that the set $\Omega_0(\varepsilon)$ defined in Lemma A.1 is nonempty. By Lemma A.1, this implies that (A.1) holds. But due to (3.39), only a finite number of indices $\mathcal{I}_{\varepsilon,m,i}(j)$ can be in $\mathcal{S}_n$ (with corresponding variable $y_n^{\mathcal{I}_{\varepsilon,m,i}(j)}$ being one) and therefore $\mathbb{P}\left(\lim_{N\to+\infty} \frac{1}{N}\sum_{j=1}^{N} y_n^{\mathcal{I}_{\varepsilon,m,i}(j)} = 0\right) = 1$, which is a contradiction with (A.1).

The proof of (3.53) is similar to the proof of (3.40), by contradiction and using (3.52) and Lemma A.1-(ii) (see also [5,6]). □

## References


1. Andersen, E.D., Andersen, K.D.: The MOSEK optimization toolbox for MATLAB manual. Version 9.2, 2019. https://www.mosek.com/documentation/
2. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numer. Math. **4**, 238–252 (1962)
3. Birge, J.R.: Decomposition and partitioning methods for multistage stochastic linear programs. Oper. Res. **33**, 989–1007 (1985)
4. Ding, L., Ahmed, S., Shapiro, A.: A python package for multi-stage stochastic programming. Optimization Online (2019)
5. Girardeau, P., Leclere, V., Philpott, A.B.: On the convergence of decomposition methods for multistage stochastic convex programs. Math. Oper. Res. **40**, 130–145 (2015)
6. Guigues, V.: Convergence analysis of sampling-based decomposition methods for risk-averse multistage stochastic convex programs. SIAM J. Optim. **26**, 2468–2494 (2016)
7. Guigues, V.: Dual dynamic programing with cut selection: convergence proof and numerical experiments. Eur. J. Oper. Res. **258**, 47–57 (2017)
8. Guigues, V.: Inexact cuts in stochastic dual dynamic programming. SIAM J. Optim. **30**, 407–438 (2020)
9. Guigues, V.: Inexact stochastic mirror descent for two-stage nonlinear stochastic programs. Accepted for publication in Mathematical Programming (2020). https://arxiv.org/pdf/1805.11732.pdf
10. Guigues, V., Bandarra, M.: Single cut and multicut SDDP with cut selection for multistage stochastic linear programs: convergence proof and numerical experiments. Computational Management Science. https://arxiv.org/abs/1902.06757
11. Guigues, V., Monteiro, R., Svaiter, B.: Inexact cuts in SDDP applied to multistage stochastic nondifferentiable problems. arXiv (2020). https://arxiv.org/abs/2004.02701
12. Guigues, V., Römisch, W.: Sampling-based decomposition methods for multistage stochastic programs based on extended polyhedral risk measures. SIAM J. Optim. **22**, 286–312 (2012)
13. Guigues, V., Römisch, W.: SDDP for multistage stochastic linear programs based on spectral risk measures. Oper. Res. Lett. **40**, 313–318 (2012)
14. Guigues, V., Shapiro, A., Cheng, Y.: Duality and sensitivity analysis of multistage linear stochastic programs. Optimization OnLine (2019)

15. Guigues, V., Tekaya, W., Lejeune, M.: Regularized decomposition methods for deterministic and stochastic convex optimization and application to portfolio selection with direct transaction and market impact costs. Optim. Eng. **21**, 1133–1165 (2020)
16. Kelley, J.E.: The cutting plane method for solving convex programs. J. SIAM **8**, 703–712 (1960)
17. Kiwiel, K.C.: An aggregate subgradient method for nonsmooth convex minimization. Math. Program. **27**, 320–341 (1983)
18. Kiwiel, K.C.: Proximity control in bundle methods for convex nondifferentiable minimization. Math. Program. **46**, 105–122 (1990)
19. Kozmík, V., Morton, D.P.: Evaluating policies in risk-averse multi-stage stochastic programming. Math. Program. **152**, 275–300 (2015)
20. Lemaréchal, C.: An extension of Davidon methods to non-differentiable problems. Math. Program. Study **3**, 95–109 (1975)
21. Lemaréchal, C., Nemirovski, A., Nesterov, Y.: New variants of bundle methods. Math. Program. **69**, 111–147 (1995)
22. Liu, R.P., Shapiro, A.: Risk neutral reformulation approach to risk averse stochastc programming. arXiv (2018). arXiv:1901.01302
23. Pereira, M.V.F., Pinto, L.M.V.G.: Multi-stage stochastic optimization applied to energy planning. Math. Program. **52**, 359–375 (1991)
24. Philpott, A., de Matos, V.: Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. Eur. J. Oper. Res. **218**, 470–483 (2012)
25. Philpott, A., de Matos, V., Finardi, E.: Improving the performance of stochastic dual dynamic programming. J. Comput. Appl. Math. **290**, 196–208 (2012)
26. Philpott, A.B., Guan, Z.: On the convergence of stochastic dual dynamic programming and related methods. Oper. Res. Lett. **36**, 450–455 (2008)
27. Powell, W.P.: Approximate Dynamic Programming, 2nd edn. Wiley (2011)
28. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
29. Rockafellar, T.: Conjugate Duality and Optimization. No 16 in Conference Board of Math. Sciences Series, pp. 1–79. SIAM Publications (1974)
30. Ruszczyński, A., Shapiro, A.: Conditional risk mappings. Math. Oper. Res. **31**, 544–561 (2006)
31. Ruszczyński, A., Shapiro, A.: Optimization of convex risk functions. Math. Oper. Res. **31**, 433–452 (2006)
32. Shapiro, A.: Analysis of stochastic dual dynamic programming method. Eur. J. Oper. Res. **209**, 63–72 (2011)
33. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on Stochastic Programming: Modeling and Theory. SIAM, Philadelphia (2009)
34. Shapiro, A., Ding, L.: Periodical multistage stochastic programs. SIAM J. Optim. **30**, 2083–2102 (2020)
35. Van Slyke, R.M., Wets, R.J.-B.: L-shaped linear programs with applications to optimal control and stochastic programming. SIAM J. Appl. Math. **17**, 638–663 (1969)
36. Wolkowicz, H., Saigal, R., Vandenberghe, L.: Handbook of Semidefinite Programming. Springer, Berlin (2000)