DUALITY AND SENSITIVITY ANALYSIS OF MULTISTAGE LINEAR STOCHASTIC PROGRAMS
VINCENT GUIGUES∗, ALEXANDER SHAPIRO†, AND YI CHENG ‡

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

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

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

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

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

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

1. Introduction. Duality plays a key role in optimization. For generic optimization problems, weak duality allows to bound the optimal value. Dual information is also used in many optimization algorithms such as the Uzawa algorithm [2], primal-dual projected gradient [19] or Stochastic Dual Dynamic Programming (SDDP) [20]. Moreover, for several classes of optimization problems, the dual is easier to solve than the primal problem, for instance when it is amenable to decomposition techniques such as price decomposition [4]. Even when there is a duality gap between the primal and dual optimal values, solving the dual already gives a bound on the optimal value, as mentioned earlier. Duality is also a fundamental tool in the reformulation of Robust Optimization problems, see for instance [3]. Finally, derivatives of the value function of classes of optimization problems can be related to optimal dual solutions, see [5], [22] and more recently [8, 10, 11] for the characterization of subdifferentials, subgradients, and ε-subgradients of value functions of convex optimization problems.

For stochastic control problems, stochastic Lagrange multipliers were already used in [14, 15, 16]. In the context of multistage stochastic programs, duality was studied in [23, 13], see also [25] for a review. More recently, the sensitivity analysis of multistage stochastic programs was discussed in [6] and [27]. In [6] the authors study the sensitivity with respect to parameters driving the considered price model. The corresponding parameters are in the objective function and the analysis of the estimate of marginal price is based on Danskin’s theorem with the SDDP method used for the numerical calculations. In [27], the authors use the Envelope Theorem for the sensitivity analysis. The required derivatives are described in terms of Lagrange multipliers associated with the value functions.

∗ School of Applied Mathematics, Fundação Getulio Vargas, 190 Praia de Botafogo, Rio de Janeiro, Brazil, (vincent.guigues@fgv.br). Research of this author was partially supported by CNPq grants 04872/2018-9 and 311289/2016-9.
† School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA, (ashapiro@isye.gatech.edu). Research of this author was partly supported by NSF grant 1633196.
‡ School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA, (cheng.yi@gatech.edu).
In this paper, focusing our attention on the dual of a Multistage Stochastic Linear Program (MSLP), we are able to provide insights into three important problems for MSLPs: sensitivity analysis, computation of a sequence of deterministic upper bounds on the optimal value which converges to the optimal value, and use of duality to solve Dynamic Programming (DP) equations on the dual which are simpler to solve (in the sense that they have convex cost-to-go functions) than primal DP equations for problems with interstage dependent cost coefficients. Our main contributions are summarized below.

**Sensitivity analysis of MSLPs.** We explain how to compute derivatives of the optimal value, seen as a function of the problem parameters, of a MSLP in terms of dual optimal solutions. Therefore, the construction of the dual problem is essential for our approach, contrary to [6]. With respect to the sensitivity analysis [27], in our approach, we do not use value functions directly, which are not known and can only be approximated, but rather construct the dual problem which is solved by an SDDP type algorithm, called Dual SDDP.

**Writing Dynamic Programming equations for the dual problem.** A simple but crucial ingredient for our developments and subsequent analysis of solution methods for the dual problem of a MSLP is to write DP equations for that dual problem. We are not aware of another paper with these equations. However, a similar study was done in [17]. More precisely, for a stochastic linear control problem with uncertainty in the right-hand-side, in [17], DP equations are written for the conjugate of the cost-to-go functions and using an SDDP type method for these DP equations, a sequence of upper bounds on the MSP optimal value is constructed which is the sequence of conjugate of the approximate first stage cost-to-go functions evaluated at the initial state \( x_0 \). Our approach has the advantage of being much simpler: contrary to derivations in [17] which require some algebra, our DP equations can be immediately obtained from the dual problem formulation, this latter being known (given in [25] for instance). On top of that, we relax two assumptions made in [17]: (a) the relatively complete recourse assumption of the dual and (b) randomness in the right-hand-side of the constraints only and interstage independent. The next three paragraphs describe how the scope of (a) and (b) was extended in our analysis.

**Dual SDDP for dual problems without relatively complete recourse.** In [17], it is assumed that the dual problem of the considered MSLP satisfies an assumption (Assumption 3) stronger than relatively complete recourse. This assumption may not be easy to check or may not be satisfied (for instance it is not satisfied for the inventory and hydro-thermal problems considered in Section 5). Therefore, it is desirable to extend the scope of Dual SDDP in such a way that it can still compute a deterministic converging sequence of upper bounds without this assumption. We present two variants of Dual SDDP that can do that: Dual SDDP with penalizations and Dual SDDP with feasibility cuts.

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

**Dual SDDP for problems with interstage dependent cost coefficients.** Finally, we also relax assumption (b) considering problems having interstage dependent cost coefficients. Writing DP equations for the corresponding dual problem, we give the Dual SDDP algorithm to solve these equations, which, interestingly, have concave cost-to-go functions whereas primal cost-to-go functions are not convex. This is in sharp contrast with the solution methods proposed so far such as [6, 18] which apply SDDP on the primal cost-to-go functions using a Markov chain approximation of the cost coefficients process.

The outline of the paper is the following. Our building blocks are elaborated in Section 2 where we write DP equations for the dual, we explain how to build upper bounding functions for the cost-to-go functions of the dual using penalizations, and study the dynamics of Lagrange multipliers. Sensitivity analysis of MSLPs is conducted in Section 3 while Dual SDDP and its variants are studied in Section 4. Finally, the results of numerical simulations testing the tools developed on an inventory and an hydro-thermal problem are presented in Section 5. The interested reader can find and test the code of all implementations and of Primal and Dual SDDP for MSLPs (4.15) given below at https://github.com/vguigues/Dual_SDDP_Library_Matlab and https://github.com/vguigues/Primal_SDDP_Library_Matlab. Proofs are collected in the Appendix.
2. Duality of multistage linear stochastic programs.

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

\[
\begin{align*}
\min_{x_t \geq 0} & \quad E \left[ \sum_{t=1}^{T} c_t^\top x_t \right] \\
\text{s.t.} & \quad A_1 x_1 = b_1, \\
& \quad B_t x_{t-1} + A_t x_t = b_t, \quad t = 2, ..., T.
\end{align*}
\]

(2.1)

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

The Lagrangian of problem (2.1) is

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

(2.2)

in variables \( x = (x_1(\xi_{[1]}), ..., x_T(\xi_{[T]})) \) and \( \pi = (\pi_1(\xi_{[1]}), ..., \pi_T(\xi_{[T]})) \) with the convention that \( x_0 = 0 \). Dualization of the feasibility constraints leads to the following dual of problem (2.1) (cf., [25, Section 3.2.3]):

\[
\begin{align*}
\max_{\pi} & \quad E \left[ \sum_{t=1}^{T} b_t^\top \pi_t \right] \\
\text{s.t.} & \quad A_1^\top \pi_T \leq c_T, \\
& \quad A_{t-1}^\top \pi_{t-1} + E_{[\xi_{[t-1]}]} [B_t^\top \pi_t] \leq c_{t-1}, \quad t = 2, ..., T.
\end{align*}
\]

(2.3)

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

Assuming that the random process \( \xi_t, t = 1, ..., T \), has a finite number of realizations (scenarios), problem (2.1) can be viewed as a large linear program and (2.3) as its dual. Assuming further that problem (2.1) has a finite optimal value, it follows by the standard theory of linear programming that the optimal values of problems (2.1) and (2.3) are equal to each other and both problems have optimal solutions. Unless stated otherwise, we make the following assumption throughout the paper.

(A1) The process \( \xi_1, ..., \xi_T \) is stagewise independent (i.e., random vector \( \xi_{t+1} \) is independent of \( \xi_{[t]}, \quad t = 1, ..., T - 1 \), and distribution of \( \xi_t \) has a finite support, \( \{\xi_{t_1}, ..., \xi_{t_{N_t}}\} \) with respective probabilities \( p_{t_j}, \quad j = 1, ..., N_t, \quad t = 2, ..., T \). We denote by \( A_{t_j}, B_{t_j}, c_{t_j}, b_{t_j} \) the respective scenarios corresponding to \( \xi_{t_j} \).

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

\[
\begin{align*}
\max_{\pi_T} & \quad E \left[ b_T^\top \pi_T \right] \\
\text{s.t.} & \quad A_T^\top \pi_T \leq c_T, \\
& \quad A_{T-1}^\top \pi_{T-1} + E [B_T^\top \pi_T] \leq c_{T-1}.
\end{align*}
\]

(2.4)

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

\[
\begin{align*}
\max_{\pi_{T_1}, ..., \pi_{T_N}} & \quad \sum_{j=1}^{N_T} p_{T_j} b_{T_j}^\top \pi_{T_j} \\
\text{s.t.} & \quad A_{T_j}^\top \pi_{T_j} \leq c_{T_j}, \quad j = 1, ..., N_T, \\
& \quad A_{T-1}^\top \pi_{T-1} + \sum_{j=1}^{N_T} p_{T_j} B_{T_j}^\top \pi_{T_j} \leq c_{T-1}.
\end{align*}
\]

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

\[
\begin{align*}
(2.6) \quad \max_{\pi_1, \ldots, \pi_{TN_t}} & \sum_{j=1}^{N_t} p_{tj} \left[ b_{tj}^T \pi_t + V_{t+1}(\pi_{tj}, \xi_{tj}) \right] \\
\text{s.t.} & \quad A_{t-1}^T \pi_{t-1} + \sum_{j=1}^{N_t} p_{tj} B_{tj}^T \pi_j \leq c_{t-1},
\end{align*}
\]

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

\[
(2.7) \quad \max_{\pi_1} b_1^T \pi_1 + V_2(\pi_1, \xi_1).
\]

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

\[
\begin{align*}
(2.8) \quad \min_{x_{t} \geq 0} & \quad c_{tj}^T x_t + Q_{t+1}(x_t) \\
\text{s.t.} & \quad B_{tj} x_{t-1} + A_{tj} x_t = b_{tj},
\end{align*}
\]

with

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

Unlike in the primal problem, the optimization (maximization) problems (2.5) and (2.6) do not decompose into separate problems with respect to each \(\pi_{tj}\) and should be solved as one linear program with respect to \((\pi_1, \ldots, \pi_{TN_t})\).

Note that the value function \(V_t(\pi_{t-1}, \xi_{t-1})\) is a concave function of \(\pi_{t-1}\), and if \(A_t\) and \(c_t\), \(t = 2, \ldots, T\), are deterministic, then \(V_t(\pi_{t-1})\) is only a function of \(\pi_{t-1}\).

**2.2. Two stage.** It is instructive to consider just the two stage case, \(T = 2\). That is consider the problem

\[
(2.9) \quad \min_{x_{1} \geq 0, x_{2} \geq 0} \mathbb{E} \left[ c_1^T x_1 + c_2 (\xi_2)^T x_2 (\xi_2) \right] \\
\text{s.t.} \quad A_1 x_1 = b_1, \quad B_2 (\xi_2) x_1 + A_2 (\xi_2) x_2 (\xi_2) = b_2 (\xi_2).
\]

The Lagrangian of this problem is

\[
L(x_1, x_2, \pi_1, \pi_2) = \mathbb{E} \left[ c_1^T x_1 + c_2 (\xi_2)^T x_2 (\xi_2) \right] + \pi_1^T (b_1 - A_1 x_1) + \pi_2^T (b_2 - B_2 x_1 - A_2 x_2) = \mathbb{E} \left[ (c_1 - A_1^T \pi_1 - B_2^T \pi_2)^T x_1 + (c_2 - A_2^T \pi_2)^T x_2 + b_1^T \pi_1 + b_2^T \pi_2 \right].
\]

The dual of (2.9) is the problem

\[
(2.10) \quad \max_{\pi_1, \pi_2} \min_{x_{1} \geq 0, x_{2} \geq 0} L(x_1, x_2, \pi_1, \pi_2),
\]

that is

\[
\begin{align*}
(2.11) \quad \max_{\pi_1, \pi_2} & \quad b_1^T \pi_1 + \mathbb{E}[b_2 (\xi_2)^T \pi_2 (\xi_2)] \\
\text{s.t.} & \quad c_2 - A_2^T \pi_2 \geq 0, \quad c_1 - A_1^T \pi_1 - \mathbb{E}[B_2^T \pi_2] \geq 0.
\end{align*}
\]

\(^1\)Note that problem (2.5) may have more than one optimal solution. In case of finite number of scenarios the considered linear program always has a solution provided its optimal value is finite.
For finite number $N$ of scenarios the respective primal and dual problems can be written as

$$\begin{align}
\min_{x_1 \geq 0, x_2, \ldots, x_N \geq 0} & \quad c_1^\top x_1 + \sum_{j=1}^{N} p_{2j} c_{2j}^\top x_{2j} \\
\text{s.t.} & \quad A_1 x_1 = b_1, \\
& \quad B_{2j} x_1 + A_{2j} x_{2j} = b_{2j}, \quad j = 1, \ldots, N,
\end{align}$$

(2.12)

and

$$\begin{align}
\max_{\pi_{1}, \pi_{21}, \ldots, \pi_{2N}} & \quad b_1^\top \pi_1 + \sum_{j=1}^{N} p_{2j} b_{2j}^\top \pi_{2j} \\
\text{s.t.} & \quad c_{2j} - A_{2j}^\top \pi_{2j} \geq 0, \quad j = 1, \ldots, N, \\
& \quad c_1 - A_1^\top \pi_1 - \sum_{j=1}^{N} p_{2j} B_{2j}^\top \pi_{2j} \geq 0.
\end{align}$$

(2.13)

Problem (2.13) can also be written as

$$\begin{align}
\max_{\pi_1} & \quad b_1^\top \pi_1 + V_2(\pi_1),
\end{align}$$

(2.14)

where $V_2(\pi_1)$ is the optimal value of the problem

$$\begin{align}
\max_{\pi_{21}, \ldots, \pi_{2N}} & \quad \sum_{j=1}^{N} p_{2j} b_{2j}^\top \pi_{2j} \\
\text{s.t.} & \quad c_{2j} - A_{2j}^\top \pi_{2j} \geq 0, \quad j = 1, \ldots, N, \\
& \quad c_1 - A_1^\top \pi_1 - \sum_{j=1}^{N} p_{2j} B_{2j}^\top \pi_{2j} \geq 0.
\end{align}$$

(2.15)

Remark 2.2. Unlike the primal problem, the dual problem does not decompose for a given $\pi_1$ into separate optimization problems with respect to $\pi_{2j}$. The value function $V_2(\pi_1)$ is concave piecewise linear and could take value $-\infty$ for some $\pi_1$, i.e., it could happen that for some $\pi_1$ the set of $\pi_{2j}$ satisfying the feasibility constraints of the dual problem is empty. That is, it could happen that the dual problem does not have relatively complete recourse, even if the primal problem has relatively complete recourse. This could create a problem for numerical solutions, we will discuss this later.

Remark 2.3. For a given $x_1$ the dual of the primal problem is obtained by dualization of the last constraints in (2.12), that is

$$\begin{align}
\max_{\lambda_1, \ldots, \lambda_N} & \quad \sum_{j=1}^{N} p_{2j} (b_{2j} - B_{2j} x_1)^\top \lambda_j \\
\text{s.t.} & \quad c_{2j} - A_{2j}^\top \lambda_j \geq 0, \quad j = 1, \ldots, N.
\end{align}$$

(2.16)

If the primal problem has relatively complete recourse and finite optimal value, then for $x_1$ satisfying the feasibility constraints of the first stage problem, the second stage problem has finite optimal value. It follows that the optimal value of problem (2.16) is finite and hence that problem has a feasible solution. Comparing problems (2.15) and (2.16), we can conclude that the infeasibility in (2.15) can happen because the last constraint of this problem does not have a solution for some $\pi_1$.

2.3. Relatively complete recourse. As it was pointed in Remark 2.2, already in the two stage case it could happen that the dual problem does not have relatively complete recourse even if the primal problem has it. Recall that we assume that the set of possible realizations (scenarios) of the data process is finite.

Definition 2.1. We say that a sequence $\pi_t$, $t = 1, \ldots, T$, is generated by the forward (dual) process if $\pi_1 \in \mathbb{R}^{n_1}$ and for $\pi_{t-1} = \pi_{t-1}$, $t = 2, \ldots, T$, going forward in time, $\pi_t$ coincides with some $\pi_{tj}$, $j = 1, \ldots, N_t$, where $\pi_{t1}, \ldots, \pi_{tN_t}$ is a feasible solution of the respective dynamic program - program (2.6) for $t = 2, \ldots, T-1$, and program (2.5) for $t = T$. We say that the dual problem (2.3) has relatively complete recourse if at every stage $t = 2, \ldots, T$, for any generated $\pi_{t-1}$ by the forward process, the respective dynamic program has a feasible solution at stage $t$ for every realization of the random data.
Without the relatively complete recourse it could happen that \( V_t(\pi_{t-1}, \xi_{t-1}) = -\infty \) for a generated \( \pi_{t-1} \) and \( \xi_{t-1} = \xi_{t-1,j} \). One way to deal with the problem of absence of relatively complete recourse in numerical procedures is to use feasibility cuts, we will discuss this later. Another way is the following penalty approach which will be used in Section 4. As it was pointed in Remark 2.3, the infeasibility of problem (2.5) can happen because of its last constraint. In order to deal with this, consider the following relaxation of problem (2.5):

\[
\begin{align*}
\max_{\pi_{T,1}, \ldots, \pi_{T,N_T}, \zeta_T \geq 0} & \quad \sum_{j=1}^{N_T} \lambda_T b_{Tj}^T \pi_{Tj} - v_T^T \zeta_T \\
\text{s.t.} & \quad A_T^T \pi_{Tj} \leq c_{Tj}, \quad j = 1, \ldots, N_T, \\
& \quad A_{T-1}^T \pi_{T-1} + \sum_{j=1}^{N_T} p_{Tj} B_{Tj}^T \pi_{Tj} \leq c_{T-1} + \zeta_T,
\end{align*}
\]

(2.17)

where \( v_T \) is a vector with positive components. We have that problem (2.17) is always feasible and hence its optimal value \( \tilde{V}_T(\pi_{T-1}, \xi_{T-1}) > -\infty \). We also have that

\[
\tilde{V}_T(\pi_{T-1}, \xi_{T-1}) \geq V_T(\pi_{T-1}, \xi_{T-1}),
\]

(2.18)

with the equality holding if \( \zeta_T = 0 \) in the optimal solution of (2.17). When \( V_T(\pi_{T-1}, \xi_{T-1}) \) is finite, this equality holds if the components of vector \( v_T \) are large enough.

Similarly, problems (2.6) can be relaxed to

\[
\begin{align*}
\max_{\pi_{t,1}, \ldots, \pi_{t,N_t}, \zeta_t \geq 0} & \quad \sum_{j=1}^{N_t} \lambda_j b_{tj}^T \pi_{tj} + \tilde{V}_{t+1}(\pi_{tj}, \xi_{tj}) - v_t^T \zeta_t \\
\text{s.t.} & \quad A_{t-1}^T \pi_{t-1} + \sum_{j=1}^{N_t} p_{tj} B_{tj}^T \pi_{tj} \leq c_{t-1} + \zeta_t,
\end{align*}
\]

(2.19)

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

2.4. Dynamics of Lagrange multipliers. Consider the two stage problem (2.9). It can be written as

\[
\begin{align*}
\min_{x_1 \geq 0} & \quad c_1^T x_1 + \mathbb{E}[Q(x_1, \xi_2)] \\
\text{s.t.} & \quad A_1 x_1 = b_1,
\end{align*}
\]

(2.20)

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

\[
\begin{align*}
\min_{x_2 \geq 0} & \quad c_2^T x_2 \quad \text{s.t.} \quad B_2(\xi_2)x_1 + A_2(\xi_2)x_2 = b_2(\xi_2).
\end{align*}
\]

(2.21)

The Lagrangian of problem (2.21) is

\[
L(x_1, x_2, \lambda, \xi_2) = c_2(\xi_2)^T x_2 + \lambda^T (b_2(\xi_2) - B_2(\xi_2)x_1 - A_2(\xi_2)x_2).
\]

In the dual form, \( Q(x_1, \xi_{2j}) \) is given by the optimal value of the problem (see (2.16))

\[
\max_{\lambda_j} (b_{2j} - B_{2j} x_1)^T \lambda_j \quad \text{s.t.} \quad c_{2j} - A_{2j}^T \lambda_j \geq 0.
\]

(2.22)

We have that if \( x_1 = \bar{x}_1 \) is an optimal solution of the first stage problem, then optimal Lagrange multipliers \( \pi_{2j} \) are given by the optimal solution of problem (2.22).
This can be extended to the multistage setting of problem (2.1) (recall that the stagewise independence condition is assumed). At the last stage $t = T$, given optimal solution $\bar{x}_{T-1}$, the following problem should be solved

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

For a realization $\xi_T = \xi_{Tj}$, the dual of problem (2.23) is the problem

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

We then have that $\pi_{Tj}$ are given by the optimal solution of problem (2.24).

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

$$
(2.25) \quad \min_{x_{T-1} \geq 0} c_{T-1}(\xi_{T-1})^\top x_{T-1} + Q_T(x_{T-1}) \text{ s.t. } A_{T-1}(\xi_{T-1})x_{T-1} = b_{T-1}(\xi_{T-1}) - B_{T-1}(\xi_{T-1})\bar{x}_{T-2}.
$$

We have that $Q_T(\cdot)$ is a convex piecewise linear function. Therefore for every realization $\xi_{T-1} = \xi_{T-1j}$, it is possible to represent (2.25) as a linear program and hence to write its dual. The optimal Lagrange multipliers of that dual give the corresponding Lagrange multipliers $\pi_{T-1j}$. And so on for other stages going backward in time. That is, we have the following.

**Remark 2.4.** If $(\bar{x}_1, \ldots, \bar{x}_T(\xi_T))$ is an optimal solution of the primal problem, then for $x_{t-1} = \bar{x}_{t-1}$ the Lagrange multiplier $\pi_{tj}$ is given by the respective Lagrange multiplier of problem (2.8).

### 3. Sensitivity analysis

#### 3.1. General case

Suppose now that the data $c_t(\xi_t, \theta), b_t(\xi_t, \theta), B_t(\xi_t, \theta), A_t(\xi_t, \theta)$ of problem (2.1) also depends on parameter vector $\theta \in \mathbb{R}^k$. Denote by $\vartheta(\theta)$ the optimal value of the parameterized problem (2.1) considered as a function of $\theta$, and by $\mathcal{S}(\theta)$ and $\mathcal{D}(\theta)$ the sets of optimal solutions of the respective primal and dual problems. Recall that the sets $\mathcal{S}(\theta)$ and $\mathcal{D}(\theta)$ are nonempty provided the optimal value $\vartheta(\theta)$ is finite. We assume that the data functions are continuously differentiable functions of $\theta$. Suppose further that for a given $\theta = \bar{\theta}$ the optimal value $\vartheta(\bar{\theta})$ is finite and the sets $\mathcal{S}(\bar{\theta})$ and $\mathcal{D}(\bar{\theta})$ of optimal solutions are bounded. Then we have the following formula for the directional derivatives of the optimal value function (e.g., [5, Proposition 4.27])

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

where $L(x, \pi, \theta)$ is the corresponding Lagrangian (see (2.2)) considered as a function of $\theta$. In particular if $\mathcal{S}(\bar{\theta}) = \{\bar{x}\}$ and $\mathcal{D}(\bar{\theta}) = \{\bar{\pi}\}$ are singletons, then $\vartheta(\cdot)$ is differentiable at $\bar{\theta}$ and

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

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

#### 3.2. Application to an inventory model

Consider the inventory model

$$
(3.3) \quad \min \mathbb{E} \left[ \sum_{t=1}^T \alpha_t(y_t - x_{t-1}) + \gamma_t(D_t - y_t)_+ + \eta_t(y_t - D_t)_+ \right] \text{ s.t. } \begin{aligned} x_t &= y_t - D_t, y_t \geq x_{t-1}, t = 1, \ldots, T. \end{aligned}
$$

Here $D_1, \ldots, D_T$ is a (random) demand process, $\alpha_t, \gamma_t, \eta_t$ are the ordering, back-order penalty and holding costs per unit, respectively, $x_t$ is the inventory level and $y_t - x_{t-1}$ is the order quantity at time $t$, the initial
inventory level \( x_0 \) is given. We refer to [28] for a thorough discussion of that model. Note that \( D_t \) is a random variable whereas \( d_t \) stands for a particular realization. We assume that \( g_t > a_t \geq 0, h_t > 0, t = 1, ..., T \).

In the classical setting the demand process is assumed to be stagewise independent, i.e., \( D_{t+1} \) is assumed to be independent of \( D_{[t]} = (D_1, ..., D_t) \) for \( t = 1, ..., T - 1 \). In order to capture the autocorrelation structure of the demand process it is tempting to model it as, say first order, autoregressive process \( D_t = \mu + \phi D_{t-1} + \epsilon_t \), where errors \( \epsilon_t \) are assumed to be a sequence i.i.d (independent identically distributed) random variables. However this approach may result in some of the realizations of the demand process to be negative, which of course does not make sense. One way to deal with this is to make the transformation \( Y_t := \log D_t \) and to model \( Y_t \) as an autoregressive process. A problem with this approach is that it leads to nonlinear equations for the original process \( D_t \), which makes it difficult to use in the numerical algorithms discussed below.

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

\[
D_t = \epsilon_t (\phi D_{t-1} + \mu), \quad t = 1, ..., T,
\]

where \( \phi \in (0, 1), \mu \geq 0 \) are parameters and \( D_0 \geq 0 \) is given. The errors \( \epsilon_t \) are i.i.d with log-normal distributions having means and standard deviations given by \( \mathbb{E}[\epsilon_t] = 1 \) and \( \text{Var}(\epsilon_t) = \sigma^2 > 1 \), respectively. This guarantees that all realizations of the demand process are positive. It is possible to view (3.4) as a linearization of the log-transformed process \( \log D_t \) (cf., [26]). See Section 3.2.1 for a discussion of statistical properties of the process (3.4).

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

### 3.2.1. Statistical properties of the multiplicative autoregressive process.

Consider the multiplicative autoregressive process (3.4). Note that under the specified conditions the demand process is not stationary. Indeed, since the errors \( \epsilon_t \) are i.i.d and \( \mathbb{E}[\epsilon_t] = 1 \) we have that \( \mathbb{E}[D_t] = \phi \mathbb{E}[D_{t-1}] + \mu \) and

\[
\text{Var}(D_t) = \mathbb{E}\left[\text{Var}(\epsilon_t (\phi D_{t-1} + \mu) | D_{t-1})\right] + \text{Var}(\mathbb{E}(\epsilon_t (\phi D_{t-1} + \mu) | D_{t-1})) = \sigma^2 \mathbb{E}\left[(\phi D_{t-1} + \mu)^2\right] + \phi^2 \text{Var}(D_{t-1}).
\]

It follows that \( \mathbb{E}[D_t] \) converges to \( \mu/(1 - \phi) \) as \( t \to \infty \). Suppose, for example, that \( \mu = 0 \). Then \( D_t = \epsilon_t \phi D_{t-1} = D_0 \phi^t \prod_{\tau=1}^t \epsilon_\tau, \ t = 1, ..., T, \mathbb{E}[D_t] = D_0 \phi^t \to 0, \) and \( \text{Var}(D_t) = D_0^2 \phi^{2t}|(1 + \sigma^2) - 1| \). Therefore if \( \phi^2(1 + \sigma^2) < 1 \), then \( \text{Var}(D_t) \to 0 \); and if \( \phi^2(1 + \sigma^2) > 1 \), then \( \text{Var}(D_t) \to \infty \) provided \( D_0 > 0 \).

### 3.2.2. Basic derivations.

Consider the inventory model (3.3) with the demand modeled as the multiplicative autoregressive process (3.4). We view \( D_t \) as additional state variables, governed by equations (3.4). That is write problem (3.3) in the form

\[
\begin{align*}
\min \quad & \mathbb{E}\left[\sum_{t=1}^T a_t(y_t - x_{t-1}) + \Psi_t(y_t, D_t)\right] \\
\text{s.t.} \quad & x_t = y_t - D_t, y_t \geq x_{t-1}, \ t = 1, ..., T, \\
& D_t = \epsilon_t (\phi D_{t-1} + \mu), \ t = 1, ..., T,
\end{align*}
\]

where \( \Psi_t(y_t, D_t) := g_t(d_t - y_t)_+ + h_t(y_t - d_t)_+ \). Here \((x_t, D_t)\) is viewed as state variables and \( \epsilon_t \) as the underlying random process. As before we denote by \( \overline{\theta}(\phi, \mu) \) the optimal value of problem (3.6) viewed as a function of \( \phi \) and \( \mu \). In order to compute the derivatives of \( \overline{\theta}(\phi, \mu) \) we proceed as follows.

Let us write the dynamic programming equations for problem (3.6) (as before we use notation \( \epsilon_{t[0]} := (\epsilon_1, ..., \epsilon_t) \)). At the last stage \( t = T \) we need to solve the problem (conditional on \( \epsilon_{[T-1]} \))

\[
\begin{align*}
\min_{y_T \geq x_{T-1}} \quad & a_T(y_T - x_{T-1}) + \mathbb{E}\left[\Psi_T(y_T, \epsilon_T (\phi D_{T-1} + \mu))\right], \\
\text{s.t.} \quad & x_T = y_T - D_T, y_T \geq x_{T-1}, \\
& D_T = \epsilon_T (\phi D_{T-1} + \mu).
\end{align*}
\]
where the expectation is taken with respect to random variable $\epsilon_T$. We use here the property that $y_T$ and $x_{T-1}$ are functions of $\epsilon_{[T-1]}$ and that $\epsilon_T$ is independent of $\epsilon_{[T-1]}$. The optimal value of problem (3.7) is a function of $(x_{T-1}, D_{T-1})$ and is denoted $Q_T(x_{T-1}, D_{T-1})$. At stages $t = T - 1, \ldots, 1$, the value (cost-to-go) function $Q_t(x_{t-1}, D_{t-1})$ is given by the optimal value of the problem

$$(3.8) \quad \min_{y_t \geq x_{t-1}} a_t(y_t - x_{t-1}) + \mathbb{E} \left[ \Psi_t(y_t, \epsilon_t(\phi D_{t-1} + \mu)) + Q_{t+1}(y_t - \epsilon_t(\phi D_{t-1} + \mu), \epsilon_t(\phi D_{t-1} + \mu)) \right].$$

Note that constraints (3.4) of problem (3.6) are linear in $D_t$, and hence the value functions $Q_t(\cdot, \cdot)$ are convex.

We now proceed to the dualization of constraints (3.4). The Lagrangian of problem (3.6) is

$$(3.9) \quad L(x, y, \pi) = \mathbb{E} \left[ \sum_{t=1}^T a_t(y_t - x_{t-1}) + \Psi_t(y_t, D_t) + \pi_t(-D_t + \epsilon_t(\phi D_{t-1} + \mu)) \right],$$

and the Lagrangian dual of problem (3.6) is

$$(3.10) \quad \max_{\pi} \min_{y_t \geq x_{t-1}} L(x, y, \pi) \text{ s.t. } x_t = y_t - D_t, \ t = 1, \ldots, T.$$ We assume that the number of scenarios is finite. In that case the primal problem (3.6) can be written as a linear program and (3.10) as its dual, and hence the optimal values of problems (3.6) and (3.10) are equal to each other and both problems have optimal solutions.

We can now write the following formulas for the derivatives of the optimal value function (see (3.1) and (3.2))

$$(3.11) \quad \partial \varphi(\phi, \mu)/\partial \phi = \partial L(\bar{x}, \bar{y}, \bar{\pi})/\partial \phi = \mathbb{E} \left[ \sum_{t=1}^T \bar{\pi}_t \epsilon_t D_{t-1} \right],$$

$$(3.12) \quad \partial \varphi(\phi, \mu)/\partial \mu = \partial L(\bar{x}, \bar{y}, \bar{\pi})/\partial \mu = \mathbb{E} \left[ \sum_{t=1}^T \bar{\pi}_t \epsilon_t \right],$$

where $(\bar{x}, \bar{y})$ is an optimal solution of the primal problem and $\bar{\pi}$ are the corresponding Lagrange multipliers. Note that the right hand sides of (3.11) and (3.12) do not depend on $(\bar{x}, \bar{y})$, and that $\bar{\pi}$ is an optimal solution of the dual problem and is the same for any optimal solution $(\bar{x}, \bar{y})$ of the primal problem. Note also that the demand process $D_t$ depends only on the initial value $D_0$ and realizations of the errors $\epsilon_t$. Therefore in order to compute the derivatives of $\partial(\phi, \mu)$ we only need to have a way to compute the Lagrange multipliers $\bar{\pi}$. The required Lagrange multipliers can be computed either by solving the respective dual dynamic programming equations or by solving the primal problem and use Lagrange multipliers of the dynamic equations of the primal problem (see Section 2.4). We also verified numerically that both ways produce the same Lagrange multipliers up to small numerical errors. Then the expectations in (3.11) and (3.12) can be estimated by generating realizations of the error process ($\epsilon_t$) and the corresponding averaging.

4. Dual SDDP. In this section, using the results of Section 2, we propose a solution method for dual problem (2.3) which converges to the optimal value of this problem and therefore to the optimal value of primal problem (2.1). For convenience of numerical calculations we formulate the problem in the following way. The corresponding Dual SDDP method is described for problems of form (2.1) splitting decision $x_t$ for stage $t$ in state $x_t$ and control $u_t$. More precisely, control variables $u_t$ satisfy the dynamics $A_t x_t + C_t u_t = b_t$, $D_t x_t + F_t u_t \leq f_t$, and for $t = 2, \ldots, T$,

$$(4.13) \quad A_t x_t + B_t x_{t-1} + C_t u_t = b_t,$$

$$(4.14) \quad D_t x_t + E_t x_{t-1} + F_t u_t \leq f_t,$$

where $A_t, C_t, b_t, D_t, F_t, f_t$ are deterministic matrices and vectors of appropriate dimensions and $A_t, B_t, C_t, D_t, E_t, F_t, b_t, f_t$, $t = 2, \ldots, T$, are random matrices and vectors (functions of $\xi_t$) of appropriate dimensions. That is, compared with (2.1), now the feasibility constraints are considered in the form (4.13) - (4.14).
We assume that decision \( u_t \) at stage \( t \) is taken given \( \xi_\ell := (\xi_1, \xi_2, \ldots, \xi_\ell) \) generated by the data process. The cost function for stage \( t \) is \( c^T_t x_t + d^T_t u_t \). Therefore, we have the following optimization problem

\[
\min_{x_1, u_1, \ldots, x_T, u_T} \mathbb{E} \left[ \sum_{t=1}^{T} c^T_t x_t + d^T_t u_t \right]
\]  
(4.15)

\[
\begin{align*}
\text{s.t.} & \quad A_1 x_1 + C_1 u_1 = b_1, \\
& \quad D_1 x_1 + F_1 u_1 \leq f_1 \\
& \quad A_t x_t + B_t x_{t-1} + C_t u_t = b_t, \ a.s., t = 2, \ldots, T, \\
& \quad D_t x_t + E_t x_{t-1} + F_t u_t \leq f_t, \ a.s., t = 2, \ldots, T,
\end{align*}
\]

where the optimization is performed over policies \( \pi = ((x_1, u_1), (x_2(\xi_2), u_2(\xi_2)), \ldots, (x_T(\xi_T), u_T(\xi_T))) \) satisfying the feasibility constraints. The dual of (4.15) is given by

\[
\max_{\pi, \mu} \mathbb{E} \left[ \sum_{t=1}^{T} \pi^T_t b_t + \mu^T_t f_t \right]
\]  
(4.16)

\[
\begin{align*}
\text{s.t.} & \quad A^\pi_T \pi_T + D^\pi_T \mu_T = c_T, \ a.s., \\
& \quad C^\pi_T \pi_T + F^\pi_T \mu_T = d_T, \ a.s., \\
& \quad A^\pi_{t-1} \pi_{t-1} + D^\pi_{t-1} \mu_{t-1} + \mathbb{E}[E_{t-1}^\pi + E_{t+1}^\pi | \xi_\ell] = c_t, \ a.s., t = 1, \ldots, T - 1, \\
& \quad C^\pi_{t-1} \pi_{t-1} + F^\pi_{t-1} \mu_{t-1} = d_t, \ a.s., t = 1, \ldots, T - 1, \\
& \quad \mu_t \leq 0, \ a.s., t = 1, \ldots, T.
\end{align*}
\]

For such generic formulation, more convenient for applications, the interested reader can find the implementation of Primal SDDP and all variants of Dual SDDP described in this section at https://github.com/vguigues/Dual_SDDP_Library_Matlab and https://github.com/vguigues/Primal_SDDP_Library_Matlab.

4.1. Dual SDDP for problems with uncertainty in the right-hand side only. We first consider the case where only \( b_t, f_t \) is random in \( \xi_t \). In this case, dynamic programming equations (2.5), (2.6), (2.7) (which are DP equations for dual problem (4.16)) can be written:

\[
V_T(\pi_{T-1}, \mu_{T-1}) = \max_{\pi_T, \mu_T} \sum_{j=1}^{N_T} p_{Tj} \left( \pi^T_{Tj} b_{Tj} + \mu^T_{Tj} f_{Tj} \right)
\]  
(4.17)

\[
\begin{align*}
\text{s.t.} & \quad A^\pi_T \pi_T + D^\pi_T \mu_T = c_T, \ j = 1, \ldots, N_T, \\
& \quad C^\pi_T \pi_T + F^\pi_T \mu_T = d_T, \ j = 1, \ldots, N_T, \\
& \quad \mu_{Tj} \leq 0, \ j = 1, \ldots, N_T, \\
& \quad A^\pi_{T-1} \pi_{T-1} + D^\pi_{T-1} \mu_{T-1} + \sum_{j=1}^{N_T} p_{Tj} \left( B^\pi_{Tj} \pi_{Tj} + E^\pi_{Tj} \mu_{Tj} \right) = c_{T-1},
\end{align*}
\]

for \( t = 2, \ldots, T - 1: \)

\[
V_t(\pi_{t-1}, \mu_{t-1}) = \max_{\pi_t, \mu_t} \sum_{j=1}^{N_t} p_{ij} \left( \pi^T_{tj} b_{tj} + \mu^T_{tj} f_{tj} + V_{t+1}(\pi_{tj}, \mu_{tj}) \right)
\]  
(4.18)

\[
\begin{align*}
\text{s.t.} & \quad C^\pi_t \pi_{tj} + F^\pi_t \mu_{tj} = d_t, \ j = 1, \ldots, N_t, \\
& \quad \mu_{tj} \leq 0, \ j = 1, \ldots, N_t, \\
& \quad A^\pi_{t-1} \pi_{t-1} + D^\pi_{t-1} \mu_{t-1} + \sum_{j=1}^{N_t} p_{tj} \left( B^\pi_{tj} \pi_{tj} + E^\pi_{tj} \mu_{tj} \right) = c_{t-1},
\end{align*}
\]

and the first stage problem is

\[
\max_{(\pi_1, \mu_1) \in D_1} \pi^T_1 b_1 + \mu^T_1 f_1 + V_2(\pi_1, \mu_1),
\]  
(4.19)

where, here and in what follows, \( D_1 \) denotes the set \( D_1 := \{ (\pi_1, \mu_1) : C^\pi_1 \pi_1 + F^\pi_1 \mu_1 = d_1, \mu_1 \leq 0 \} \). We will make the following assumption.

(A2) For every stage \( t \), any realization of \( \xi_t^{t-1} = (\xi_1, \ldots, \xi_t) \), and any state \( x_{t-1} \), the feasible set for stage \( t \) is nonempty and bounded for all possible values of \( \xi_t \).
If the dual problem (4.16) satisfies the RCR assumption, we can apply an SDDP type method to compute upper approximations of functions $V_t$ in (4.17), (4.18), (4.19). More precisely, concave value functions $V_t$, $t = 2, \ldots, T$, are approximated at the end of iteration $k$ by polyhedral upper bounding functions $V_t^k$ given by:

$$
V_t^k(\pi_{t-1}, \mu_{t-1}) = \min_{0 \leq i \leq k} \bar{\theta}_i + \langle \bar{\pi}_i, \pi_{t-1} \rangle + \langle \bar{\gamma}_i, \mu_{t-1} \rangle.
$$

The algorithm also uses valid upper bounds on the norm of dual optimal solutions:

**Lemma 4.1.** Let Assumptions (A1) and (A2) hold. Let $((\pi_1, \mu_1^0), \ldots, (\pi_T, \mu_T^0))$ be an optimal solution of dual problem (4.16). Then for every $t = 1, \ldots, T$, there is some compact set $B_t$ such that almost surely $(\pi_t^*, \mu_t^*) \in B_t$.

A proof of Lemma 4.1 and a way to obtain the corresponding sets $B_t$ can be found in [10]. The corresponding Dual SDDP algorithm is given in the Appendix and the convergence of the method is given in Theorem 4.2 below.

**Theorem 4.2.** Consider optimization problem (4.15) and let Assumptions (A1) and (A2) hold. Assume that RCR holds for dual problem (4.16) and that the samples in the forward passes are independent. Then the sequence $V^k$ generated by Dual SDDP is a deterministic sequence of upper bounds on the optimal value of (4.15) which converges to the optimal value of this problem.

As mentioned earlier, the RCR on the dual may not be satisfied, even if RCR holds for the primal. In the rest of this section, we discuss two variants of Dual SDDP adapted to this situation.

**Dual SDDP with penalizations.** Dual SDDP with penalizations is based on the developments of Section 2.3. It introduces slack variables in the constraints which may become infeasible for some past decisions in the subproblems solved in the forward passes of Dual SDDP. Slack variables are penalized in the objective function with sequences $(q_{tk})_k$ and $(r_{tk})_k$ of penalizing coefficients. Therefore, all subproblems solved in forward and backward passes of this variant of Dual SDDP, called Dual SDDP with penalizations, are always feasible and at iteration $k$, the method still builds polyhedral upper bounding function $V_t^k$ for $V_t$ of form (4.20) (see Proposition 4.3). Dual SDDP we have just presented can be seen as a limiting case, obtained taking null sequences for penalizing coefficients $(q_{tk})_k$ and $(r_{tk})_k$. Similarly to SDDP, trial points are generated in a forward pass and cuts for $V_t$ are computed in a backward pass. The detailed Dual SDDP method with penalizations is given below.

**Dual SDDP with penalizations for DP equations (4.17), (4.18), (4.19).**

**Initialization.** For $t = 2, \ldots, T$, take $V_0^0 = +\infty$, i.e., take $\bar{\theta}_0 = +\infty$, $\bar{\beta}_1 = 0$, and $\pi_0^0 = 0$, for $t = 2, \ldots, T$. Set iteration counter $k$ to 1.

**Step 1: forward pass (computation of dual trial points).** Compute an optimal solution $(\pi_1^k, \mu_1^k)$ of

$$
V^{k-1} = \max_{(\pi_1, \mu_1) \in B_1 \cap D_1} b_1^\top \pi_1 + f_1^\top \mu_1 + V_2^{k-1}(\pi_1, \mu_1).
$$

For $t = 2, \ldots, T - 1$, given $(\pi_{t-1}^k, \mu_{t-1}^k)$, compute an optimal solution of

$$
\begin{align*}
\max_{\pi_t, \mu_t} & \sum_{j=1}^{N_t} p_{tj} \left( b_{tj}^\top \pi_j + \mu_j^\top f_{tj} + V_{t+1}^{k-1}(\pi_j, \mu_j) \right) - q_{tk} \nu_t^+ - r_{tk} \nu_t^-
\end{align*}
$$

$$
C_t^\top \pi_j + F_t^\top \mu_t = d_t, \quad j = 1, \ldots, N_t,
$$

$$
A_{t-1}^\top \pi_{t-1}^k + D_{t-1}^\top \mu_{t-1}^k + \sum_{j=1}^{N_t} p_{tj} \left( B_t^\top \pi_j + E_t^\top \mu_j \right) + \nu_t^+ - \nu_t^- = c_{t-1},
$$

$$
\mu_{tj} \leq 0, \quad (\pi_{tj}, \mu_{tj}) \in B_t, \quad j = 1, \ldots, N_t.
$$
An optimal solution of the problem above has \( N_t \) components \((\pi_{t1}, \mu_{t1}), (\pi_{t2}, \mu_{t2}), \ldots, (\pi_{tN_t}, \mu_{tN_t})\). We take \((\pi^*_t, \mu^*_t)\) to be one of these components knowing that component \(i\) is sampled with probability \( p_{ti} \).

**Step 2: backward pass (computation of new cuts).** Let \((\alpha, \lambda, \delta)\) be an optimal solution of

\[
(4.23) \quad \min_{\alpha, \lambda, \delta} \delta^\top (c_{T-1} - A^\top_{T-1} \pi^k_{T-1} - D^\top_{T-1} \mu^k_{T-1}) + \sum_{j=1}^{N_T} (c^\top_j \alpha_j + d^\top_j \lambda_j + \Psi^\top_{j1} \pi_T + \Psi^\top_{j2} \pi_T + \Psi^\top_{j3} \pi_T + \Psi^\top_{j4} \mu_T) + A \alpha_j + p_r \beta_T + C_T \lambda_j + \Psi_{j1} \lambda_j + \Psi_{j2} \lambda_j + \Psi_{j3} \lambda_j + \Psi_{j4}, j = 1, \ldots, N_T,
\]

and the cut coefficients \( \beta_T = -A_{T-1} \delta, \gamma_T = -D_{T-1} \delta \).

For \( t = T - 1, \ldots, 2 \), compute an optimal solution \((\nu, \lambda, \delta)\) of

\[
(4.24) \quad \min_{\nu, \lambda, \delta} \delta^\top c_{t-1} + \sum_{j=1}^{N_t} \left( c^\top_j \alpha_j + d^\top_j \lambda_j + \Psi^\top_{j1} \pi_t + \Psi^\top_{j2} \pi_t + \Psi^\top_{j3} \pi_t + \Psi^\top_{j4} \mu_t \right) + \sum_{i=0}^{k} \sum_{j=1}^{N_t} \nu_i(j)
\]

\[
C_T \lambda_j + p_t b_t \delta - \sum_{i=0}^{k} \nu_i(j) \sum_{i=0}^{k} \nu_i(j) \sum_{i=0}^{k} \nu_i(j) \Psi^\top_{j1} + \Psi^\top_{j3} + \Psi^\top_{j4} \leq p_t f_{ij}, j = 1, \ldots, N_t,
\]

\[
\sum_{i=0}^{k} \nu_i(j) \leq p_t, j = 1, \ldots, N_t,
\]

\[
\Psi_{j1}, \Psi_{j3} \geq 0, \Psi_{j2}, \Psi_{j4} \leq 0, j = 1, \ldots, N_t,
\]

\[
\nu_0, \ldots, \nu_k \geq 0, -q_{tk} \leq \delta \leq r_{tk},
\]

and the cut coefficients

\[
\beta^k_i = \delta^\top c_{t-1} + \sum_{j=1}^{N_t} \left( c^\top_j \lambda_j + \Psi^\top_{j1} \pi_t + \Psi^\top_{j2} \pi_t + \Psi^\top_{j3} \pi_t + \Psi^\top_{j4} \mu_t \right) + \sum_{i=0}^{k} \sum_{j=1}^{N_t} \nu_i(j), \beta^k_i = -A_{t-1} \delta, \gamma^k_i = -D_{t-1} \delta.
\]

**Step 3:** Do \( k \leftarrow k + 1 \) and go to Step 1.

The validity of the cuts computed in the backward pass of Dual SDDP with penalizations is shown in Proposition 4.3.

**Proposition 4.3.** Consider Dual SDDP algorithm with penalizations. Let Assumptions (A1) and (A2) hold for \((4.15)\) and assume the samples in the forward passes are independent. Then for every \( t = 2, \ldots, T \), the sequence \( V^k_t \) is a nonincreasing sequence of upper bounding functions for \( V_t \), i.e., for every \( k \geq 1 \) and \((\pi_{t-1}, \mu_{t-1})\) we have \( V_t(\pi_{t-1}, \mu_{t-1}) \leq V^k_t(\pi_{t-1}, \mu_{t-1}) \) and therefore \( V^k \) (recall that \( V^{k-1} \) is the optimal value of \((4.21)\)) is a nonincreasing deterministic sequence of upper bounds on the optimal value of \((4.15)\).

To understand the effect of the sequence of penalizing parameters \((q_{tk})\) and \((r_{tk})\) on Dual SDDP with

\[\text{(We suppressed the dependence of the optimal solution on } T \text{ and } k \text{ to alleviate notation.)}\]
penalizations, we define the following Dynamic Programming equations:

\[(4.15)\]

\[
V_T^\gamma(\pi_{T-1}, \mu_{T-1}) = \begin{cases} 
\max_{\pi_T, \mu_T} & \sum_{j=1}^{N_T} p_{Tj} (\pi_T^{\top} b_{Tj} + \mu_T^{\top} f_{Tj}) - \gamma \mathbf{e}^{\top} (\nu_T^+ + \nu_T^-) \\
\text{s.t.} & A_T^{\top} \pi_{Tj} + D_T^{\top} \mu_{Tj} = c_T, \ j = 1, \ldots, N_T, \\
& C_T^{\top} \pi_{Tj} + F_T^{\top} \mu_{Tj} = d_T, \ j = 1, \ldots, N_T, \\
& \mu_{Tj} \leq 0, \ j = 1, \ldots, N_T, \\
& A_T^{\top} \pi_{T-1} + D_{T-1}^{\top} \mu_{T-1} + \sum_{j=1}^{N_T} p_{Tj} (B_T^{\top} \pi_{Tj} + E_T^{\top} \mu_{Tj}) + \nu_T^+ - \nu_T^- = c_{T-1}, \\
& \nu_T^+, \nu_T^- \geq 0, 
\end{cases}
\]

for \( t = 2, \ldots, T - 1 \):

\[(4.26)\] \(V_T^\gamma(\pi_{t-1}, \mu_{t-1}) = \begin{cases} 
\max_{\pi_t, \mu_t, \nu_t} & \sum_{j=1}^{N_t} p_{tj} (\pi_t^{\top} b_{tj} + \mu_t^{\top} f_{tj} + V_{t+1}^\gamma(\pi_{tj}, \mu_{tj})) - \gamma \mathbf{e}^{\top} (\nu_t^+ + \nu_t^-) \\
\text{s.t.} & C_t^{\top} \pi_{tj} + F_t^{\top} \mu_{tj} = d_t, \ j = 1, \ldots, N_t, \\
& \mu_{tj} \leq 0, \ j = 1, \ldots, N_t, \\
& A_{t-1}^{\top} \pi_{t-1} + D_{t-1}^{\top} \mu_{t-1} + \sum_{j=1}^{N_t} p_{tj} (B_t^{\top} \pi_{tj} + E_t^{\top} \mu_{tj}) + \nu_t^+ - \nu_t^- = c_{t-1}, \\
& \nu_t^+, \nu_t^- \geq 0, 
\end{cases}
\]

and we define the first stage problem

\[(4.27)\] \[
\max_{(\pi_1, \mu_1) \in \mathcal{D}_1} \pi_1^{\top} b_1 + \mu_1^{\top} f_1 + V_2^\gamma(\pi_1, \mu_1),
\]

where \( \mathbf{e} \) is a vector of ones and \( \gamma \) is a positive real number. As we will see below, \( V_t^\gamma \) can be seen as an upper bounding concave approximation of \( V_t \) which gets “closer” to \( V_t \) when \( \gamma \) increases. For inventory problem (3.3), it is easy to see that functions \( V_t \) in DP equations (4.17), (4.18), (4.19) and functions \( V_t^\gamma \) in DP equations (4.25), (4.26), (4.27) (obtained using in these equations data \( c_t, d_t, A_t, B_t, C_t, D_t, E_t, F_t, b_t, f_t \), corresponding to the inventory problem) are only functions of one-dimensional state variable \( \pi_{t-1} \). Therefore, Dynamic Programming can be used to solve these Dynamic Programming equations and obtain good approximations of functions \( V_t \) and \( V_t^\gamma \). To obtain these approximations, we need to obtain approximations of the domains of functions \( V_t \) and compute approximations of these functions on a set of points in that domain. To observe the impact of penalizing term \( \gamma \) on \( V_t^\gamma \), we run Dynamic Programming both on DP equations (4.17), (4.18), (4.19) and on DP equations (4.25), (4.26), (4.27) for \( \gamma = 1, 100 \), and 1000, on an instance of the inventory problem with \( T = N_t = 20 \). The corresponding graphs of \( V_2 \) (bold dark solid line) and of \( V_2^\gamma \) for \( \gamma = 1, 10, 1000 \), are represented in Figure 1. We observe that all functions \( V_2^\gamma \) are, as expected, concave upper bounding functions for \( V_2 \) finite everywhere. We also see that on the domain of \( V_2 \), \( V_2^\gamma \) gets closer to \( V_2 \) when \( \gamma \) increases and eventually coincides with \( V_2 \) on this domain when \( \gamma \) is sufficiently large. Similar graphs were observed for remaining functions \( V_t, V_t^\gamma, t = 3, \ldots, T \). Therefore, convergence of Dual SDDP with penalizations requires the coefficients \( q_{tk} \) and \( r_{tk} \) to become arbitrarily large:

**Theorem 4.4.** Consider optimization problem (4.15) and Dual SDDP with penalizations applied to this problem. Let Assumptions (A1) and (A2) hold. Assume that the samples in the forward passes are independent and that \( \lim_{k \to +\infty} q_{tk} = \lim_{k \to +\infty} r_{tk} = +\infty \) for all stage \( t \). Then the sequence \( V^k \) is a deterministic sequence of upper bounds on the optimal value of (4.15) which converges to the optimal value of this problem.

**Dual SDDP with feasibility cuts.** For dual problems not satisfying the RCR assumption, a subproblem for a given stage \( t \) in the forward pass can be infeasible. In this situation, as was done in Section 5 of [8] for SDDP, we can build a feasibility cut for stage \( t - 1 \) and go back to the previous stage \( t - 1 \) to resolve the problem with that feasibility cut added, and so on until a sequence of feasible states is obtained.
for all stages. The backward pass is similar, with the feasibility cuts added. More precisely, we define for stage $t$ the set

$$S_t := \left\{ (\pi_t, \mu_t) : \tilde{C}_t^T \pi_t + \tilde{F}_t^T \mu_t \leq \tilde{d}_t \right\},$$

where matrices and vectors $\tilde{C}_t, \tilde{F}_t, \tilde{d}_t$ are updated along the iterations and such that a feasible $\pi_t, \mu_t$ must belong to $S_t$. The detailed algorithm is given below and the analysis of the algorithm is given in the Appendix.

**Dual SDDP with feasibility cuts for DP equations (4.17), (4.18), (4.19).**

**Initialization.** For $t = 2, \ldots, T$, take $V^0_t \equiv +\infty$, i.e., take $\theta^0_t = +\infty$ and $\beta^0_t = 0, \gamma^0_t = 0$, for $t = 2, \ldots, T$. Set iteration counter $k$ to 1.

**Step 1: forward pass (computation of dual trial points).**

$t = 1$.

**While** $t \leq T$

- **if** $t = 1$ compute an optimal solution $\pi^k_t, \mu^k_t$ of

  $$V^{k-1} = \max_{(\pi_1, \mu_1) \in B_1 \cap D_1 \cap S_1} b_1^T \pi_1 + f_1^T \mu_1 + V^{k-1}_2(\pi_1, \mu_1),$$

  and do $t \leftarrow t + 1$.

- **else**

  - **if** $t = T$

    // Check feasibility of the subproblem for stage $T$ given $(\pi^k_{T-1}, \mu^k_{T-1})$ computing an optimal solution $\delta, \alpha, \lambda$ of

      $$
      \hat{V}_T(\pi^k_{T-1}, \mu^k_{T-1}) := \begin{cases} 
      \max_{\delta, \alpha, \lambda} \sum_{j=1}^{N_T} c_{T,j}^T \alpha_j + d_{T,j}^T \lambda_j + \delta^T (c_{T-1}^T - A_{T-1}^T \pi^k_{T-1} - D_{T-1}^T \mu^k_{T-1}) \\
      p_T B_T \delta + A_T \alpha_j + C_T \lambda_j = 0, \ j = 1, \ldots, N_T, \\
      p_T E_T \delta + D_T \alpha_j + F_T \lambda_j \geq 0, \ j = 1, \ldots, N_T, \\
      -e \leq \delta \leq e,
      \end{cases}
      $$

    where $e$ is a vector of ones.

    - if $\hat{V}_T(\pi^k_{T-1}, \mu^k_{T-1}) > 0$ then add to $\tilde{C}_{T-1}^T$ the row $-(A_{T-1} \delta)^T$, add to $\tilde{F}_{T-1}$ the row $-(D_{T-1} \delta)^T$, add to $\tilde{d}_{T-1}$ the component $-\delta^T c_{T-1} - \sum_{j=1}^{N_T} (c_{T,j}^T \alpha_j + d_{T,j}^T \lambda_j)$, and do $t \leftarrow t - 1$. 

Fig. 1. Graph of $V_2$ and of $V_2^\gamma$ for $\gamma = 1, 100, 1000$. 

else $t \leftarrow t + 1$
end if
else compute an optimal solution $\delta, \alpha, \lambda$ of

$$\hat{V}_i^k(\pi_{i-1}^k, \mu_{i-1}^k) := \begin{cases} \max_{\delta, \alpha, \lambda} & \sum_{j=1}^{N_t} d_{i_1}^T \lambda_j + \tilde{d}_{i_1}^T \tilde{\lambda}_j + \delta^T (c_{t-1} - A_{t-1}^T \pi_{t-1}^k - D_{t-1}^T \mu_{t-1}^k) \\ & p_{ij} B_t \delta + C_t \lambda_j + \tilde{C}_t \tilde{\lambda}_j = 0, j = 1, \ldots, N_t, \\ & p_{ij} E_t \delta + F_t \lambda_j + \tilde{F}_t \lambda_j \geq 0, j = 1, \ldots, N_t, \\ & -e \leq \delta \leq e, \lambda_j \leq 0, j = 1, \ldots, N_t, \end{cases}$$

\hspace{1cm} (4.30)

where $e$ is a vector of ones.

if $\hat{V}_i^k(\pi_{i-1}^k, \mu_{i-1}^k) > 0$ then add to $\tilde{C}_{t-1}^T$ the row $-(A_{t-1} \delta)^T$, add to $\tilde{F}_{t-1}$ the row $-(D_{t-1} \delta)^T$, add to $\tilde{d}_{t-1}$ the component $-\delta^T c_{t-1} - \sum_{j=1}^{N_t} (d_{i}^T \lambda_j + \tilde{d}_{i}^T \tilde{\lambda}_j)$, and do $t \leftarrow t - 1$.
else compute an optimal solution $(\pi_i^k, \mu_i^k)$, of

$$\begin{align*}
\max_{\pi_j, \mu_j} & \sum_{j=1}^{N_t} p_{ij} (b_{i}^T \pi_{ij} + \mu_{i}^T f_t + V_{t+1}^{k-1}(\pi_{ij}, \mu_{ij})) \\
& C_t^T \pi_{ij} + F_t^T \mu_{ij} = \tilde{d}_t, \quad \tilde{C}_t^T \pi_{ij} + \tilde{F}_t^T \mu_{ij} \leq \tilde{d}_t, \\& A_{t-1}^T \pi_{i-1}^k + D_{t-1}^T \mu_{i-1}^k + \sum_{j=1}^{N_t} p_{ij} (B_t^T \pi_{ij} + E_t^T \mu_{ij}) = c_{t-1}, \\& \mu_{ij} \leq 0, \quad (\pi_{ij}, \mu_{ij}) \in B_t, \quad j = 1, \ldots, N_t.
\end{align*}$$

An optimal solution of the problem above has $N_t$ components $(\pi_{i1}, \mu_{i1}), (\pi_{i2}, \mu_{i2}), \ldots, (\pi_{iN_t}, \mu_{iN_t})$. We take $(\pi_i^k, \mu_i^k)$ to be one of these components knowing that component $i$ is sampled with probability $p_{ti}$ and do $t \leftarrow t + 1$.
end if
end if
end while

**Step 2: backward pass.** Compute a new cut for $V_i$ to build $V_i^k$ exactly as in Dual SDDP.

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

The validity of the feasibility and optimality cuts computed by Dual SDDP with feasibility cuts is given in Proposition 4.5 below.

**Proposition 4.5.** Consider optimization problem (4.15) and let Assumptions (A1) and (A2) hold. The following holds for Dual SDDP with feasibility cuts:

(i) all subproblems solved in the forward passes are feasible and almost surely after a finite number of iterations no feasibility cuts are added.

(ii) For every $t = 2, \ldots, T$, the sequence $V_i^k$ generated by Dual SDDP with feasibility cuts is a nonincreasing sequence of upper bounding functions for $V_t$ and therefore $(V^k)$ is a nonincreasing deterministic sequence of upper bounds on the optimal value of (4.15).

Theorem 4.6 states the convergence of Dual SDDP with feasibility cuts.

**Theorem 4.6.** Consider Dual SDDP with feasibility cuts applied to optimization problem (4.15) and let Assumptions (A1) and (A2) hold. Assume that the samples in the forward passes are independent. Then the sequence $V^k$ is a deterministic sequence of upper bound on the optimal value of (4.15) which converges to the optimal value of this problem.
4.2. Dual SDDP for problems with uncertainty in all parameters. We have seen in Section 2.1 how to write DP equations on the dual problem of a MSLP when all data \((A_t, B_t, c_t, b_t)\) in \((\xi_t)\) is random. In this situation, cost-to-go functions \(V_t\) are functions \(V_t(\pi_{t-1}, \mu_{t-1}, \xi_{t-1})\) of both past decision \((\pi_{t-1}, \mu_{t-1})\) and past value \(\xi_{t-1}\) of process \((\xi_t)\). Also recall that functions \(V_t(\cdot, \xi_{t-1})\) are concave for all \(\xi_{t-1}\). Therefore, Dual SDDP and Dual SDDP with penalizations from the previous section must be modified as follows. For each stage \(t = 2, \ldots, T\), instead of computing just one approximation of a single function \((V_t(\cdot, \xi_{t-1}))\), we now need to compute approximations of each stage \(t\). Dual SDDP and Dual SDDP with penalizations from the previous section must be modified as follows. For each stage \(t = 2, \ldots, T\), instead of computing just one approximation of a single function \((V_t(\cdot, \xi_{t-1}))\), we now need to compute approximations of each stage \(t\).

\[
\begin{align*}
Q_{t}(x_{t-1}, c_{t-1}, \ldots, c_{t-p}) &= \mathbb{E}_{b_t, \varepsilon_t} \left[ Q_{t+1}(x_{t+1}, c_{t+1}, \ldots, c_{t+1-p}) \right] \\
A_{t}x_t + B_{t}x_{t-1} &= b_t,
\end{align*}
\]

while the first stage problem is

\[
\begin{align*}
\min_{x_t} & \quad c_1^\top x_1 + Q_2(x_1, c_1, \ldots, c_2-p) \\
A_1x_1 &= b_1.
\end{align*}
\]

The standard SDDP does not apply directly to solve Dynamic Programming equations (4.33)-(4.34) because functions \(Q_t\) given by (4.33)-(4.34) are not convex. Nevertheless, we can use the Markov Chain discretization variant of the SDDP to solve Dynamic Programming equations (4.33)-(4.34). On the other
hand, as it was pointed above, it is possible to apply the SDDP method for the dual problem with the added state variables.

We first write Dynamic Programming equations for the dual of (2.1) with (c_t) of form (4.32). For the last stage T, we have to solve the problem:

$$V_T(\pi_{T-1}, c_{T-1}, \ldots, c_{T-p}) = \max_{\pi_T} \left\{ \sum_{j=1}^{N_T} p_{Tj} \pi_T^{\top} B_{Tj} \right\}$$

s.t. $$A_T \pi_T \leq c_T \circ (\mu_T + \sum_{t=1}^{p} \Phi_T c_{T-\ell}), j = 1, \ldots, N_T,$$

$$\sum_{j=1}^{N_T} p_{Tj} B_T \pi_T \leq c_{T-1} - A_{T-1} \pi_{T-1}. $$

Next for stage $t = 2, \ldots, T - 1$, given $V_{t+1}$, we need to solve the problem

$$V_t(\pi_{t-1}, c_{t-1}, \ldots, c_{t-p}) = \max_{\pi_t} \left\{ \sum_{j=1}^{N_t} p_{tj} \pi_t^{\top} b_{tj} + V_{t+1} \left( \pi_t, \varepsilon_t \circ (\mu_t + \sum_{\ell=1}^{p} \Phi_{t\ell} c_{t-\ell}), c_{t-1}, \ldots, c_{t+1-p} \right) \right\}$$

s.t. $$\sum_{j=1}^{N_t} p_{tj} B_t \pi_t \leq c_{t-1} - A_{t-1} \pi_{t-1},$$

while the first stage problem is

$$\max_{\pi_1} \pi_1^{\top} b_1 + V_2(\pi_1, c_1, \ldots, c_{2-p}).$$

Observe that functions $V_t$ in (4.35)-(4.36) are concave. Dual SDDP with penalizations applied to DP equations (4.35)-(4.36) builds polyhedral approximations of these functions of form

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

at iteration $k$. At iteration $k$, a forward pass computes trial points $\pi_{t-1}^k, c_{t-1}^k, \ldots, c_{t-p}^k$ at which cuts are computed for $V_t$ in a backward pass. As before, we denote by $B_t$ a compact set containing the set of optimal dual solutions for stage $t$. The algorithm is given below.

**Step 0. Initialization.** Coefficients $\beta_0^i, \theta_t^i$ are chosen in such a way that $V_1^0$ is an upper bound on $V_t$. Define positive penalizing sequences $(q_{tk})_k$ for $t = 2, \ldots, T$ and set the iteration counter $k$ to one.

**Step 1. Forward pass.** Sample a trajectory $c_1, c_2^k, \ldots, c_T^k$, of process $(c_t)$ for stages $t = 1, \ldots, T$. Find an optimal solution $\pi_t^k$ of

$$\max_{\pi_t \in B_1} \pi_t^{\top} b_1 + V_2^{k-1}(\pi_1, c_1, \ldots, c_{2-p}).$$

For $t = 2, \ldots, T - 1$, compute an optimal solution $(\pi_t, \ldots, \pi_{tN_t}, v_t)$ of

$$\max_{\pi_t, v_t} \left\{ \sum_{j=1}^{N_t} p_{tj} \left( \pi_t^{\top} b_{tj} + V_{t+1} \left( \pi_t, \varepsilon_t \circ (\mu_t + \sum_{\ell=1}^{p} \Phi_{t\ell} c_{t-\ell}), c_{t-1}, \ldots, c_{t+1-p} \right) \right) \right\}$$

s.t. $$-v_t + \sum_{j=1}^{N_t} p_{tj} B_t \pi_t \leq c_{t-1} - A_{t-1} \pi_{t-1}, v_t \geq 0, \pi_t \in B_t, j = 1, \ldots, N_t.$$ 

Take $\pi_t^k = \pi_{tj(t,k)}$ where index $j(t,k)$ is such that $\varepsilon_t^k = \varepsilon_{tj(t,k)}$.

**Step 2. Backward pass.** Compute an optimal solution $(\delta, \lambda_1, \ldots, \lambda_{N_T})$ of

$$\min_{\delta, \lambda} \left( c_{T-1}^k - A_{T-1}^k \pi_{T-1}^k \right) + \sum_{j=1}^{N_T} \lambda_j^T \left( \varepsilon_{Tj} \circ (\mu_T + \sum_{\ell=1}^{p} \Phi_{T\ell} c_{T-\ell}) \right)$$

s.t. $$A_T \lambda_j + p_{Tj} B_T \delta = p_{Tj} b_{Tj}, \lambda_j \geq 0, j = 1, \ldots, N_T,$$

$$0 \leq \delta \leq q_{tk}. $$

We suppress the dependence of the solution with respect to $T, k$ to alleviate notation.
Compute

\[
\theta^k_T = \sum_{j=1}^{N_T} \lambda_j^T (\varepsilon_{Tj} \circ \mu_T), \quad \beta^k_{T0} = -A_{T-1} \delta, \\
\beta^k_{T1} = \delta + \sum_{j=1}^{N_T} \Phi^T_{T1}(\lambda_j \circ \varepsilon_{Tj}), \quad \beta^k_{T\ell} = \sum_{j=1}^{N_T} \Phi^T_{T\ell}(\lambda_j \circ \varepsilon_{Tj}), \quad 2 \leq \ell \leq p.
\]

For \( t = T-1, \ldots, 2 \), compute an optimal solution \((\pi_t, \ldots, \pi_{tN_t}, \Psi_1, \ldots, \Psi_{N_t}, v_t)^t\) of (4.42)

\[
\begin{aligned}
\min_{\delta, \nu} & \sum_{i=0}^{k} \sum_{j=1}^{N_T} \nu_{ij} \left( \theta^i_{t+1} + \langle \beta^i_{t+11}, \varepsilon_{tj} \circ (\mu_t + \sum_{\ell=1}^{p} \Phi_t(c_{t-\ell})) \rangle + \sum_{\ell=2}^{p} \langle \beta^i_{t+1\ell}, c_{t+1-\ell} \rangle \right) + \delta^T (c^k_{t-1} - A^T_{t-1} \pi^k_{t-1}) \\
p_{ij} &= \sum_{i=0}^{k} \nu_{ij}, \quad j = 1, \ldots, N_t, \\
0 &\leq \delta \leq q_k, \nu \geq 0, \\
p_{ij}b_{ij} &= p_{ij}B_i \delta - \sum_{i=0}^{k} \nu_{ij} \beta^i_{t+10}, \quad j = 1, \ldots, N_t.
\end{aligned}
\]

Compute

\[
\begin{aligned}
\theta^k_t &= \sum_{i=0}^{k} \sum_{j=1}^{N_T} \nu_{ij} (\theta^i_{t+1} + \langle \beta^i_{t+11}, \varepsilon_{tj} \circ \mu_t \rangle), \quad \beta^k_{t0} = -A_{t-1} \delta, \\
\beta^k_{t1} &= \delta + \sum_{i=1}^{k} \sum_{j=1}^{N_t} \nu_{ij} \left( \beta^i_{t+12} + \Phi^T_{t1}(\beta^i_{t+11} \circ \varepsilon_{tj}) \right), \\
\beta^k_{t\ell} &= \sum_{i=1}^{k} \sum_{j=1}^{N_t} \nu_{ij} \left( \Phi^T_{t\ell}(\beta^i_{t+11} \circ \varepsilon_{tj}) + \beta^i_{t+1\ell+1} \right), \quad 2 \leq \ell \leq p-1, \\
\beta^k_{t\ell p} &= \sum_{i=1}^{k} \sum_{j=1}^{N_t} \nu_{ij} \Phi^T_{t\ell p}(\beta^i_{t+11} \circ \varepsilon_{tj}).
\end{aligned}
\]

**Step 4.** Do \( k \leftarrow k + 1 \) and go to Step 1.

It is easy to check that the cut

\[
\theta^k_t + \langle \beta^k_{t0}, \pi_{t-1} \rangle + \sum_{j=1}^{p} \langle \beta^k_{tj}, c_{t-j} \rangle
\]

computed at iteration \( k \) given by (4.43) is an upper bounding function for \( V_t \). Moreover, under Assumption (A2) and if samples in the forward passes are independent, the sequence of optimal values of the approximate first stage problems converges almost surely to the optimal value of the problem.

**5. Numerical experiments.** In this section, we report numerical results obtained applying Primal SDDP and variants of Dual SDDP to the inventory problem and to a Brazilian interconnected power system problem. All methods were implemented in Matlab and run on an Intel Core i7, 1.8GHz, processor with 12.0 Go of RAM. Optimization problems were solved using Mosek [1].

**5.1. Dual SDDP for the inventory problem.** We consider the inventory problem (3.3) which is of form (4.15). The following parameters are chosen: \( a_t = 1.5 + \cos(\frac{\pi t}{T}) \), \( p_t = \frac{1}{N} \) where \( N \) is the number of realizations for each stage, \( \xi_{tj} = (5 + 0.5t)(1.5 + 0.1z_{tj}) \) where \((z_{t1}, \ldots, z_{tN})\) is a sample from the standard Gaussian distribution, \( x_0 = 10, g_t = 2.8, \) and \( h_t = 0.2. \)

**Illustrating the correctness of DP equations (4.17), (4.18), (4.19), and checking the convergence of the variants of Dual SDDP.** We solve this inventory problem using Dynamic Programming

\[\text{We again suppress the dependence of the solution with respect to } t, k \text{ to alleviate notation}\]
applied both to DP equations (4.17)-(4.18) and to DP equations (4.25)-(4.26) for $\gamma = 1, 10, 1000$. In this latter case, we obtain approximations of functions $V_1^\gamma$. We also run Primal SDDP, Dual SDDP with feasibility cuts, and Dual SDDP with penalties $q_k = r_k = 1, 10, 1000$, on the same instance, knowing that Dual SDDP variants were run for 100 iterations (the upper bounds computed by these methods stabilize in less than 10 iterations) and Primal SDDP was stopped when the gap is $< 0.1$ where the gap is defined as $\frac{Ub - Lb}{Ub}$, where $Ub$ and $Lb$ correspond to upper and lower bounds computed by Primal SDDP along iterations. The lower bound $Lb$ is the optimal value of the first stage problem and the upper bound $Ub$ is the upper end of a 97.5%-one-sided confidence interval on the optimal value obtained using the sample of total costs computed by all previous forward passes. With this stopping criterion and the considered instance of the inventory problem, Primal SDDP was run for 232 iterations.

In Figure 2, we report the graph of $V_2$ and the cuts computed for $V_2$ by Dual SDDP with feasibility cuts (right panel), Dual SDDP with penalties $q_k = r_k = 1000$ (left panel), and Dual SDDP with penalties $q_k = r_k = 100$ (middle panel). All cuts are, as expected, upper bounding affine functions for $V_2$ on its domain. However, it is interesting to notice that for Dual SDDP with feasibility cuts, few different cuts are computed and these cuts are tangent or very close to $V_2$. On the contrary, the variants of Dual SDDP with penalties, especially for penalties $q_k = r_k = 100$, compute many cuts that are dominated by others on the domain of $V_2$. Therefore, cut selection techniques, for instance along the lines of [9] [12] using Limited Memory Level 1 cut selection, could be useful for Dual SDDP.

We report in Table 1 the approximate optimal values and the time needed to compute them with Primal SDDP, Dual SDDP, and Dynamic Programming applied to respectively (4.17), (4.18), (4.19) and (4.25), (4.26), (4.27) with $\gamma = 1, 100, 1000$. The approximate optimal values reported are the last upper bound computed for variants of Dual SDDP and the last lower bound computed for Primal SDDP. All approximate optimal values are very close (showing that all variants were correctly implemented) and Dynamic Programming is much slower than the other sampling-based algorithms. For Dual SDDP with penalization, if penalties are too small the upper bound can be $+\infty$ while if penalties are sufficiently large the algorithm converges to an optimal policy.

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

**Tests on a larger instance.** We now run Primal and Dual SDDP on a larger instance with $T = N_t = 100$ for 600 iterations. The evolution of the upper bounds computed along the iterations of Dual SDDP (both with feasibility cuts and with penalizations $q_k = r_k = 1000$) and of the upper and lower bounds computed by Primal SDDP for the first 100 iterations are represented in Figure 4. We also report in Table 2 the values of these bounds for iterations 2, 3, 5, 10, 50, 100, 200, 300, 400, 500, and 600. We see that for the
first iterations, the upper bound decreases more quickly with the variants of Dual SDDP, the most important decrease being obtained for Dual SDDP with feasibility cuts. However, on this instance, the convergence of Dual SDDP with feasibility cuts is slower, i.e., a solution of high accuracy is obtained quicker using Dual SDDP with penalizations. More precisely, we fix confidence levels \( \varepsilon = 0.2, 0.15, 0.1, 0.05, 0.01 \), and for each confidence level, we compute the time needed, running Primal and Dual SDDP in parallel, to obtain a solution with relative accuracy \( \varepsilon \) stopping the algorithm when the upper bound \( \text{UB}_D \) computed by a variant of Dual SDDP and the lower bound \( \text{LB} \), computed by Primal SDDP, satisfies \( (\text{UB}_D - \text{LB})/\text{UB}_D < \varepsilon \). The results are reported in Table 3. In this table, we also report the time needed to obtain a solution of relative accuracy \( \varepsilon \) using only the information provided by Primal SDDP, stopping the algorithm when \( (\text{UB} - \text{LB})/\text{UB} < \varepsilon \).

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

### Table 1

<table>
<thead>
<tr>
<th>Method</th>
<th>Optimal value</th>
<th>CPU time (s.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Programming on (4.17), (4.18), (4.19)</td>
<td>321.6</td>
<td>685</td>
</tr>
<tr>
<td>Dynamic Programming on (4.25), (4.26), (4.27), ( \gamma = 1 )</td>
<td>+( \infty )</td>
<td>2 860</td>
</tr>
<tr>
<td>Dynamic Programming on (4.25), (4.26), (4.27), ( \gamma = 100 )</td>
<td>322.2</td>
<td>3 808</td>
</tr>
<tr>
<td>Dynamic Programming on (4.25), (4.26), (4.27), ( \gamma = 1000 )</td>
<td>321.8</td>
<td>3 376</td>
</tr>
<tr>
<td>Primal SDDP</td>
<td>322.5</td>
<td>105</td>
</tr>
<tr>
<td>Dual SDDP with penalties, ( q_{tk} = r_{tk} = 1 )</td>
<td>2 131.4</td>
<td>9.4</td>
</tr>
<tr>
<td>Dual SDDP with penalties, ( q_{tk} = r_{tk} = 100 )</td>
<td>322.5</td>
<td>11.3</td>
</tr>
<tr>
<td>Dual SDDP with penalties, ( q_{tk} = r_{tk} = 1000 )</td>
<td>322.5</td>
<td>11.9</td>
</tr>
<tr>
<td>Dual SDDP with feasibility cuts</td>
<td>322.5</td>
<td>10.6</td>
</tr>
</tbody>
</table>

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

**Fig. 3.** Left: upper and lower bounds computed by Primal SDDP and upper bounds computed by Dual SDDP with feasibility cuts and Dual SDDP with penalties \( q_{tk} = r_{tk} = 1000 \) for the first 10 iterations. Right: same outputs for iterations 10, \ldots, 100.
In Figure 5, we report the cumulative CPU time along iterations of all methods. We see that each iteration requires a similar computational bulk and the CPU time increases exponentially with the number of iterations.

Finally, for Dual SDDP with penalizations, we report in Figure 8 in the Appendix the maximal and mean values of $\nu_k^{+} + \nu_k^{-}$ (recall that for the inventory problem $\nu_k^{+}, \nu_k^{-}$ in (4.22) are real-valued) along iterations.
for each stage $t$, where $\nu^+_k, \nu^-_k$ are optimal solution of $\nu^+_t, \nu^-_t$ in (4.22) for iteration $k$. The corresponding values are positive, meaning that indeed RCR does not hold for the dual of the inventory problem.

5.2. Sensitivity analysis for the inventory problem. Consider the inventory problem of Section 5.1 as in (3.4) and $T = 10$ stages. Our goal is to compute the derivatives in (3.11) solving the primal and dual problems by respectively Primal and Dual SDDP. We consider 4 instances with $(\phi, \mu) = (0.01, 0.1), (0.01, 3.0), (0.001, 0.1), \text{and} (0.001, 3.0)$. The remaining parameters of these instances are those from the previous section. We discretize both the primal and dual problem into $N_t = 100$ samples for each stage $t = 2, \ldots, 10$. We take the relative error $\varepsilon = 0.01$ for the stopping criterion and use 10000 Monte Carlo simulations to estimate the expectations in (3.11). For Primal SDDP, the upper bound $U_b$ and lower bound $L_b$ at termination are given in Table 4 for the four instances.

<table>
<thead>
<tr>
<th>Bound</th>
<th>Instance 1</th>
<th>Instance 2</th>
<th>Instance 3</th>
<th>Instance 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_b$</td>
<td>17.9176</td>
<td>478.687</td>
<td>15.3940</td>
<td>404.242</td>
</tr>
<tr>
<td>$L_b$</td>
<td>17.9163</td>
<td>475.017</td>
<td>15.3927</td>
<td>402.913</td>
</tr>
</tbody>
</table>

**Table 4**

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

The optimal mean values of Lagrangian multipliers for the demand constraints computed, for a given stage $t \geq 2$, averaging over the 10000 values obtained simulating 10000 forward passes after termination, are given in Table 5. In this table, $\text{LM}$ stands for the multipliers obtained using Primal SDDP as explained in Remark 2.4 whereas $\text{Dual}$ stands for the multipliers obtained using Dual SDDP with penalties. The fact that the multipliers obtained are close for both methods illustrates the validity of the two alternatives we discussed in Sections 3-4 to compute derivatives of the value function of a MSP.

With optimal dual solutions $\{\bar{\pi}_t\}$ and the realizations of $\{D_t\}$ and $\{\epsilon_t\}$ at hand, we are able to compute the sensitivity of the optimal value with respect to $\phi$ and $\mu$, using (3.11) and (3.12), with expectations estimated for 10000 Monte Carlo simulations. We benchmark our method against the finite-difference method. Specifically, for value function $\vartheta$, the finite-difference method approximates the derivative with respect to $u_0$ by $v'(u_0) \approx \frac{v(u_0+\delta)-v(u_0-\delta)}{2\delta}$ for some small $\delta$.

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

We observe that the derivatives obtained by both methods are close to each other, especially when
Table 5
Comparison between optimal Lagrange multipliers from Primal SDDP and Dual SDDP with penalties.

<table>
<thead>
<tr>
<th>Instance</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LM Dual</td>
<td>LM Dual</td>
<td>LM Dual</td>
<td>LM Dual</td>
</tr>
<tr>
<td>2</td>
<td>0.2465</td>
<td>0.2373</td>
<td>0.0444</td>
<td>0.0328</td>
</tr>
<tr>
<td>3</td>
<td>0.3218</td>
<td>0.31095</td>
<td>0.1421</td>
<td>0.1340</td>
</tr>
<tr>
<td>4</td>
<td>0.3268</td>
<td>0.3221</td>
<td>0.19439</td>
<td>0.18974</td>
</tr>
<tr>
<td>5</td>
<td>0.3086</td>
<td>0.3058</td>
<td>0.2145</td>
<td>0.2128</td>
</tr>
<tr>
<td>6</td>
<td>0.3408</td>
<td>0.3412</td>
<td>0.1421</td>
<td>0.1340</td>
</tr>
<tr>
<td>7</td>
<td>0.5026</td>
<td>0.5051</td>
<td>0.4378</td>
<td>0.4418</td>
</tr>
<tr>
<td>8</td>
<td>0.7047</td>
<td>0.7049</td>
<td>0.6327</td>
<td>0.63397</td>
</tr>
<tr>
<td>9</td>
<td>0.8985</td>
<td>0.9032</td>
<td>1.0315</td>
<td>1.0343</td>
</tr>
<tr>
<td>10</td>
<td>1.1022</td>
<td>1.1037</td>
<td>1.2302</td>
<td>1.2368</td>
</tr>
</tbody>
</table>

Table 6
Sensitivity of the optimal value with respect to $\phi$ and $\mu$ by the two methods.

<table>
<thead>
<tr>
<th>Instance</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S- $\phi$</td>
<td>S-$\mu$</td>
<td>S-gap-$\phi$(%)</td>
<td>S-gap-$\mu$(%)</td>
</tr>
<tr>
<td>1</td>
<td>403.604</td>
<td>164.578</td>
<td>0.622</td>
<td>164.158</td>
</tr>
<tr>
<td>2</td>
<td>10 716.111</td>
<td>185.346</td>
<td>0.419</td>
<td>184.847</td>
</tr>
<tr>
<td>3</td>
<td>269.514</td>
<td>134.646</td>
<td>0.026</td>
<td>134.463</td>
</tr>
<tr>
<td>4</td>
<td>7 780.570</td>
<td>158.017</td>
<td>0.132</td>
<td>158.001</td>
</tr>
</tbody>
</table>

5.3. Dual SDDP for an hydro-thermal generation problem. We repeat the experiments of Section 5.1 for the Brazilian interconnected power system problem given in [29] for $T = 12$ stages and $N_t = 50$ inflow realizations for every stage. These realizations are obtained calibrating log-normal distributions for each month of the year using historical data of inflows and sampling from these distributions. The data used for these simulations (including the inflow scenarios) is available on Github\(^5\).

We solve this problem using Primal SDDP and Dual SDDP with penalization (the variant of Dual SDDP presented in Section 6.2). For this variant of Dual SDDP, a general procedure to define sequences of penalizations $(q_{tk})$, $(r_{tk})$ ensuring convergence of the corresponding Dual SDDP method is to take $q_{tk} = r_{tk} = \gamma_0 \alpha^{k-1}$, $k \geq 1$, $t = 2, \ldots, T$, with $\alpha > 1$, $\gamma_0 > 0$. For numerical reasons, we also take a large upper bound $U$ for these sequences and use

\[
q_{tk} = r_{tk} = \min(U, \gamma_0 \alpha^{k-1}), \quad k \geq 1, t = 2, \ldots, T.
\]

We consider three variants of Dual SDDP: for the first variant, denoted by Dual SDDP 1, $q_{tk} = r_{tk}$ are as in (5.1) with $\gamma_0 = 10^4$, $\alpha = 1.3$, $U = 10^{10}$. To illustrate the fact that for constant sequences $q_{tk}, r_{tk} = \gamma_0$, Dual SDDP converges (resp. does not converge) for sufficiently large constants $\gamma_0$ (resp. sufficiently small constants $\gamma_0$) we also define two other variants corresponding to $U = +\infty$, $\gamma_0 = 10^p$, $\alpha = 1$, and $U = +\infty$, $\gamma_0 = 10^q$, $\alpha = 1$, in (5.1), respectively denoted by Dual SDDP 2 and Dual SDDP 3.

---

\(^5\)https://github.com/vguigues/Primal_SDDP_Library_Matlab
We run Dual SDDP for 1000 iterations and Primal SDDP for 3000 iterations. The evolution of the upper and lower bounds computed by the methods for the first 1000 iterations is given in Figure 6. More precisely, the values of these bounds for iterations 2, 5, 10, 50, 100, 150, 200, 250, 300, 350, 400, 1000, and 3000 are reported in Table 7. We observe that parameter $\gamma_0$ for Dual SDDP 3 is too small to allow this method to converge to the optimal value of the problem whereas the other two variants Dual SDDP 1 and Dual SDDP 2 of Dual SDDP converge. Naturally, these methods start with large upper bounds but after a few tens of iterations the upper bounds with Dual SDDP 1 and Dual SDDP 2 are better than the upper bound computed by Primal SDDP. In particular, it is interesting to notice that the best (lowest) upper bounds are obtained with the variant of Dual SDDP that uses adaptive penalizations, i.e., penalizations that increase with the number of iterations before reaching value $U$ in (5.1).

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

However, each iteration of Dual SDDP takes more time as can be seen in Figure 7 which reports the cumulative CPU time for all methods. More precisely, running Dual and Primal SDDP in parallel, we can compute the time needed to obtain a solution of relative accuracy $\varepsilon$ using the standard stopping criterion $\varepsilon$

---

6The upper bounds for Primal SDDP are computed as explained in Section 5.1.
<table>
<thead>
<tr>
<th>Iteration</th>
<th>Lb Primal SDDP</th>
<th>Ub Primal SDDP</th>
<th>Dual SDDP 1</th>
<th>Dual SDDP 2</th>
<th>Dual SDDP 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$1.317 \times 10^8$</td>
<td>$1.4398 \times 10^8$</td>
<td>$1.0002 \times 10^9$</td>
<td>$1.0002 \times 10^9$</td>
<td>$1.0002 \times 10^9$</td>
</tr>
<tr>
<td>5</td>
<td>$5.558 \times 10^5$</td>
<td>$1.0936 \times 10^8$</td>
<td>$1.0002 \times 10^9$</td>
<td>$1.0002 \times 10^9$</td>
<td>$9.9404 \times 10^8$</td>
</tr>
<tr>
<td>10</td>
<td>$1.4032 \times 10^7$</td>
<td>$8.1728 \times 10^7$</td>
<td>$3.6040 \times 10^8$</td>
<td>$1.0002 \times 10^9$</td>
<td>$4.9508 \times 10^8$</td>
</tr>
<tr>
<td>50</td>
<td>$2.3670 \times 10^7$</td>
<td>$4.1346 \times 10^7$</td>
<td>$5.4999 \times 10^7$</td>
<td>$1.0002 \times 10^9$</td>
<td>$9.6720 \times 10^7$</td>
</tr>
<tr>
<td>100</td>
<td>$2.4787 \times 10^7$</td>
<td>$3.5502 \times 10^7$</td>
<td>$3.6322 \times 10^7$</td>
<td>$6.4072 \times 10^7$</td>
<td>$8.2494 \times 10^7$</td>
</tr>
<tr>
<td>150</td>
<td>$2.5111 \times 10^7$</td>
<td>$3.2447 \times 10^7$</td>
<td>$3.0685 \times 10^7$</td>
<td>$3.5595 \times 10^7$</td>
<td>$7.9465 \times 10^7$</td>
</tr>
<tr>
<td>200</td>
<td>$2.5249 \times 10^7$</td>
<td>$3.0672 \times 10^7$</td>
<td>$2.9076 \times 10^7$</td>
<td>$3.0404 \times 10^7$</td>
<td>$7.8059 \times 10^7$</td>
</tr>
<tr>
<td>250</td>
<td>$2.5374 \times 10^7$</td>
<td>$3.0079 \times 10^7$</td>
<td>$2.8215 \times 10^7$</td>
<td>$2.8943 \times 10^7$</td>
<td>$7.6917 \times 10^7$</td>
</tr>
<tr>
<td>300</td>
<td>$2.5436 \times 10^7$</td>
<td>$2.9434 \times 10^7$</td>
<td>$2.7710 \times 10^7$</td>
<td>$2.8030 \times 10^7$</td>
<td>$7.6344 \times 10^7$</td>
</tr>
<tr>
<td>350</td>
<td>$2.5477 \times 10^7$</td>
<td>$2.9014 \times 10^7$</td>
<td>$2.7309 \times 10^7$</td>
<td>$2.7532 \times 10^7$</td>
<td>$7.5852 \times 10^7$</td>
</tr>
<tr>
<td>400</td>
<td>$2.5526 \times 10^7$</td>
<td>$2.8626 \times 10^7$</td>
<td>$2.7110 \times 10^7$</td>
<td>$2.7188 \times 10^7$</td>
<td>$7.5526 \times 10^7$</td>
</tr>
<tr>
<td>1000</td>
<td>$2.5703 \times 10^7$</td>
<td>$2.7175 \times 10^7$</td>
<td>$2.6304 \times 10^7$</td>
<td>$2.6335 \times 10^7$</td>
<td>$7.4292 \times 10^7$</td>
</tr>
<tr>
<td>3000</td>
<td>$2.5798 \times 10^7$</td>
<td>$2.6883 \times 10^7$</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7: Relative error as a function of the number of iterations for Primal SDDP, Dual SDDP 1, and Dual SDDP 2.

For an instance of the hydro-thermal problem with $T = 12$, $N_t = 50$, lower bound $Lb$ and upper bound $Ub$ computed by Primal SDDP and upper bounds computed by variants of Dual SDDP along iterations.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Primal SDDP</th>
<th>Dual SDDP 1</th>
<th>Dual SDDP 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.30</td>
<td>0.32</td>
<td>0.61</td>
</tr>
<tr>
<td>200</td>
<td>0.18</td>
<td>0.13</td>
<td>0.17</td>
</tr>
<tr>
<td>300</td>
<td>0.14</td>
<td>0.08</td>
<td>0.09</td>
</tr>
<tr>
<td>400</td>
<td>0.11</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>500</td>
<td>0.09</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>800</td>
<td>0.07</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>1000</td>
<td>0.05</td>
<td>0.02</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 8: Relative error as a function of the number of iterations for Primal SDDP, Dual SDDP 1, and Dual SDDP 2 for Primal SDDP (see [24]) or using the lower bound from Primal SDDP and the upper bound from Dual SDDP, and computing the relative error obtained with these bounds each time a new bound (either lower bound or upper bound) is computed. The results are reported in Table 9. We see that due to the fact that Dual SDDP iterations are more time consuming, for all relative accuracies but one, the use of the stopping criterion based on Dual SDDP upper bounds requires more computational bulk. From this experiment, performed on a larger problem (in terms of size of the state vector and number of control variables for each stage) than the inventory problem of Section 5.1, it seems that the use of Dual SDDP for a stopping criterion of Primal SDDP will decrease the overall computational bulk only for small problems (having a limited to small number of controls, state variables, and scenarios).

Finally, as an evidence of the fact that RCR still does not hold for the dual of the hydro-thermal problem, we report in Figure 9 in the Appendix the evolution of the mean and maximal (computed across iterations) constraint violations for all stages $t = 2, 3, \ldots, 12$. For most stages, these violations are null or small but for stages 2 and 12 they can be large.

REFERENCES

Fig. 7. Cumulative CPU time for Primal SDDP, Dual SDDP 1, Dual SDDP 2, and Dual SDDP 3.

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

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

6. Appendix.

6.1. Dual SDDP: algorithm and convergence proof. The Dual SDDP method for problem (4.15) is given below.

**Dual SDDP for DP equations (4.17), (4.18), (4.19).**

**Initialization.** For \( t = 2, \ldots, T \), take \( V_0^t \equiv +\infty \), i.e., take \( \bar{\theta}_i^0 = +\infty \), \( \bar{\beta}_i^0 = 0 \), and \( \bar{\sigma}_i^0 = 0 \) for \( t = 2, \ldots, T \). Set iteration counter \( k \) to 1.

**Step 1: forward pass (computation of dual trial points).** Compute an optimal solution \( \pi_1^k, \mu_1^k \) of

\[
V^{k-1} = \max_{(\pi_1, \mu_1) \in B_1 \cap D_1} b_1^T \pi_1 + f_1^T \mu_1 + V_2^{k-1}(\pi_1, \mu_1).
\]

For \( t = 2, \ldots, T - 1 \), given \( (\pi_{t-1}^k, \mu_{t-1}^k) \), compute an optimal solution of

\[
\max_{\pi_t, \mu_t} \sum_{j=1}^{N_t} p_{tj} \left( b_{tj}^T \pi_{tj} + \mu_{tj}^T f_{tj} + V_{t+1}^{k-1}(\pi_{tj}, \mu_{tj}) \right) \in B_t \cap D_t(\pi_{t-1}^k, \mu_{t-1}^k).
\]

An optimal solution of the problem above has \( N_t \) components \( (\pi_{t1}, \mu_{t1}), (\pi_{t2}, \mu_{t2}), \ldots, (\pi_{tN_t}, \mu_{tN_t}) \). We take \((\pi_t^k, \mu_t^k)\) to be one of these components knowing that component \( i \) is sampled with probability \( p_{ti} \).

**Step 2: backward pass (computation of new cuts).** Let \((\alpha, \lambda, \delta)\) be an optimal solution of\(^7\)

\[
V_T(\pi_{T-1}^k, \mu_{T-1}^k) = \left\{ \min_{\alpha, \lambda, \delta} \delta^T (c_{T-1} - A_{T-1}^T \pi_{T-1}^k - D_{T-1}^T \mu_{T-1}^k) + c_j^T \sum_{j=1}^{N_T} \alpha_j + d_j^T \sum_{j=1}^{N_T} \lambda_j \right\}
\]

\[
A_T \alpha_j + p_{Tj} B_T \delta + C_T \lambda_j = p_{Tj} b_T, \ j = 1, \ldots, N_T, \]

\[
p_{Tj} f_{Tj} \geq D_T \alpha_j + F_T \lambda_j + p_{Tj} E_T \delta, \ j = 1, \ldots, N_T.
\]

Compute the new cut coefficients

\[
\bar{\theta}_T^k = \delta^T c_{T-1} + c_j^T \sum_{j=1}^{N_T} \alpha_j + d_j^T \sum_{j=1}^{N_T} \lambda_j, \quad \bar{\sigma}_T^k = -A_{T-1} \delta, \quad \bar{\beta}_T^k = -D_{T-1} \delta.
\]

\(^7\)We suppressed the dependence of the optimal solution on \( T \) and \( k \) to alleviate notation.
For $t = T - 1, \ldots, 2$, compute an optimal solution $(\nu, \lambda, \delta)$ of

$$
\min_{\nu, \lambda, \delta} \delta^T [c_{t-1} - A_{t-1}^T \pi_{t-1}^k - D_{t-1}^T \mu_{t-1}] + d_t^T \sum_{j=1}^{N_t} \lambda_j + \sum_{i=0}^{k} \bar{\theta}_{i+1}^T \sum_{j=1}^{N_t} \nu_i(j)
$$

$$
C_t \lambda_j + p_{ij} B_t \delta - \sum_{i=0}^{k} \nu_i(j) \bar{\theta}_{i+1}^T = p_{ij} b_{ij}, \ j = 1, \ldots, N_t,
$$

$$
F_t \lambda_j + p_{ij} E_t \delta - \sum_{i=0}^{k} \nu_i(j) \bar{\theta}_{i+1}^T \leq p_{ij} f_{ij}, \ j = 1, \ldots, N_t,
$$

$$
\sum_{i=0}^{k} \nu_i(j) = p_{ij}, \ j = 1, \ldots, N_t,
$$

and the cut coefficients

$$
\bar{\theta}_{i+1}^T = \delta^T c_{t-1} + d_t^T \sum_{j=1}^{N_t} \lambda_j + \sum_{i=0}^{k} \bar{\theta}_{i+1}^T \sum_{j=1}^{N_t} \nu_i(j), \quad \bar{\theta}_{i+1}^T = -A_{t-1} \delta, \quad \bar{\theta}_{i+1}^T = -D_{t-1} \delta.
$$

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

**Proof of Theorem 4.2.** It suffices to combine Proposition 4.3 with the convergence proof of SDDP from [21].

**6.2. Analysis of Dual SDDP with penalizations.** To analyze Dual SDDP with penalizations, it is convenient to introduce the sequence of functions

$$
(6.1) \quad \nabla_T^k(\pi_{T-1}, \mu_{T-1}) := \left\{ \begin{array}{l}
\max_{\pi_T, \mu_T, \nu_T} \sum_{j=1}^{N_T} p_T j \left( b_T^T \pi_T j + \mu_T^T f_T j \right) - q_T^T \nu_T - r_T^T \nu_T
\end{array} \right.
$$

$$
A_T^T \pi_T j + D_T^T \mu_T j = c_T, \ j = 1, \ldots, N_T,
$$

$$
C_T^T \pi_T j + F_T^T \mu_T j = d_T, \ j = 1, \ldots, N_T,
$$

$$
\mu_T j \leq 0, \ \bar{\pi}_T \leq \pi_T, \ \bar{\mu}_T \leq \mu_T, \ j = 1, \ldots, N_T,
$$

$$
A_{T-1}^T \pi_{T-1} + D_{T-1}^T \mu_{T-1} + \sum_{j=1}^{N_T} p_T j \left( B_T^T \pi_T j + E_T^T \mu_T j \right) + \nu_T^+ - \nu_T^- = c_{T-1},
$$

$$
\nu_T^+, \nu_T^- \geq 0,
$$

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

$$
(6.2) \quad \nabla_T^k(\pi_{t-1}, \mu_{t-1}) := \left\{ \begin{array}{l}
\max_{\pi_t, \mu_t, \nu_t} \sum_{j=1}^{N_t} p_T j \left( b_T^T \pi_T j + \mu_T^T f_T j + \nu_T^+ \left( \pi_T j, \mu_T j \right) \right) - q_T^T \nu_T - r_T^T \nu_T
\end{array} \right.
$$

$$
C_T^T \pi_T j + F_T^T \mu_T j = d_T, \ j = 1, \ldots, N_t,
$$

$$
\mu_T j \leq 0, \ \bar{\pi}_T \leq \pi_T, \ \bar{\mu}_T \leq \mu_T, \ j = 1, \ldots, N_t,
$$

$$
A_{t-1}^T \pi_{t-1} + D_{t-1}^T \mu_{t-1} + \sum_{j=1}^{N_t} p_T j \left( B_T^T \pi_T j + E_T^T \mu_T j \right) + \nu_T^+ - \nu_T^- = c_{t-1},
$$

$$
\nu_T^+, \nu_T^- \geq 0.
$$

We will use the following lemma:

**Lemma 6.1.** Assume that problem (4.15) is feasible and has a finite optimal value. Then for $t = 1, \ldots, T - 1$, there exists $(\pi_t, \mu_t)$ such that

$$
C_T^T \pi_T + F_T^T \mu_T = d_T, \mu_T \leq 0,
$$
and there exists \((\pi_T, \mu_T)\) such that
\[
A_T^\top \pi_T + D_T^\top \mu_T = d_T, C_T^\top \pi_T + F_T^\top \mu_T = d_T, \mu_T \leq 0.
\]

Proof. Since the primal problem is linear, feasible, and has a finite optimal value then its dual, given by (4.16), is also feasible and has a finite optimal value which is the optimal value of the primal. The result follows recalling the constraints of dual problem (4.16). \(\square\)

Proof of Proposition 4.3. We show by induction on \(k\) that \(V_t \leq V_t^k\) for \(t = 2, \ldots, T\). For \(k = 0\) these relations hold by definition. Assume that for some \(k\) we have \(V_t \leq V_t^k\) for \(t = 2, \ldots, T\). We show by backward induction on \(t\) that \(V_t \leq V_t^{k+1}\) for \(t = 2, \ldots, T\). Observe that for any \((\pi_{T-1}, \mu_{T-1})\), optimization problem (6.1) with optimal value \(\overline{V}_T(\pi_{T-1}, \mu_{T-1})\) is feasible. Indeed, using Lemma 6.1, there exist points \((\pi_{Tj}, \mu_{Tj})\) satisfying the first three groups of constraints and for each such points we can find \(\nu_T^j, \nu_T \geq 0\) satisfying the fourth group of constraints. Therefore \(\overline{V}_T(\pi_{T-1}, \mu_{T-1})\) is finite for every \((\pi_{T-1}, \mu_{T-1})\) and is the optimal value of the corresponding dual optimization problem, i.e., for any \((\pi_{T-1}, \mu_{T-1})\) we get
\[
\overline{V}_T^k(\pi_{T-1}, \mu_{T-1}) = \begin{cases}
\min_{\alpha, \lambda, \delta} & \delta^\top (c_{T-1} - A_{T-1}^\top \pi_{T-1} - D_{T-1}^\top \mu_{T-1}) + c_T^\top \sum_{j=1}^{N_T} \alpha_j + d_T^\top \sum_{j=1}^{N_T} \lambda_j \\
A_{Tj} + p_{Tj}B_T \delta + c_T \lambda_j = p_{Tj}b_{Tj}, & j = 1, \ldots, N_T, \\
p_{Tj}F_T \geq D_T \alpha_j + F_T \lambda_j + p_{Tj}E_T \delta, & j = 1, \ldots, N_T, \\
-q_{Tk} \leq \delta \leq r_{Tk}.
\end{cases}
\]

Using this dual representation and the definition of \(\overline{g}_T^k, \overline{b}_T^k, \overline{c}_T^k\), we get for every \((\pi_{T-1}, \mu_{T-1})\):
\[
\overline{g}_T^k + \langle \overline{b}_T^k, \pi_{T-1} \rangle + \langle \overline{c}_T^k, \mu_{T-1} \rangle \geq \overline{V}_T^k(\pi_{T-1}, \mu_{T-1}).
\]

Recalling representation (6.1) for \(\overline{V}_T^k(\pi_{T-1}, \mu_{T-1})\), observe that for every \((\pi_{T-1}, \mu_{T-1}) \in \text{dom}(V_T)\) we have \(\overline{V}_T^k(\pi_{T-1}, \mu_{T-1}) \geq V_T(\pi_{T-1}, \mu_{T-1})\) whereas for \((\pi_{T-1}, \mu_{T-1}) \notin \text{dom}(V_T)\) we have \(V_T(\pi_{T-1}, \mu_{T-1}) = -\infty\) while \(\overline{V}_T^k(\pi_{T-1}, \mu_{T-1})\) is finite, which shows that for every \((\pi_{T-1}, \mu_{T-1})\) we have \(\overline{V}_T^k(\pi_{T-1}, \mu_{T-1}) \geq V_T(\pi_{T-1}, \mu_{T-1})\), which, combined with (6.3) and the induction hypothesis, gives
\[
V_T^k(\pi_{T-1}, \mu_{T-1}) \geq V_T(\pi_{T-1}, \mu_{T-1})
\]
for every \((\pi_{T-1}, \mu_{T-1})\).

Now assume that \(V_{t+1}^k(\pi_t, \mu_t) \geq V_{t+1}(\pi_t, \mu_t)\) for all \((\pi_t, \mu_t)\) for some \(t \in \{2, \ldots, T-1\}\). We want to show that \(V_k^k(\pi_{t-1}, \mu_{t-1}) \geq V_k(\pi_{t-1}, \mu_{t-1})\) for all \((\pi_{t-1}, \mu_{t-1})\). First observe that for every \((\pi_{t-1}, \mu_{t-1})\), linear program (6.2) with optimal value \(\overline{V}_t^k(\pi_{t-1}, \mu_{t-1})\) is feasible and has a finite optimal value (see Lemma 6.1).

Therefore we can express \(\overline{V}_t^k(\pi_{t-1}, \mu_{t-1})\) as the optimal value of the corresponding dual problem, i.e.,
\[
\overline{V}_t^k(\pi_{t-1}, \mu_{t-1}) = \begin{cases}
\min_{\nu, \lambda, \delta} & \delta^\top \left[c_{t-1} - A_{t-1}^\top \pi_{t-1} - D_{t-1}^\top \mu_{t-1} \right] + d_T^\top \sum_{j=1}^{N_t} \lambda_j + \sum_{i=0}^{k} \sum_{j=1}^{N_t} \nu_i(j) \\
C_t \lambda_j + p_{tj}B_t \delta - \sum_{i=0}^{k} \nu_i(j) \overline{c}_{t+1}^i = p_{tj}b_{tj}, & j = 1, \ldots, N_t, \\
F_t \lambda_j + p_{tj}E_t \delta - \sum_{i=0}^{k} \nu_i(j) \overline{c}_{t+1}^i \leq p_{tj}f_{tj}, & j = 1, \ldots, N_t, \\
\sum_{i=0}^{k} \nu_i(j) = p_{tj}, & j = 1, \ldots, N_t, \\

\nu_0, \ldots, \nu_k \geq 0, -q_{tk} \leq \delta \leq r_{tk}.
\end{cases}
\]
Using this representation of $V^k_t$ and the definition of $\overline{\theta}^k_t, \overline{\beta}^k_t, \overline{\gamma}^k_t$, we obtain for every $(\pi_{t-1}, \mu_{t-1})$:

$$\overline{\theta}^k_t + \overline{\beta}^k_t (\pi_{t-1}) + \overline{\gamma}^k_t (\mu_{t-1}) \geq V^k_t (\pi_{t-1}, \mu_{t-1}).$$

(6.4)

Next, recalling representation (6.2) for $V_t (\pi_{t-1}, \mu_{t-1})$ and the induction hypothesis, we get

$$V^k_t (\pi_{t-1}, \mu_{t-1}) \geq \hat{V}^k_t (\pi_{t-1}, \mu_{t-1}),$$

where

$$\hat{V}^k_t (\pi_{t-1}, \mu_{t-1}) = \begin{cases} \max_{\nu_t \geq 0, j = 1, \ldots, N_t,} & \sum_{j=1}^{N_t} p_{tj} \left(b_{tj}^T \pi_{tj} + \mu_{tj}^T f_{tj} + V_{t+1} (\pi^t_{t}, \mu^t_{t}) \right) \geq \sum_{j=1}^{N_t} q_{tj} \nu^t_j + r_{tj}^T \mu^t_j \nu^t_j, \\ \mu_{tj} \leq 0, & j = 1, \ldots, N_t, \\ A_{t-1}^T \pi_{t-1} + D_{t-1}^T \mu_{t-1} + \sum_{j=1}^{N_t} p_{tj} \left(B_{tj}^T \pi_{tj} + E_{tj}^T \mu_{tj} \right) + \nu^t_j - \nu^t_j = c_{t-1}, \\ \nu^t_j, \nu^t_j \geq 0. \end{cases}$$

Similarly to the induction step $t = T$, for every $(\pi_{t-1}, \mu_{t-1})$, we have

$$\hat{V}^k_t (\pi_{t-1}, \mu_{t-1}) \geq V_t (\pi_{t-1}, \mu_{t-1}).$$

Combining (6.4), (6.5), and (6.6) with the induction hypothesis, we obtain $V^k_t (\pi_{t-1}, \mu_{t-1}) \geq V_t (\pi_{t-1}, \mu_{t-1})$ for all $(\pi_{t-1}, \mu_{t-1})$ which achieves the proof of the induction step $t$.

In particular $V^2_{k-1} \geq V^2$ which implies that $V^{k-1}$ is greater than or equal to the optimal value of (4.15).

**Proof of Theorem 4.4.** Let us take an arbitrary realization of the algorithm and let us show that $V^k$ converges to the optimal value of (4.15) for that realization. Observe that there is some iteration $k_0$ such that for all iterations $k \geq k_0$, the functions $V^k_t$ and $V_t$ coincide on the domain of $V_t$ for all stage $t$, due to the fact that $\lim_{k \to +\infty} q_{tk} = \lim_{k \to +\infty} r_{tk} = +\infty$. From that iteration on, we can follow the convergence proof of SDDP from [21] using Proposition 4.3 for the validity of the cuts computed.

**6.3. Analysis of Dual SDDP with feasibility cuts. Validity of the cuts computed: proof of Proposition 4.5.** (ii) follows from the proof of Proposition 4.3. Let us prove (i). Let us first show that all subproblems solved in the forward passes are feasible and that the set of feasible $\pi_t, \mu_t$ (across nodes of stage $t$) are contained in $S_t$. Consider the linear program

$$\begin{align*}
\min_{\pi_t, \mu_t, \nu^+, \nu^-} & \quad e^T (\nu^+ + \nu^-) \\
A_{t-1}^T \pi_{t-1} + D_{t-1}^T \mu_{t-1} & + \sum_{j=1}^{N_T} p_{tj} (B_{tj}^T \pi_{tj} + E_{tj}^T \mu_{tj}) + \nu^+ - \nu^- = c_{t-1},
\end{align*}$$

(6.7)

where $e$ is a vector of ones. Clearly this problem is feasible and the optimal value of this problem is finite. Therefore the optimal value of this problem is also the optimal value its dual which is given by

$$\begin{align*}
\max_{\delta, \alpha, \lambda} & \quad \sum_{j=1}^{N_T} c_{tj} \alpha_j + d_{tj} \lambda_j + \delta^T (c_{t-1} - A_{t-1}^T \pi_{t-1} - D_{t-1}^T \mu_{t-1}) \\
p_{tj} B_{tj} \delta + A_{tj} \alpha_j + C_{tj} \lambda_j & = 0, \quad j = 1, \ldots, N_T, \\
p_{tj} E_{tj} \delta + D_{tj} \alpha_j + F_{tj} \lambda_j & \geq 0, \quad j = 1, \ldots, N_T, \\
-e \leq \delta \leq e.
\end{align*}$$

(6.8)
Recalling definition (4.29) of $\tilde{V}_T$, the optimal value of both (6.7) and (6.8) is $\tilde{V}_T(\pi_{T-1}, \mu_{T-1})$. Moreover, if $\pi_{T-1}, \mu_{T-1}$ is feasible for stage $T-1$ then necessarily $\tilde{V}_T(\pi_{T-1}, \mu_{T-1}) = 0$. Therefore, either $\tilde{V}_T(\pi^k_{T-1}, \mu^k_{T-1}) = 0$ and $\pi^k_{T-1}, \mu^k_{T-1}$ is feasible for stage $T-1$ or $\tilde{V}_T(\pi^k_{T-1}, \mu^k_{T-1}) > 0$ and in this case using the convexity of $\tilde{V}_T$, if $\pi_{T-1}, \mu_{T-1}$ is feasible then

$$0 = \tilde{V}_T(\pi_{T-1}, \mu_{T-1}) \geq \sum_{j=1}^{N_T} c^T_{T-1} \alpha_j + d^T_T \lambda_j + \delta^T (c_{T-1} - A^T_{T-1} \pi_{T-1} - D^T_{T-1} \mu_{T-1})$$

where $\alpha, \lambda, \delta$ is an optimal solution to (4.29) and we obtain the feasibility cut given in the algorithm. This feasibility cut is added to the problem of the previous stage and the subproblem for this previous stage is re-solved with this feasibility cut added.

Now for a given stage $t < T$, consider the linear program

$$\begin{align*}
\min_{\pi_t, \mu_t, \nu^+, \nu^-} & \quad e^T (\nu^+ + \nu^-) \\
A^T_{t-1} \pi_{t-1} + + D^T_{t-1} \mu_{t-1} + & \quad \sum_{j=1}^{N_t} p_{tj} (B^T_{tj} \tilde{\pi}_{tj} + E^T_{tj} \mu_{tj}) + \nu^+ - \nu^- = c_{t-1}, \\
C^T_{tj} \tilde{\pi}_{tj} + F^T_{tj} \mu_{tj} & \quad = d_t, \quad j = 1, \ldots, N_t, \\
C^T_t \pi_t + F^T_t \mu_t & \quad \leq \delta_t, \quad j = 1, \ldots, N_t, \\
\mu_{tj} & \quad \leq 0, \quad j = 1, \ldots, N_T, \\
\nu^+, \nu^- & \quad \geq 0,
\end{align*}$$

(6.9)

where $e$ is a vector of ones. Clearly this problem is feasible and the optimal value of this problem is finite. Therefore the optimal value of this problem is also the optimal value its dual which is given by

$$\begin{align*}
\max_{\delta, \alpha, \lambda} & \quad \sum_{j=1}^{N_t} d^T_t \lambda_j + d^T_T \lambda_j + \delta^T (c_{t-1} - A^T_{t-1} \pi_{t-1} - D^T_{t-1} \mu_{t-1}) \\
p_{tj} B^T_t \delta + C^T_t \lambda_j + \tilde{C}^T_t \lambda_j & \quad = 0, \quad j = 1, \ldots, N_t, \\
p_{tj} E^T_t \delta + F^T_t \lambda_j + \tilde{F}^T_t \lambda_j & \quad \geq 0, \quad j = 1, \ldots, N_t, \\
-\epsilon \leq \delta & \leq \epsilon, \quad \lambda_j \leq 0, \quad j = 1, \ldots, N_t.
\end{align*}$$

(6.10)

Recalling definition (4.30) of $\tilde{V}_t$, the optimal value of both (6.9) and (6.10) is $\tilde{V}_t(\pi_{t-1}, \mu_{t-1})$. Moreover, if $\pi_{t-1}, \mu_{t-1}$ is feasible for stage $t-1$ then necessarily $\tilde{V}_t(\pi_{t-1}, \mu_{t-1}) = 0$. Therefore, either $\tilde{V}_t(\pi^k_{t-1}, \mu^k_{t-1}) = 0$ and in this case $\pi^k_{t-1}, \mu^k_{t-1}$ yields a feasible dual state in stage $t$ and we solve problem (4.31), which is therefore feasible, to compute $\pi^k_t, \mu^k_t$. Or $\tilde{V}_t(\pi^k_{t-1}, \mu^k_{t-1}) > 0$ and in this case using the convexity of $\tilde{V}_t$, if $\pi_{t-1}, \mu_{t-1}$ is feasible then

$$0 = \tilde{V}_t(\pi_{t-1}, \mu_{t-1}) \geq \sum_{j=1}^{N_t} d^T_t \lambda_j + d^T_T \lambda_j + \delta^T (c_{t-1} - A^T_{t-1} \pi_{t-1} - D^T_{t-1} \mu_{t-1})$$

where $\alpha, \lambda, \delta$ is an optimal solution to (4.30) and we obtain the feasibility cut given in the algorithm. This feasibility cut is added to the problem of the previous stage and the subproblem for this previous stage is re-solved with this feasibility cut added. Finiteness of the number of feasibility cuts can be shown as in Theorem 5.1 in [8].

**Proof of Theorem 4.6.** The proof follows the steps of Theorem 5.1 in [8] and uses Proposition 4.3 for the validity of the cuts. □

### 6.4. Figures on constraint violation in Dual SDDP.
**Fig. 8.** Left: maximal constraint violation as a function of the stage. Right: Mean constraint violation as a function of the stage.

**Fig. 9.** Left: maximal constraint violation as a function of the stage for the hydro-thermal problem. Right: Mean constraint violation as a function of the stage for the hydro-thermal problem.