

Constant Depth Decision Rules for multistage optimization under uncertainty

Vincent Guigues *

Anatoli Juditsky †

Arkadi Nemirovski ‡

Abstract

In this paper, we introduce a new class of decision rules, referred to as *Constant Depth Decision Rules* (CDDRs), for multistage optimization under linear constraints with uncertainty-affected right-hand sides. We consider two uncertainty classes: discrete uncertainties which can take at each stage at most a fixed number d of different values, and polytopic uncertainties which, at each stage, are elements of a convex hull of at most d points. Given the *depth* μ of the decision rule, the decision at stage t is expressed as the sum of t functions of μ consecutive values of the underlying uncertain parameters. These functions are arbitrary in the case of discrete uncertainties and are poly-affine in the case of polytopic uncertainties. For these uncertainty classes, we show that when the uncertain right-hand sides of the constraints of the multistage problem are of the same additive structure as the decision rules, these constraints can be reformulated as a system of linear inequality constraints where the numbers of variables and constraints is $O(1)(n+m)d^\mu N^2$ with n the maximal dimension of control variables, m the maximal number of inequality constraints at each stage, and N the number of stages.

As an illustration, we discuss an application of the proposed approach to a Multistage Stochastic Program arising in the problem of hydro-thermal production planning with interstage dependent inflows. For problems with a small number of stages, we present the results of a numerical study in which optimal CDDRs show similar performance, in terms of optimization objective, to that of Stochastic Dual Dynamic Programming (SDDP) policies, often at much smaller computational cost.

Keywords: Stochastic Programming; Robust Optimization; Decision rules; Stochastic Dual Dynamic Programming.

AMS subject classifications: 90C15, 90C90.

*Corresponding author, School of Applied Mathematics FGV/EMAp, 22 250-900 Rio de Janeiro, Brazil
vincent.guigues@fgv.br

†LJK, Université Grenoble Alpes, 700 Avenue Centrale 38041 Domaine Universitaire de Saint-Martin-d'Hères, France, anatoli.juditsky@imag.fr

‡Georgia Institute of Technology, Atlanta, Georgia 30332, USA, nemirovs@isye.gatech.edu

1 Introduction

Multistage optimization problems under uncertainty arise in many real-life applications in finance and engineering, see for instance [6, 25] and references therein, but are challenging to solve. Such problems with stochastic uncertainties — Multistage Stochastic Programs (MSPs) — are typically replaced with their discretized (scenario) formulations [25]. Because the number of variables in the deterministic equivalent of such approximations increases very fast with the number N of stages these problems are computationally intractable [26] in general. “Practical” solution methods for convex MSPs often use decomposition techniques based on Dynamic Programming with two popular methods being Approximate Dynamic Programming (ADP) and Stochastic Dual Dynamic Programming (SDDP), introduced in [20]. The latter method is a sampling-based extension of the Nested Decomposition method [5] which relies upon computing approximations of the Bellman functions. Both techniques have been applied to a variety of real-life problems and several enhancements of these methods have been proposed recently, see, e.g., [22, 24] for an overview of these techniques and [21, 12, 15, 16, 17] for convergence analysis and some of recent variations. However, to the best of our knowledge, apart from some asymptotic convergence results, no theoretical performance guaranty is available for these methods.

Another general approach which has proved successful for various classes of uncertain optimization problems relies on restricting the control policies to belong to certain parametric families of functions of those uncertain parameters which are known at the moment when the decision is to be applied. This approach often allows for a tractable reformulation of the corresponding optimization problem. The simplest rules of this type are affine rules which were studied for Stochastic and Chance-Constrained programming [8, 7] and, more recently, in the context of Robust Optimization [4, 3]. Though affine decision strategies are generally suboptimal [10] they usually provide a convenient conservativeness/numerical cost tradeoff and have been successfully used in many real-life applications, such as production management [13], portfolio optimization [23], and unit commitment problems [19]. Nevertheless, to reduce the conservativeness of affine decisions at the price of increased computational complexity, other families of parametric decision rules were recently proposed, e.g., those using liftings as in [11], polynomial decision rules [2], or projected linear decision rules combined with dynamic chance constraints as in [18], among others.

In this paper, we discuss a new family of decision rules for multistage uncertain problems called *Constant Depth Decision Rules* (CDDRs). We consider two classes of problems: problems with discrete uncertain parameters which, at each stage, take values in a finite set, and problems with polytopic uncertainties taking values, at each stage, in a convex hull of a finite number of points. When dealing with discrete uncertainties the corresponding decision $x_t(\cdot)$ at stage t is additive with memory μ , i.e., is a sum of t (arbitrary) functions $u_s^t(\cdot)$, $s = 1, \dots, t$, of μ consecutive observations of uncertain parameters preceding stage s (here μ is a given *depth* parameter of the rule). When uncertainties are polytopic, decision variable $x_t(\cdot)$ at stage t is a poly-affine function, i.e. a sum of functions $v_s^t(\cdot)$, $s = 1, \dots, t$, of μ consecutive observations of uncertainties preceding stage s which are

affine in each argument (see the precise definition in Section 3.1). We study a class of optimization problems under uncertainty with linear constraints and uncertain right-hand sides. Our objective is to minimize convex deterministic or stochastic objective over the set of Constant Depth Decision Rules satisfying the constraints. Our principal contributions are the following.

- We provide a tractable equivalent reformulation of the problem to be solved to compute optimal CDDRs. Specifically, for an N -stage optimization problem with no more than m linear constraints and n variables at each stage t , and at most d possible values of uncertain parameters at each stage t , and right-hand side uncertainties of the same additive structure as our decision rules, we rewrite the system of linear constraints on CDDRs $x_t(\cdot)$ as a linear system of constraints with the total numbers of scalar decision variables and scalar constraints bounded with $O(1)(m+n)N^2d^\mu$.
- We establish similar results for the case when the uncertainty at each stage runs through a polytope with at most d vertices, and decision rules and uncertain right-hand sides of the linear constraints are allowed to be poly-affine, with memory depth μ , functions of the uncertainties.
- Finally, to illustrate an application of the proposed approach, we compare it to that of SDDP on a toy problem of hydro-thermal production planning using a MATLAB toolbox https://github.com/vguigues/Constant_Depth_Decision_Rules_Library for computing optimal CDDRs for multistage linear programs with discrete uncertainties.

The paper is organized as follows. In Section 2 we describe the CDDR approach in the case of discrete uncertainties. In particular, we show how to reformulate the system of uncertain linear constraints of the problem of interest, when solved using CDDRs, as a linear system of inequalities on the coefficients of the CDDRs, with size of the system polynomial, the memory depth μ being fixed, in the sizes of the problem of interest. In Section 3 we develop CDDRs applied to problems with uncertainties supported on polytopes. We illustrate the use of the proposed methodology on an example of hydro-thermal production planning application described in Section 4. Results of numerical experiments comparing CDDRs to SDDP are reported in Section 4.2 along with a comparative discussion of these approaches to the application in question.

2 Fixed memory decision making under discrete uncertainty

2.1 The problem

Consider the situation where we control a system S evolving over time horizon $1, \dots, N$ and affected by our decisions $x_t \in \mathbf{R}^{n_t}$ and external disturbances (“uncertainties”) ξ_t , $1 \leq t \leq N$, where x_t is allowed to be a function of $\xi^t = (\xi_1, \dots, \xi_t)$: $x_t = x_t(\xi^t)$. We assume that

A.1. For every $t \leq N$, disturbance ξ_t takes values in a finite set of cardinality d_t , which we identify, without loss of generality, with $D_t = \{1, 2, \dots, d_t\}$.

A.2. Feasibility of controls x_t is expressed by linear constraints

$$\sum_{\tau=1}^t A^{t\tau} x_\tau(\xi^\tau) \leq b_t(\xi^t) \in \mathbf{R}^{m_t}, \quad t = 1, \dots, N \quad (2.1)$$

which should be satisfied for all trajectories $\xi^N \in D^N = D_1 \times \dots \times D_N$.

Under these restrictions, we want to minimize a given objective. In order to make the problem computationally tractable, we further restrict the structure of the decision rules we allow for, same as the structure of the objective and the type of dependence of $b_t(\xi^t)$ on ξ^t , specifically, as follows.

Preliminaries: additive functions with memory μ . To avoid messy notation, we augment a sequence $\xi^\tau = (\xi_1, \xi_2, \dots, \xi_\tau) \in D^\tau := D_1 \times \dots \times D_\tau$ with terms ξ_s with nonpositive indices $s \leq 0$; all these terms take values in the singletons $D_s = \{1\}$, that is, from now on $\xi_s = 1$ when $s \leq 0$. Besides this, for a *trajectory of disturbances* — a sequence $\xi^N = (\xi_1, \dots, \xi_N) \in D^N$ — and $p \leq q \leq N$, we denote by $\xi_{p:q}$ the fragment $\xi_p, \xi_{p+1}, \dots, \xi_q$ of ξ^N , with the already made convention that $\xi_s = 1$ for $s \leq 0$. Finally, let us agree that notation ξ^t when used in the same context with ξ_p or $\xi_{p:q}$, with $q \leq t$, means that ξ_p is the p -th entry, and $\xi_{p:q}$ is the fragment $(\xi_p, \xi_{p+1}, \dots, \xi_q)$ of the sequence $\xi^t = (\xi_1, \dots, \xi_t) \in D^t$.

Given a positive integer μ , let us call a function $g(\xi_1, \xi_2, \dots, \xi_t) : D^t \rightarrow \mathbf{R}^\nu$ *additive with memory μ* , if

$$g(\xi_1, \dots, \xi_t) = \sum_{\tau=1}^t u_{\tau \xi_{\tau-\mu+1:\tau}},$$

where *coefficients* $u_{\tau \xi_{\tau-\mu+1:\tau}}$ of g take values in \mathbf{R}^ν . For example, an additive with memory 1 function $g : D^t \rightarrow \mathbf{R}^\nu$ is specified by the collection $\{u_{\tau \xi} \in \mathbf{R}^\nu : 1 \leq \tau \leq t, 1 \leq \xi \leq d_\tau\}$ of coefficients of g , and the value of g at a $\xi^t \in D^t$ is the sum $\sum_{\tau=1}^t u_{\tau \xi_\tau}$ of the coefficients taken “along the trajectory $\xi^t = (\xi_1, \dots, \xi_t)$.”

Structural restrictions. In the sequel, aside of Assumptions **A.1-2** we have already made, we fix a positive integer μ and impose the following restrictions on the structure of problem constraints and decision rules:

A.3. The right-hand sides $b_t(\xi^t)$ in design specifications (2.1) are additive with memory μ .

A.4. The decision rules $x_t(\xi^t)$ are restricted to be additive with memory μ .

According to **A.3**, we have

$$b_t(\xi^t) = \sum_{\tau=1}^t \beta_{\tau\xi_{\tau-\mu+1:\tau}}^t \quad (2.2)$$

for some collection

$$\left\{ \beta_{\tau\xi}^t \in \mathbf{R}^{m_t}, 1 \leq \tau \leq t, \xi \in D_{\tau-\mu+1:\tau} \right\}, \\ D_{p:q} = D_p \times D_{p+1} \times \dots \times D_q.$$

Similarly, by **A.4**, candidate decision rules $x^N = \{x_t(\cdot) : 1 \leq t \leq N\}$ can be parameterized by collections

$$u^N = \{u_{\tau\xi}^t \in \mathbf{R}^{n_t} : 1 \leq \tau \leq t \leq N, \xi \in D_{\tau-\mu+1:\tau}\}$$

according to

$$x_t(\xi^t) = \sum_{\tau=1}^t u_{\tau\xi_{\tau-\mu+1:\tau}}^t. \quad (2.3)$$

Our final assumption is as follows:

A.5. The objective to be minimized is an efficiently computable convex function $f(u^N)$ of the vector u^N of parameters of a candidate decision rule.

Immediate examples of the objectives of the required structure are given by the following construction: we are given a real-valued function $F(x^N, \xi^N)$ which is convex in $x^N = [x_1; x_2; \dots; x_N] \in \mathbf{R}^{n_1} \times \dots \times \mathbf{R}^{n_N}$, and a probability distribution P on the set D^N of N -element trajectories of disturbances, and our objective is the expectation $f(u^N) = \mathbf{E}_{\xi^N \sim P} \{F(x^N(\xi^N), \xi^N)\}$ of the “loss” $F(x^N, \xi^N)$ as evaluated at our controls. When decision rules $x_t(\xi^t)$ are additive, with memory μ , objectives of this type definitely are convex in u^N ; whether they are efficiently computable depends on the structure of P . Computability definitely takes place when P is known and supported on a subset of D^N of moderate cardinality. When this is not the case, but we can efficiently sample from P , we can arrive at the latter situation when replacing the actual objective by its Sample Average Approximation (that is, approximating P by the uniform distribution on a reasonably large sample of trajectories of disturbances drawn from P).

Another important example of objective satisfying **A.5** is the maximum, over all trajectories $\xi^N \in D^N$, of a linear functional $\sum_{t=1}^N \eta_t^T x_t(\xi^t)$ of the control trajectory, cf. Section 2.3.

2.2 Processing the problem

Treating $u^N = \{u_{\tau\xi}^t \in \mathbf{R}^{n_t} : 1 \leq \tau \leq t \leq N, \xi \in D_{\tau-\mu+1:\tau}\}$ as our design variables, the constraints (2.1) read

$$\sum_{\tau=1}^t A^{t\tau} x_{\tau}(\xi^{\tau}) - b_t(\xi^t) \leq 0 \quad \forall (t, 1 \leq t \leq N, \xi^N \in D^N), \quad (2.4)$$

or equivalently (see (2.2), (2.3))

$$\sum_{s=1}^t \left[\sum_{\tau=s}^t A^{t\tau} u_{s\xi_{s-\mu+1:s}}^{\tau} - \beta_{s\xi_{s-\mu+1:s}}^t \right] \leq 0 \quad \forall (t, 1 \leq t \leq N, \xi^N \in D^N). \quad (2.5)$$

The crucial fact for us is that *constraints (2.5) can be reduced to an explicit system of linear inequality constraints on the design variables $u_{\tau\xi}^t$ and additional “analysis” variables.* The construction goes as follows.

1. We introduce variables $y_{\tau\xi}^t \in \mathbf{R}^{m_t}$, $1 \leq \tau \leq t \leq N, \xi \in D_{\tau-\mu+1:\tau}$, and link them to our decision variables $u_{\tau\xi}^t$ by linear equality constraints

$$y_{s\xi}^t = \sum_{\tau=s}^t A^{t\tau} u_{s\xi}^{\tau} - \beta_{s\xi}^t, \quad \xi \in D_{s-\mu+1:s}, 1 \leq s \leq t \leq N. \quad (2.6)$$

In terms of these variables (2.5) reads

$$\sum_{s=1}^t y_{s\xi_{s-\mu+1:s}}^t \leq 0 \quad \forall \xi^N \in D^N. \quad (2.7)$$

To avoid messy notation, we describe our subsequent actions separately for the case of $\mu = 1$ and of $\mu > 1$.

2.A: Case of $\mu = 1$. For every $t \in \{1, \dots, N\}$, we introduce variables $z_s^t \in \mathbf{R}^{m_t}$, $1 \leq s \leq t$, and impose the linear inequalities

$$\begin{aligned} z_s^t &\geq y_{s\xi}^t + z_{s+1}^t \quad \forall \xi \in D_s, s = t, t-1, \dots, 1, & (a) \\ z_1^t &\leq 0 & (b) \end{aligned} \quad (2.8)$$

where $z_{t+1}^t \equiv 0$. Clearly, the i -th entry $[z_s^t]_i$ in z_s^t is an upper bound on $\max_{\xi^N \in D^N} \left[\sum_{r=s}^t y_{r\xi_r}^t \right]_i$, and constraints (a) allow to make this bound equal to the latter quantity. Consequently, the system \mathcal{S} of constraints (2.6) and (2.8) on variables u, y, z provides a polyhedral representation of the solution set of (2.4). In other words, a collection u^N of actual design variables $u_{\tau\xi}^t$ satisfies constraints (2.4) if and only if u^N can be extended, by properly selected values of y - and z -variables, to a feasible solution of \mathcal{S} . On the other hand, (2.8) is entrywise decomposable (it is a collection of m_t systems

of linear inequalities, with the i -th system involving only the i -th entries in y - and z -vectors), and as far as the i -th entries in $y_{s\xi}^t$ and z_s^t are concerned, (2.8.a) is nothing but the “backward” Dynamic Programming description of an upper bound $[z_1^t]_i$ on $\max_{\xi^N \in D^N} \left[\sum_{s=1}^t y_{s\xi_s}^t \right]_i$ (recall that we are in the case of $\mu = 1$), while (2.8.b) says that the resulting bound should be nonpositive for all i , exactly as required in (2.7).

2.B: Case of $\mu \geq 2$. Same as in the case of $\mu = 1$, what follows is nothing but backward Dynamic Programming description, expressed by linear inequalities, of vectors with i -th entry, $i \leq m_t$, upper-bounding $\max_{\xi^N \in D^N} \left[\sum_{s=1}^t y_{s\xi_{s-\mu+1:s}}^t \right]_i$. As is immediately seen, to get this description it suffices to introduce variables $z_{s\eta}^t \in \mathbf{R}^{m_t}$, $1 \leq s \leq t$, $\eta \in D_{s-\mu+1:s-1}$, and subject them, along with the y -variables, to linear constraints

$$\begin{aligned} z_{t\xi_{t-\mu+1:t-1}}^t &\geq y_{t\xi_{t-\mu+1:t}}^t \quad \forall \xi_t \in D_t, \\ z_{s\xi_{s-\mu+1:s-1}}^t &\geq y_{s\xi_{s-\mu+1:s}}^t + z_{(s+1)\xi_{s-\mu+2:s}}^t \quad \forall \xi^N \in D^N, s = t-1, t-2, \dots, 1, \\ z_{1\xi_{2-\mu:0}}^t &\leq 0. \end{aligned} \tag{2.9}$$

Similarly to the case of $\mu = 1$, the system of all constraints (2.6) and (2.9) on variables u, y, z gives a polyhedral representation of the solution set of (2.4).

The bottom line is that *under Assumptions A.1-5, the problem of interest can be straightforwardly reduced to the problem of minimizing an efficiently computable convex objective $f(u^N)$ over u -, y -, z -variables satisfying an explicit system of linear inequality constraints*. Note that for every fixed μ , the total number of variables and constraints in the resulting problem \mathcal{P} is polynomial in the sizes of the problem of interest. Specifically, assuming $d_t \leq d$, $m_t \leq m$ and $n_t \leq n$ for all t , the total numbers of scalar decision variables and scalar constraints in \mathcal{P} do not exceed $O(1)(m+n)N^2d^\mu$.

2.3 Modifications

In the above exposition, we treated vectors $\beta_{\tau\xi}^t$ as part of the data. It is immediately seen that when changing the status of some of β 's from being part of the data to being additional decision variables and adding, say, linear constraints on these variables, we preserve tractability of the resulting problem: our backward Dynamic Programming still allows us to convert constraints (2.6) and (2.7) modified in this way into an explicit system of linear inequalities on “variables of interest” (components of u^N and new design variables coming from β 's) and additional y - and z -analysis variables.

An immediate application of this observation is as follows. Suppose that Assumptions A.1-4 hold and assume that our goal is to minimize the worst case (i.e., the largest over $\xi^N \in D^N$) value

of the function

$$F[\{x_t(\xi^t)\}_{t=1}^N] = \max_{\ell \leq L} \sum_{t=1}^N h_{t\ell}^T x_t(\xi^t).$$

To this end it suffices to augment our original design variables $u^N = \{u_{\tau\xi}^t\}$ parameterizing additive, with memory μ , candidate decision rules with a new decision variable w and extend the original system $\sum_{\tau=1}^N A^{N\tau} x_{\tau}(\xi^{\tau}) \leq b_N(\xi^N)$ of the last stage constraints by adding to it constraints

$$\sum_{t=1}^N h_{t\ell}^T x_t(\xi^t) \leq w, \ell = 1, \dots, L.$$

As a result, we get a parametric, the parameter being $\xi^N \in D^N$, system of linear inequalities on u^N and w . Applying backward Dynamic Programming in exactly the same way as above, we convert this system into an explicit system of linear inequalities on u^N , w , and additional y - and z -variables. Minimizing the worst-case value of the above criterion is thus reduced to an explicit Linear Programming problem in w and u -, y -, z -variables.

Until now we have assumed that control feasibility is expressed in terms of the system (2.1) of linear constraints with uncertain right-hand sides. It may be worth mentioning that the proposed approach can be straightforwardly modified to deal with linear constraints with uncertain matrices or even specific nonlinear constraints at the price of restricting severely the class of control strategies.

Indeed, let us assume from now on that the control action x_t depends solely on $\xi_{t-\mu+1:t} = (\xi_{t-\mu+1}, \dots, \xi_t)$ so that representation (2.3) reduces to

$$x_t(\xi^t) = u_{\xi_{t-\mu+1:t}}^t.$$

Now, let us consider an uncertain linear system (2.1) satisfying Assumptions **A.1.**, **A.2.** and **A.3.** with uncertain technology matrices $A^{t\tau}$. More precisely, we assume that matrices $A^{t\tau}$ depend on the fragment $\xi_{\tau-\mu+1:\tau} = (\xi_{\tau-\mu+1}, \xi_{\tau-\mu+2}, \dots, \xi_{\tau})$ of ξ^{τ} , i.e., constraints (2.1) are replaced with the constraints

$$\forall(t : 1 \leq t \leq N) : \sum_{\tau=1}^t A^{t\tau}(\xi_{\tau-\mu+1:\tau}) u_{\xi_{\tau-\mu+1:\tau}}^{\tau} \leq b_t(\xi^t),$$

where b_t , same as before, is additive with memory μ .

As was done in Section 2.2, we can rewrite the constraints replacing y -variables with

$$y_{s\xi}^t = A^{ts}(\xi) u_{\xi}^s - \beta_{s\xi}^t, \quad \xi \in D_{s-\mu+1:s}, 1 \leq s \leq t \leq N,$$

and therefore all the machinery developed in Section 2.2 can be applied.

Given $x_0 \in \mathbf{R}^{n_0}$, let us now consider a system of convex nonlinear constraints

$$\sum_{\tau=1}^t G^{t\tau}(x_{\tau}, x_{\tau-1}, \xi_{\tau-\mu+1:\tau}) \leq 0 \in \mathbf{R}^{m_t}, \quad t = 1, \dots, N, \quad (2.10)$$

“coupling” control actions at subsequent stages, which should be satisfied for all trajectories $\xi^N \in D^N = D_1 \times \dots \times D_N$ with (ξ_t) satisfying **A.1**. We suppose that components $G_i^{t\tau}(x, x', \xi_{\tau-\mu+1:\tau})$, $i = 1, \dots, m_t$, of $G^{t\tau}$ are convex in x and x' for all possible $(\xi_1, \xi_2, \dots, \xi_N)$ and all t, τ . Same as above, in this setting, we derive the following representation of constraints (2.10) in variables $u_{\xi_{t-\mu+1:t}}^t$, $t = 1, \dots, N$, $\xi_{t-\mu+1:t} \in D_{t-\mu+1:t}$, and $z_{s, \xi_{s-\mu:s-1}}^t$, $2 \leq s \leq t \leq N$, $\xi_{s-\mu:s-1} \in D_{s-\mu:s-1}$, by Dynamic Programming:

$$\begin{aligned}
& G^{11}(u_{\xi_{2-\mu:1}}^1, x_0, \xi_{2-\mu:1}) \leq 0, \\
& \forall t = 2, \dots, N, \forall \xi_{t-\mu:t} \in D_{t-\mu:t} : \\
& \quad z_{t, \xi_{t-\mu:t-1}}^t \geq G^{tt}(u_{\xi_{t-\mu+1:t}}^t, u_{\xi_{t-\mu:t-1}}^{t-1}, \xi_{t-\mu+1:t}), \\
& \forall 3 \leq s \leq t \leq N, \xi_{s-\mu-1:s-1} \in D_{s-\mu-1:s-1} : \\
& \quad z_{s-1, \xi_{s-1-\mu:s-2}}^t \geq G^{ts-1}(u_{\xi_{s-\mu:s-1}}^{s-1}, u_{\xi_{s-\mu-1:s-2}}^{s-2}, \xi_{s-\mu:s-1}) + z_{s, \xi_{s-\mu:s-1}}^t, \\
& \forall 2 \leq t \leq N : \\
& \quad z_{2, \xi_{2-\mu:1}}^t + G^{t1}(u_{\xi_{2-\mu:1}}^1, x_0, \xi_{2-\mu:1}) \leq 0.
\end{aligned}$$

3 Fixed memory decision making under polytopic uncertainty

3.1 The problem

So far, we have considered multi-stage decision making under discrete uncertainty, where the external disturbance acting at the controlled system at time t takes one of d_t values known in advance. Let us now consider the case of *polytopic* uncertainty, where the disturbance at time t is a vector ζ_t taking values in a given polytope $\Delta_t \in \mathbf{R}^{\nu_t-1}$. Same as above, we allow for our decision at time t , $x_t \in \mathbf{R}^{n_t}$, to depend on the sequence $\zeta^t = (\zeta_1, \dots, \zeta_t)$. From now on we make the following assumptions (cf. Assumptions **A.1-2**):

B.1. For every $t \leq N$, disturbance ζ_t takes values in polytope $\Delta_t \subset \mathbf{R}^{\nu_t-1}$ given by the list of d_t scenarios χ_{ts} :

$$\Delta_t = \text{Conv}\{\chi_{ts}, 1 \leq s \leq d_t\} \subset \mathbf{R}^{\nu_t-1}. \quad (3.11)$$

We also assume that the scenarios affinely span \mathbf{R}^{ν_t-1} (this is w.l.o.g., since we can always replace the embedding space \mathbf{R}^{ν_t-1} of Δ_t by the affine span of Δ_t). Thus, $\nu_t \leq d_t$ for all t .

B.2. Feasibility of controls x_t is expressed by linear constraints

$$\sum_{\tau=1}^t A^{t\tau} x_{\tau}(\zeta^{\tau}) \leq b_t(\zeta^t) \in \mathbf{R}^{m_t}, \quad t = 1, \dots, N \quad (3.12)$$

which should be satisfied for all trajectories $\zeta^N \in \Delta^N = \Delta_1 \times \dots \times \Delta_N$.

Under these restrictions, we want to minimize a given objective. To ensure computational tractability we, same as in the case of discrete disturbances, impose structural restrictions on the allowed $x_t(\cdot)$'s, $b_t(\cdot)$'s, and on the objective. Our main restriction is that the policies $x_t(\cdot)$ and the right-hand sides $b_t(\cdot)$ are *poly-affine with memory μ* .

Poly-affine functions with memory μ . For notational convenience, we augment a sequence $\zeta^\tau = (\zeta_1, \zeta_2, \dots, \zeta_\tau) \in \Delta^\tau := \Delta_1 \times \dots \times \Delta_\tau$ with terms ζ_s with nonpositive indices $s \leq 0$; all these terms take values in the singletons $\Delta_s = \{0\} = \mathbf{R}^0$, that is, from now on $\zeta_s = 0 \in \mathbf{R}$ when $s \leq 0$. Given a positive integer μ , let us call function $g(\zeta_1, \zeta_2, \dots, \zeta_t) : \Delta^t \rightarrow \mathbf{R}^\nu$ *poly-affine with memory μ* , if

$$g(\zeta_1, \dots, \zeta_t) = \sum_{\tau=1}^t g_\tau(\zeta_{\tau-\mu+1}, \dots, \zeta_\tau),$$

where every component $g_\tau(\zeta_{\tau-\mu+1}, \dots, \zeta_\tau)$ of g takes values in \mathbf{R}^ν and is affine in each of its arguments $\zeta_{\tau-\mu+1}, \dots, \zeta_\tau$.

For example, an affine vector-valued function $g(\zeta^t) = \sum_{\tau=1}^t [P_\tau \zeta_\tau + p_\tau]$ of ζ_1, \dots, ζ_t is poly-affine with memory $\mu = 1$; its components are $g_\tau(\zeta_\tau) = P_\tau \zeta_\tau + p_\tau$.

Given $t \leq N$, we denote by $\lambda(\zeta_t) \in \mathbf{R}^{\nu_t}$ the vector of coefficients in the representation of $\zeta_t \in \mathbf{R}^{\nu_t-1}$ as an affine combination of the standard basic orths and the origin:

$$\lambda_i(\zeta_t) = [\zeta_t]_i, 1 \leq i < \nu_t, \lambda_{\nu_t}(\zeta_t) = 1 - \sum_{i=1}^{\nu_t-1} [\zeta_t]_i$$

with $\lambda_1(\zeta_t) \equiv 1$ when $t \leq 0$ (and, according to our convention, $\zeta_t = 0 \in \Delta_0 = \mathbf{R}^0 = \{0\}$). From affinity of $g_\tau(\zeta_{\tau-\mu+1}, \dots, \zeta_\tau)$ in each argument it follows that

$$g_\tau(\zeta_{\tau-\mu+1}, \dots, \zeta_\tau) = \sum_{\varkappa \in \mathcal{I}_\tau} \left[\prod_{s=1}^{\mu} \lambda_{\varkappa_s}(\zeta_{\tau-\mu+s}) \right] g_{\tau\varkappa},$$

$$\mathcal{I}_\tau = \{\varkappa = (\varkappa_1, \dots, \varkappa_\mu) : 1 \leq \varkappa_s \leq \nu_{\tau-\mu+s}, 1 \leq s \leq \mu\}.$$

As a result, a poly-affine function of ζ^t taking values in \mathbf{R}^ν is fully specified by the *collection of its coefficients*

$$g^t = \{g_{\tau\varkappa} \in \mathbf{R}^\nu : 1 \leq \tau \leq t, \varkappa \in \mathcal{I}_\tau\}$$

according to

$$g(\zeta^t) = \sum_{\tau=1}^t \sum_{\varkappa \in \mathcal{I}_\tau} \left[\prod_{s=1}^{\mu} \lambda_{\varkappa_s}(\zeta_{\tau-\mu+s}) \right] g_{\tau\varkappa}. \quad (3.13)$$

Note that every collection $\{g_{\tau\varkappa} \in \mathbf{R}^\nu : 1 \leq \tau \leq t, \varkappa \in \mathcal{I}_\tau\}$ is a collection of coefficients of a poly-affine with memory μ function $g(\zeta^t)$ taking values in \mathbf{R}^ν .

Structural restrictions imposed in the sequel on the decision rules and right-hand sides in the constraints (3.12) are as follows (cf. **A.3-4**):

B.3. The right-hand sides $b_t(\zeta^t)$ in design specifications (3.12) are poly-affine with memory μ .

B.4. The decision rules $x_t(\zeta^t)$ are restricted to be poly-affine with memory μ .

By **B.4**, candidate decision rules $x^N = \{x_t(\cdot) : 1 \leq t \leq N\}$ in question can be parameterized by finite-dimensional collections

$$v^N = \{v_{\tau\kappa}^t \in \mathbf{R}^{n_t} : 1 \leq \tau \leq t \leq N, \kappa \in \mathcal{I}_\tau\} \quad (3.14)$$

according to

$$x_t(\zeta^t) = \sum_{\tau=1}^t \sum_{\kappa \in \mathcal{I}_\tau} \left[\prod_{s=1}^{\mu} \lambda_{\kappa_s}(\zeta_{\tau-\mu+s}) \right] v_{\tau\kappa}^t. \quad (3.15)$$

Whatever be a selection of vectors $v_{\tau\kappa}^t \in \mathbf{R}^{n_t}$ in (3.14), the resulting collection specifies candidate decision rules $x_t(\cdot)$, $t \leq N$, satisfying **B.4**.

Finally, we make the following assumption (cf. **A.5**):

B.5. The objective to be minimized is an efficiently computable convex function $f(v^N)$ of the vector v^N of parameters of a candidate decision rule.

3.2 Processing the problem

Let $D_t = \{1, \dots, d_t\}$ be the set of indices of scenarios χ_{ts} specifying Δ_t according to (3.11). Every “trajectory of indices” $\xi^t \in D^t := D_1 \times \dots \times D_t$ specifies the trajectory of disturbances

$$\zeta^t[\xi^t] = \{\zeta_\tau[\xi_\tau] := \chi_{\tau\xi_\tau} : 1 \leq \tau \leq t\} \in \Delta^t := \Delta_1 \times \dots \times \Delta_t.$$

Clearly, when $\xi^\tau \in D^\tau$ is the initial fragment of $\xi^t \in D^t$, then $\zeta^\tau[\xi^\tau]$ is the initial fragment of $\zeta^t[\xi^t]$. Let us make two immediate observations:

Lemma 3.1 *Let $g(\zeta^t)$ be a poly-affine, with memory μ , function taking values in \mathbf{R}^ν :*

$$g(\zeta^t) = \sum_{\tau=1}^t \sum_{\kappa \in \mathcal{I}_\tau} \left[\prod_{s=1}^{\mu} \lambda_{\kappa_s}(\zeta_{\tau-\mu+s}) \right] g_{\tau\kappa}. \quad (3.16)$$

Then the mapping

$$\bar{g}(\xi^t) := g(\zeta^t[\xi^t]) : D^t \rightarrow \mathbf{R}^\nu$$

is additive with memory μ .

Proof. Setting, as in Section 2, $D_{p:q} = D_p \times D_{p+1} \times \dots \times D_q$ and given t, τ, ξ with $1 \leq \tau \leq t \leq N$ and $\xi = (\xi_{\tau-\mu+1}, \xi_{\tau-\mu+2}, \dots, \xi_\tau) \in D_{\tau-\mu+1:\tau}$, let us put

$$\bar{g}_{\tau\xi} = \sum_{\varkappa \in \mathcal{I}_\tau} \left[\prod_{s=1}^{\mu} \lambda_{\varkappa_s}(\chi_{\tau-\mu+s, \xi_{\tau-\mu+s}}) \right] g_{\tau\varkappa}$$

(the right-hand side indeed depends only on τ and $(\xi_{\tau-\mu+1}, \xi_{\tau-\mu+2}, \dots, \xi_\tau) \in D_{\tau-\mu+1:\tau}$). It remains to note that by (3.16) we have

$$\bar{g}(\xi^t) = g(\chi_{1\xi_1}, \chi_{2\xi_2}, \dots, \chi_{t\xi_t}) = \sum_{\tau=1}^t \bar{g}_{\tau\xi_{\tau-\mu+1:\tau}} \quad \forall \xi^t \in D^t = D_1 \times \dots \times D_t,$$

as it should be for an additive with memory μ function. \square

Our second observation is as follows:

Lemma 3.2 *Let $f(\zeta^N)$ be a real-valued affine in every component ζ_t of ζ^N function (e.g., a poly-affine real-valued function with memory μ). Then among maximizers of $f(\zeta^N)$ over $\zeta^N \in \Delta^N$ there are those of the form $\zeta[\xi^N]$ with properly selected $\xi^N \in D^N$.*

Proof. Let $\bar{\zeta}^N$ be a maximizer of $f(\zeta^N)$ on Δ^N with the largest possible number, let it be M , of scenario components $\bar{\zeta}_t$ (i.e., those belonging to $\{\chi_{ts}, 1 \leq s \leq d_t\}$). All we need to prove is that $M = N$. This is evident: assuming that $M < N$, i.e., that for some t the component $\bar{\zeta}_t$ of $\bar{\zeta}^N$ is not in the set $\{\chi_{ts}, 1 \leq s \leq d_t\}$, let us “freeze” in $f(\zeta^N)$ all arguments ζ_s with $s \neq t$ at the values $\bar{\zeta}_s$ and vary the t -th argument. Since f is affine in every ζ_s , among the maximizers of the resulting function of ζ_t over $\zeta_t \in \Delta_t$ there will be an extreme point of Δ_t , that is, a point from the set of scenarios of stage t . Replacing in $\bar{\zeta}^N$ the component $\bar{\zeta}_t$ with this scenario, we get another maximizer of f on Δ^N with more than M scenario components, which is impossible. \square

Now we are ready to explain how to process the problem of interest numerically. Let us associate with our problem (call it continuous) a *discrete* problem as follows. The structure of the discrete problem is as considered in Section 2, with inherited from the continuous problem number of stages N , matrices $A^{t\tau}$, and cardinalities d_t of the sets D_t of values of disturbance ξ_t at stage t . As about the right-hand sides $\bar{b}_t(\xi^t)$ in the constraints of the discrete problem, we specify them as

$$\bar{b}_t(\xi^t) = b_t(\zeta^t[\xi^t]).$$

Now, candidate decision rules $x_t(\zeta^t)$, $t \leq N$, in the continuous problem induce candidate decision rules

$$\bar{x}_t(\xi^t) := x_t(\zeta^t[\xi^t])$$

in the discrete problem. By Lemma 3.1, restrictions **B.3-4** on the structure of $b_t(\cdot)$'s and $x_t(\cdot)$'s ensure the validity of **A.3-4** for $\bar{b}_t(\cdot)$'s and $\bar{x}_t(\cdot)$'s. Besides this, Lemma 3.2 says that under restrictions **B.3-4** decision rules $x_t(\cdot)$ are feasible for the continuous problem if and only if the decision rules $\bar{x}_t(\cdot)$ are feasible for the discrete problem. As we remember, the latter is equivalent to the fact that the collection

$$u^N = \{u_{\tau\xi}^t : 1 \leq \tau \leq t \leq N, \xi \in D_{\tau-\mu+1:\tau}\}$$

of parameters of the additive, with memory μ , decision rules \bar{x}^N can be augmented by properly selected y - and z -variables to yield a feasible solution to certain system \mathcal{S} of linear constraints. For μ fixed, the number of constraints and variables in \mathcal{S} , same as the computational effort to build this system, is polynomial in all sizes of the problem (for details, see Section 2). We are in the situation where u^N is obtained from the “primitive” design variables, specifically, the collection v^N of parameters specifying the decision rules $x_t(\cdot)$ by known to us linear transformation:

$$\begin{aligned} u_{\tau\xi}^t &= \sum_{\kappa \in \mathcal{I}_\tau} \left[\prod_{s=1}^{\mu} \lambda_{\kappa_s}(\chi_{\tau-\mu+s, \xi_{\tau-\mu+s}}) \right] v_{\tau\kappa}^t, \\ \text{for } 1 \leq \tau \leq t, \xi &= (\xi_{\tau-\mu+1}, \xi_{\tau-\mu+2}, \dots, \xi_\tau) \in D_{\tau-\mu+1:\tau}. \end{aligned} \quad (3.17)$$

Extending \mathcal{S} to the system of linear constraints by adding variables v^N and constraints (3.17) linking the v - and the u -variables, we get a system \mathcal{S}^+ of linear constraints in “actual” design variables v^N and additional analysis variables (specifically, u^N and y - and z -variables inherited from \mathcal{S}). The bottom line is that under Assumption **B.5**, for μ fixed, the problem of interest can be reduced to the problem of minimizing an efficiently computable convex function of v -variables under a system \mathcal{S}^+ of linear constraints on v, u, y, z -variables, with the total number of variables and constraints in \mathcal{S}^+ and the computational effort of building this system which are polynomial in the sizes of the problem of interest.

Note that, similarly to the case of discrete uncertainty, an objective which is the worst-case value over all trajectories ζ^N of a linear functional $\sum_{t=1}^N \eta_t^T x_t(\xi^t)$ of the control trajectory satisfies **B.5**, cf. the beginning of Section 2.3. Furthermore, in the case of random disturbances, an objective of the form

$$f(v^N) = \mathbf{E}_{\zeta^N \sim P} \{F(x^N(\zeta^N), \zeta^N)\}$$

with function $F(x^N, \zeta^N)$ which is convex in decision variables $x^N = [x_1; x_2; \dots; x_N]$ and efficiently computable, can be replaced by its Sample Average Approximation

$$\hat{f}_N(v^N) = \frac{1}{L} \sum_{\ell=1}^L F(x^N(\zeta^{N\ell}), \zeta^{N\ell})$$

over a large number L of scenarios — realizations of disturbance trajectories $\zeta^{N\ell} = (\zeta_1^\ell, \zeta_2^\ell, \dots, \zeta_N^\ell)$, $\ell = 1, \dots, L$. In this case, Assumption **B.5** holds for the approximate objective $\hat{f}_N(v^N)$ with decision

$x_t(\zeta^{t\ell})$ at stage t of scenario ℓ linked to v^N -variables by the explicit linear relation

$$x_t(\zeta^{t\ell}) = \sum_{\tau=1}^t \sum_{\varkappa \in \mathcal{I}_\tau} \left[\prod_{s=1}^{\mu} \lambda_{\varkappa_s}(\zeta_{\tau-\mu+s}^\ell) \right] v_{\tau\varkappa}^t,$$

see (3.15).

4 An application to hydro-thermal production planning

In this section, we illustrate the application of our methodology on a toy example of a hydro-thermal production planning problem formulated as a Multistage Stochastic Linear Program with linear constraints.

4.1 Problem description

Our problem modeling is as follows. Consider a set of thermal electricity production plants and hydroelectric plants distributed in K regions which have to produce electricity to satisfy the demand in each region and each time step $t = 1, \dots, N$ of a given planning horizon. We will assume that in each region, all thermal facilities are aggregated into a single thermal plant, and similarly, all hydroelectric plants and reservoirs are aggregated into a single hydroelectric plant and a single reservoir. The objective is to minimize the production cost which is a sum of the cost of thermal generation and the penalties paid for the unsatisfied demand over the planning horizon under constraints of demand satisfaction, minimal and maximal levels of the hydroelectric reservoirs and capacity constraints of the production units.

We use the following notation for time $t = 1, \dots, N$:

- $v_t \in \mathbf{R}^K$ for reservoir levels at the end of stage t ;
- $w_t \in \mathbf{R}^K$ for thermal generation at stage t ;
- $h_t \in \mathbf{R}^K$ for hydroelectric generation;
- $r_t \in \mathbf{R}^K$ for unsatisfied demand;
- $\mathcal{I}_t \in \mathbf{R}^K$ for inflows;
- $G_t \in \mathbf{R}^{K \times K}$ is a diagonal matrix; $G_t \mathcal{I}_t$ is the vector of actual inflows to the reservoirs and $(I - G_t) \mathcal{I}_t$ is the part of inflows automatically converted into energy by run-of-river plants;
- $\delta_t \in \mathbf{R}^K$ is the deterministic vector of energy demands;
- $c_t \in \mathbf{R}^K$ is the vector of thermal generation unit costs, and $p_t \in \mathbf{R}^K$ is the vector of penalties for the unsatisfied demand at time t .

The hydrothermal production planning problem consists in minimizing the expected cost

$$\mathbf{E}_{\mathcal{I}^N} \left[\sum_{t=1}^N c_t^T w_t + \sum_{t=1}^N p_t^T r_t \right],$$

under the following system of constraints to be satisfied almost surely:

$$\begin{aligned} h_t &\leq v_{t-1} - v_t + G_t \mathcal{I}_t, & [\text{water balance}] \\ h_t + w_t + r_t &\geq \delta_t - (I - G_t) \mathcal{I}_t, & [\text{demand satisfaction}] \\ v_t &\leq v_t \leq \bar{v}_t, & [\text{bounds on reservoir levels}] \\ 0 &\leq h_t \leq \bar{h}_t, & [\text{hydroelectric generation capacity}] \\ 0 &\leq w_t \leq \bar{w}_t, & [\text{thermal generation capacity}] \\ r_t &\geq 0. & [\text{nonnegativity of unsatisfied demand}] \end{aligned} \tag{4.18}$$

We assume that inflows \mathcal{I}_t , $t \leq 1$ are deterministic (i.e., ..., \mathcal{I}_0 , \mathcal{I}_1 are known at $t = 1$ when production plan is computed for the horizon $t = 1, \dots, N$) and for $t \geq 2$ inflows satisfy the periodic autoregressive model:

$$\begin{aligned} \mathcal{I}_t &= \theta_t + \eta_t, \\ \eta_t &= \sum_{j=1}^{\ell_t} B_t^j \eta_{t-j} + C_t \zeta_t \end{aligned} \tag{4.19}$$

where $\theta_t \in \mathbf{R}^K$ are given along with $K \times K$ diagonal matrices B_t^j , C_t ; while innovations ζ_t , $t = 2, \dots, N$, are independent with known distribution P_t over the support set $\{\chi_{t1}, \dots, \chi_{td_t}\}$ of ζ_t .

Note that inflows \mathcal{I}_t satisfying recursive equations (4.19) can be straightforwardly rewritten in the form

$$\mathcal{I}_t = \nu_t + \sum_{s=2}^t R_s^t \zeta_s, \tag{4.20}$$

with deterministic $\nu_t \in \mathbf{R}^K$ and $R_s^t \in \mathbf{R}^{K \times K}$, implying that \mathcal{I}_t for $t \geq 2$ is an affine function of ζ_1, \dots, ζ_t given by (4.20); we denote it $\mathcal{I}_t(\zeta^t)$. We allow decisions v_t, w_t, h_t , and r_t to depend on $\mathcal{I}^t := (\mathcal{I}_1, \dots, \mathcal{I}_t)$. Therefore, decision vector $\bar{x}_t = [h_t; v_t; r_t; w_t] \in \mathbf{R}^K \times \mathbf{R}^K \times \mathbf{R}^K \times \mathbf{R}^K$ at stage t is a function $\bar{x}_t(\zeta^t)$ of innovations (ζ_t) up to time t , so that system (4.18) of problem constraints can be written as:

$$\sum_{\tau=t-1}^t A^{t\tau} \bar{x}_\tau(\zeta^\tau) \leq \bar{b}_t(\zeta^t) \in \mathbf{R}^{m_t}, \quad t = 1, \dots, N, \tag{4.21}$$

where $A^{t,t}$ and $A^{t,t-1}$ are given matrices and the right-hand side $\bar{b}_t(\zeta^t)$ is a linear function of ζ^t .

Now, to apply the methodology of Section 2 it suffices to reformulate the problem in terms of disturbances ξ_t taking values in finite sets of integers $D_t = \{1, 2, \dots, d_t\}$ of cardinality d_t with innovations $\zeta^t[\xi^t] = (\chi_{1\xi_1}, \chi_{2\xi_2}, \dots, \chi_{t\xi_t})$ and controls $x_t(\xi^t) := \bar{x}_t(\chi_{1\xi_1}, \chi_{2\xi_2}, \dots, \chi_{t\xi_t})$ thus replacing

constraints (4.21) with

$$\sum_{\tau=t-1}^t A^{t\tau} x_{\tau}(\xi^{\tau}) \leq b_t(\xi^t) := \bar{b}_t(\zeta^t[\xi^t]) \in \mathbf{R}^{m_t}, \quad t = 1, \dots, N, \quad \forall \xi^t \in D_1 \times \dots, D_t, \quad (4.22)$$

which is clearly of form (2.1). Observe that Assumptions **A.1.** and **A.2** clearly hold for the reformulated system. It is also easily seen that Assumption **A.3.** holds true for the right-hand side b_t in (4.22) which is an additive with memory $\mu = 1$ function $b_t(\xi^t) = \bar{b}_t(\zeta^t[\xi^t])$ with $\bar{b}_t(\cdot)$ linear in ζ^t . Finally, let us assume that Assumption **A.4.** holds for the decision rules x_t which are restricted to be additive with memory μ functions of ξ^t , and let us denote

$$u^N = \{u_{\tau\xi}^t \in \mathbf{R}^{n_t} : 1 \leq \tau \leq t \leq N, \xi \in D_{\tau-\mu+1:\tau}\}$$

parameter collections in the representation

$$x_t(\xi^t) = \sum_{\tau=1}^t u_{\tau\xi_{\tau-\mu+1:\tau}}^t$$

of candidate decision rules. Note that because the problem objective is linear in x^N Assumption **A.5.** obviously holds. Moreover, when the discrete distribution P_t of ζ_t (and thus distribution of ξ_t) is known the objective $f(u^N)$ to be minimized in order to compute optimal constant depth decision rules is known in closed-form. Specifically, denoting $f_t^T x_t(\xi^t)$ the cost per stage t we have

$$f(u^N) = \sum_{t=1}^N f_t^T \sum_{s=1}^t \sum_{\xi_{s-\mu+1:s} \in D_{s-\mu+1:s}} \left(\prod_{r=s-\mu+1}^s P_r(\zeta_r = \chi_r \xi_r) \right) u_{s\xi_{s-\mu+1:s}}^t.$$

4.2 Numerical experiments: comparing CDDRs and SDDP

Numerical simulations described in this section utilize a MATLAB library for computing optimal Constant Depth Decision Rules for Multistage Stochastic Linear Programs with constraints of the form (2.1) satisfying the conditions in Section 2 with known distribution of perturbations ξ^N .¹

We compare CDDR and Stochastic Dual Dynamic Programming (SDDP, see, e.g., [20, 24, 14] and references therein) solutions on an instance of the hydro-thermal production planning problem with interstage dependent inflows described in the previous section. Parameters of the problem are initialized to mimic the mid-term Brazilian hydro-thermal problem with $K = 4$ equivalent subsystems for the country considered for instance in [9] and [14]. For the sake of simplicity we considered only one equivalent thermal plant per subsystem (as described in the previous section)

¹The functions in the library allow to load the linear program whose solutions are optimal CDDRs and solves it using Mosek [1] solver. The library is available at https://github.com/vguigues/Constant_Depth_Decision_Rules_Library. A function to run the simulations of this section is also provided, together with the implementation of SDDP for the considered hydro-thermal application.

N	d_t	SDDP LB	SDDP UB	CDDR $\mu = 1$	CDDR $\mu = 2$	CDDR $\mu = 3$	CDDR $\mu = 4$
6	6	2.02	2.10	2.70	2.30	2.17	2.07
6	10	2.31	2.43	2.91	2.91	2.43	2.37
6	20	2.04	2.15	2.71	2.29	2.17	-
6	40	1.69	1.77	2.30	1.94	-	-
12	6	4.92	5.12	7.01	5.90	5.41	-
12	10	4.52	4.74	7.41	5.90	5.41	-
12	20	5.22	5.48	8.17	6.38	-	-
12	40	4.18	4.40	7.34	5.71	-	-

Table 1: Hydro-thermal production planning example: optimal expected cost using CDDR policies and lower and upper bound at the last iteration of SDDP. All costs have been divided by 10^6 to improve readability.

N	d_t	SDDP	CDDR $\mu = 1$	CDDR $\mu = 2$	CDDR $\mu = 3$	CDDR $\mu = 4$
6	6	26.1	0.14	1.1	4.8	14.0
6	10	37.8	0.6	3.1	22.7	111.2
6	20	970	0.58	14.2	349.8	-
6	40	3 384	1.07	65.5	-	-
12	6	1877	1.1	10.3	144.3	-
12	10	1920	2.1	27.8	2 821.2	-
12	20	5448	3.45	268.9	-	-
12	40	15 147	7.5	948.5	-	-

Table 2: Hydro-thermal production planning example: CPU times (in seconds) to compute SDDP and CDDR policies.

an increased demand; when compared to the original setting of [14] we allow less water in reservoirs at the first stage and fix the standard deviation of components of perturbations ζ_t to be 0.2 (for detailed description of the set of parameters used in the simulation, see the library description).

We consider 8 instances with $N \in \{6, 12\}$ (each stage representing a month) and $d_t = d \in \{6, 10, 20, 40\}$ (the number of realizations d is the same for every stage).

In our experiments the size of the state vector of SDDP² varies from 14 to 20, making the Dynamic Programming equations challenging to solve.

For each instance, we run SDDP and CDDRs with $\mu = 1, 2, 3$, and 4. We stop SDDP algorithm when the gap between the upper and lower bounds becomes less than 5%.

The results of this experiment are presented in Table 1; for all instances we report the optimal expected costs obtained using CDDRs along with lower bound SDDP LB and upper bound SDDP UB at the last iteration of SDDP. CPU times to compute these policies are given in Table 2 (we report results on instances in which Mosek was able to solve the corresponding LP); some typical sizes of the deterministic LPs to be solved to compute the optimal CDDRs are represented in Figure 1.

Our experiments highlight some strengths and drawbacks of Constant Depth Decision Rules when applied to a Multistage Stochastic Program. Compared to linear decision rules which are

²Here, the state vector stores the reservoir levels at the end of the stage and the relevant history of inflows. Since lags of the inflow model are time dependent, this history and therefore the size of the state vector depends on time.

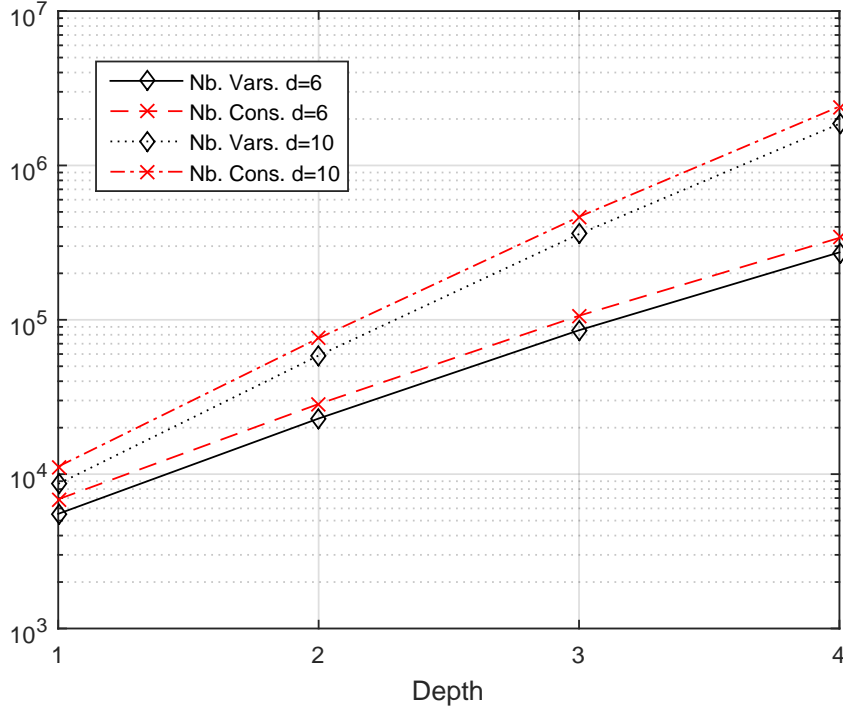


Figure 1: Number of constraints (Nb. Cons.) and variables (Nb. Vars.) of the LPs solved to compute CDDRs for the hydro-thermal production planning application for $d = 6, 10$, as a function of the depth μ for $N = 6$.

suboptimal in this application (their cost cannot be less than the cost of an optimal CDDR with $\mu = 1$), CDDRs are more flexible and allow for a larger class of policies. Notice that the optimal cost of CDDRs decreases with μ and becomes close to the lower and upper bounds computed at the last iteration of SDDP when μ is large enough. Although “short memory” CDDRs (with $\mu = 1$ on $\mu = 2$) are suboptimal in this application, they are computed much faster than the corresponding SDDP policies, and can be used to initiate SDDP cuts.

In the problem under consideration, a relatively complete recourse is available, which is necessary for application of the SDDP. Note that utilizing CDDRs does not require this assumption. Furthermore, unlike the SDDP, CDDRs can handle problems with any interstage dependent discrete stochastic process; it also allows using continuous modeling of disturbances with distributions supported on polytopes. When SDDP can be applied, we can list the following advantages of CDDRs over SDDP. When applied to a problem with linear constraints with random right-hand side noises satisfying autoregressive equation the time to solve the problem increasing exponentially fast with the size of the state vector of the SDDP which is proportional to the maximal lag of the autoregression. On the contrary, the size of the problem to solve to compute optimal CDDRs is independent on the value of the lag. Furthermore, the numerical cost of computing SDDP policies depends on the variance of random disturbances — the larger the variance, the higher the cost. For CDDRs, noise variance has no influence on computation times.

As a rule of thumb, it is recommended to use CDDRs with very moderate depth μ (between 1 and 4) when the maximal number d of possible realizations of the uncertainty for each stage is also a moderate integer and one of the above conditions applies (for instance, for problems with interstage dependent disturbances with large lags). The principal drawback of the CDDRs is that the size of the optimization linear problem to be solved to compute the optimal rules grows exponentially with μ becoming prohibitive already for some toy problems considered in this section. As a future work, decomposition techniques could be investigated to allow using CDDRs for problems with larger values of memory depth.

Acknowledgments

Research of Vincent Guigues was supported by an FGV grant, CNPq grants 204872/2018-9 and 401371/2014-0, FAPERJ grant E-26/201.599/2014. Research of Anatoli Juditsky was supported by MIAI Grenoble Alpes (ANR-19-P3IA-0003), CNPq grant 401371/2014-0 and NSF grant CCF-1523768. Research of Arkadi Nemirovski was supported by NSF grants CCF-1523768, CCF-1415498, and CNPq grant 401371/2014-0.

References

- [1] E. D. Andersen and K. Andersen. *The MOSEK optimization toolbox for MATLAB manual. Version 9.2*, 2019. <https://www.mosek.com/documentation/>.
- [2] D. Bampou and D. Kuhn. Scenario-free stochastic programming with polynomial decision rules. *IEEE Conference on Decision and Control and European Control Conference*, pages 351–376, 2011.
- [3] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.
- [4] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust counterpart of uncertain linear programs. *Mathematical Programming*, 99:351–376, 2003.
- [5] J. Birge. Decomposition and partitioning methods for Multistage Stochastic Linear Programs. *Oper. Res.*, 33:989–1007, 1985.
- [6] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer-Verlag, New York, 1997.
- [7] A. Charnes and W. W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations research*, 11(1):18–39, 1963.
- [8] A. Charnes, W. W. Cooper, and G. H. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management science*, 4(3):235–263, 1958.

- [9] L. Ding, S. Ahmed, and A. Shapiro. Python package for Multi-stage Stochastic Programming. *Optimization Online*, 2019.
- [10] S. Garstka and R. J. Wets. On decision rules in Stochastic Programming. *Mathematical Programming*, 7:117–143, 1974.
- [11] A. Georghiou, W. Wiesemann, and D. Kuhn. Generalized decision rule approximations for Stochastic Programming via liftings. *Mathematical Programming*, 152:301–338, 2015.
- [12] P. Girardeau, V. Leclere, and A. Philpott. On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, 40:130–145, 2015.
- [13] V. Guigues. Robust production management. *Optimization and Engineering*, 10:505–532, 2009.
- [14] V. Guigues. SDDP for some interstage dependent risk-averse problems and application to hydro-thermal planning. *Computational Optimization and Applications*, 57:167–203, 2014.
- [15] V. Guigues. Convergence analysis of sampling-based decomposition methods for risk-averse multistage stochastic convex programs. *SIAM Journal on Optimization*, 26:2468–2494, 2016.
- [16] V. Guigues. Inexact cuts in Stochastic Dual Dynamic Programming. *SIAM Journal on Optimization*, 30:407–438, 2020.
- [17] V. Guigues and M. Bandarra. Single cut and multicut SDDP with cut selection for multistage stochastic linear programs: convergence proof and numerical experiments. Available at <https://arxiv.org/abs/1902.06757>, 2019.
- [18] V. Guigues and R. Henrion. Joint dynamic probabilistic constraints with projected linear decision rules. *Optimization Methods & Software*, 32:1006–1032, 2017.
- [19] A. Lorca, X. Sun, E. Litvinov, and T. Zheng. Multistage adaptive robust optimization for the unit commitment problem. *Operations Research*, 64:32–51, 2016.
- [20] M. Pereira and L. Pinto. Multi-stage Stochastic Optimization applied to energy planning. *Math. Program.*, 52:359–375, 1991.
- [21] A. B. Philpott and Z. Guan. On the convergence of Stochastic Dual Dynamic Programming and related methods. *Operations Research Letters*, 36:450–455, 2008.
- [22] W. Powell. *Approximate Dynamic Programming*. John Wiley and Sons, 2nd edition, 2011.
- [23] P. Rocha and D. Kuhn. Multistage stochastic portfolio optimisation in deregulated electricity markets using linear decision rules. *European Journal of Operational Research*, 216:397–408, 2012.

- [24] A. Shapiro. Analysis of Stochastic Dual Dynamic Programming method. *European Journal of Operational Research*, 209:63–72, 2011.
- [25] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, Philadelphia, 2009.
- [26] A. Shapiro and A. Nemirovski. On complexity of Stochastic Programming problems. *in Continuous Optimization: Current Trends and Applications, V. Jeyakumar and A.M. Rubinov (Eds.), Springer*, pages 111–144, 2005.