SIAM J. OPTIM.
Vol. 30, No. 4, pp. 2841–2865

DISTRIBUTIONALLY ROBUST STOCHASTIC DUAL DYNAMIC PROGRAMMING*

DANIEL DUQUE† AND DAVID P. MORTON†

Abstract. We consider a multistage stochastic linear program that lends itself to solution by stochastic dual dynamic programming (SDDP). In this context, we consider a distributionally robust variant of the model with a finite number of realizations at each stage. Distributional robustness is with respect to the probability mass function governing these realizations. We describe a computationally tractable variant of SDDP to handle this model using the Wasserstein distance to characterize distributional uncertainty.

Key words. distributionally robust optimization, multistage stochastic programming, stochastic dual dynamic programming

AMS subject classifications. 90C15, 90C39, 90C47

DOI. 10.1137/19M1309602

1. Introduction.

Distributionally robust optimization (DRO) is a paradigm in which the distribution governing the uncertain parameters of a mathematical optimization model is unknown, and yet the set of potential distributions can be characterized. In a typical stochastic program (SP), the goal is to minimize the expected value of a function that depends on both the decision variables and the random parameters. In DRO, the underlying model is formulated as a min-max problem in which the inner maximization is designed to make the outer problem's decisions robust, by selecting a worst-case probability distribution from a specified *distributional uncertainty set* (DUS).

The DRO paradigm naturally arises in the context of data-driven optimization in which the "true" distribution of the random parameters is unknown. See Rahimian and Mehrotra [35] for a recent review of DRO and its connections to risk aversion and regularization. The literature on DRO provides different representations of a DUS seeking to attain, for example, asymptotic and finite sample performance guarantees. There are approaches to build a DUS informed by data based on assumed moments of the distribution [10, 14, 42], likelihood functions [41], and goodness-of-fit measures from hypothesis tests [6, 5]. A related stream of DRO literature proposes building uncertainty sets in which candidate distributions are within a specified distance of the nominal distribution. Phi-divergence "distances" (e.g., Kullback–Leibler divergence, various χ^2 distances, total variation distance, among others) have been studied in the context of stochastic programming for the purpose of DRO [3]. Such approaches have also been used to identify important scenarios, and remove other scenarios, in static and two-stage models, using the total variation distance [33, 34].

*Received by the editors December 30, 2019; accepted for publication (in revised form) June 29, 2020; published electronically October 7, 2020.
https://doi.org/10.1137/19M1309602
Funding: This work was supported, in part, by the U.S. Department of Homeland Security under Grant Award 2017-ST-061-QA0001, and by Northwestern University's Center for Optimization and Statistical Learning. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.
†Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208 USA (danielduque@u.northwestern.edu, david.morton@northwestern.edu).

The Wasserstein metric has been extensively used in recent literature for various DRO problems; see, e.g., [8, 17, 28, 44]. A common aspect of these papers is that the uncertainty set includes continuous distributions. This offers additional flexibility compared to phi-divergence schemes in which the DUS is restricted to discrete distributions supported on the original data points. Zhao and Guan [44] reformulate the inner maximization problem in the context of two-stage stochastic programming and establish convergence of optimal solutions from the DRO problem to those of the SP for the underlying distribution as the number of data points grows large. And, in Zhao's dissertation [43], the reformulated problem is solved via Benders' decomposition, where the main computational effort lies in solving a separation problem in which bilinear terms appear [43]. Mohajerin Esfahani and Kuhn [28] propose reformulations that can handle, for example, piecewise affine functions, certain system reliability problems, and important types of two-stage stochastic linear programs. Gao and Kleywegt [17] propose a reformulation similar to the one in [44], but it allows for a more general support set for the random vector, a general nominal distribution, and higher orders for the Wasserstein distance. Blanchet and Murthy [8] propose a reformulation leveraging duality as well in which the DUS is based on optimal transport costs, which include Wasserstein-based sets as a special case. Hanasusanto and Kuhn [20] propose copositve formulations to bound the optimal value of a two-stage distributionally robust linear program using the Wasserstein metric. They establish conditions under which the gap between upper and lower bounds is zero, and provide a linear reformulation of a two-stage problem for the special case in which the randomness is only on the right-hand side of the second-stage constraints and the Wasserstein metric is defined with the one norm. Blanchet, Murthy, and Zhang [9] study a class of strongly convex optimal transport problems, which induce the same property in the resulting DRO problem; this facilitates attractive structural and algorithmic properties.

In this paper we extend the ideas of DRO to a class of multistage stochastic linear programs (MSPs) using Wasserstein-based uncertainty sets. For the case in which distributional robustness is not considered in the multistage problem, a standard solution approach originates with Pereira and Pinto [29], provided the problem has random elements that are interstage independent or satisfy appropriate notions of dependence. In their stochastic dual dynamic programming (SDDP) algorithm, the expected cost of future stages, i.e., the cost-to-go function, can be approximated by a piecewise linear convex function. The SDDP algorithm iteratively refines this approximation via two steps that alternate: first, a scenario—a sample path to a leaf node in the scenario tree—is sampled and decisions are taken sequentially, in a nonanticipative manner, under that sample path and under the current piecewise linear approximation of the expected cost at each stage (forward pass); and second, a new cut is generated at each stage by solving the immediate descendant problems and using the corresponding dual variables (backward pass).

In a *distributionally robust multistage SP* (DR-MSP), there is a nested min-max structure given that the underlying model assumes distributional uncertainty at each stage; see, e.g., [39]. Philpott, de Matos, and Kapelevich [32] propose an SDDP variant to solve a DR-MSP in which the uncertainty set is based on the modified χ^2 distance centered at the nominal distribution. To this end, the worst-case expectation is replaced by a piecewise linear approximation that is again refined by forward and backward passes. In order to compute a new cut at a particular stage, Philpott, de Matos, and Kapelevich [32] solve the inner maximization problem to obtain a proxy for the worst-case probability distribution, which is incorporated in the cut gradient and intercept. Although such a distribution might not be truly worst-case at intermediate iterations, it produces a valid lower-bounding cut for the worst-case expectation. As in [32], Huang, Zhou, and Guan [21] deal with finite prespecified support. They use uncertainty sets defined by an infinity norm and apply SDDP by reformulating the decomposition algorithm's subproblems at each stage, replacing the inner maximization with a convex combination of expected value and conditional value at risk of the cost-to-go function. Other decomposition algorithms for DRO under the Wasserstein metric have been proposed for two-stage models [2] and logistic regression [27]. Luo and Mehrotra [27] reformulate the distributionally robust problem as a semi-infinite program and solve it with a central cutting-surface algorithm, with application to distributionally robust logistic regression. Bansal, Huang, and Mehrotra [2] propose a decomposition algorithm for a broader class of problems including stochastic mixed binary problems. The algorithm is based on the L-shaped method for stochastic integer programming, but incorporates parametric cuts to deal with binary variables in both stages.

Our proposed approach to solving DR-MSP involves taking the dual of the inner maximization problem at each stage. This strategy is pervasive in the literature on robust optimization, and it has been applied in the DRO setting extensively [4, 6, 8, 9, 14, 18, 20, 28, 36, 37, 38, 42, 44]. This literature applies the single-level reformulation to static or two-stage SPs—as opposed to a multistage program—and the work establishes equivalence of the original min-max model using strong duality or results from infinite-dimensional convex programming. In this paper, we compare and contrast several algorithm variants, including the ones we derive and that of Philpott, de Matos, and Kapelevich [32]. We further investigate the merit of using a DRO-based model versus the corresponding nominal stochastic programming model in terms of out-of-sample performance, as also studied, for example, by Anderson and Philpott [1]. In other words, our primary motivation for pursuing DR-MSP is not that we believe nature to be adversarial in its selection of scenarios, but rather that we wish to prevent "overtraining" of the policy derived by the SDDP algorithm to the model's specific realizations. A parameter-robust, rather than distributionally robust, MSP is considered in [18], and our algorithm is a variant of theirs when the Wasserstein radius is sufficiently large and we restrict attention to a discrete support.

Employing empirical likelihood results, Lam and Zhou [25] construct confidence intervals using DRO. Further work provides systematic ways of constructing such uncertainty sets using Burg-entropy divergence [24] and more general divergence functions [16]. Duchi, Glynn, and Namkoong [16] also show how an asymptotic expansion of DRO involves the empirical mean as well as a variance term, which serves as a regularizer. Although our work does not focus on statistical properties inherited from uncertainty sets, it is worth noting that, even after a parametric form of a DUS is fixed, choosing its size, e.g., through a radius parameter, is an important aspect of DRO. In our computational experiments, we use cross validation to do so.

This paper is organized as follows. Section 2 states the problem and our assumptions. Section 3 reformulates DR-MSP and describes our variants of SDDP. Section 4 establishes convergence results. Section 5 describes extensions to section 2's model, which can be solved with the same algorithm. Section 6 presents numerical experiments that assess both the algorithm and the robustness of the resulting policies. Finally, section 7 concludes and outlines avenues for future research.

Notation: We let $[n] = \{1, 2, \ldots, n\}$, and $a = (a_i : i \in [n]) = (a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$.

2. Problem statement. A DRO problem may be defined as

$$\min_{x\in\mathcal{X}} \max_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[f(x,\xi)\right],$$

where $\mathcal{X} \subseteq \mathbb{R}^{d_x}$, $\mathcal{P}$ is a DUS defined over a sample space Ω with support Ξ, and $f : \mathcal{X} \times \Xi \to \mathbb{R}$. The uncertainty set's definition can incorporate observations of the random vector, ξ. In a data-driven application, we are given n data points $\Xi^n = \{\xi^i : i \in [n]\}$ with nominal distribution $\mathbb{Q}^n \equiv (q^1,\dots,q^n)$, often $q^i = 1/n$ for all $i \in [n]$. Set $\mathcal{P}$ contains a class of distributions that are within a radius of $\mathbb{Q}^n$ for some specified distance metric between two probability distributions.

This approach to uncertainty can be extended to an MSP [21, 32, 39]. In our DR-MSP, there is an uncertainty set, $\mathcal{P}_t$, for each stage, which we assume to be independent of the uncertainty sets and decisions of previous stages. Under this assumption, the model of interest is

$$\begin{aligned}\min_{x_1} \quad & c_1x_1 + \max_{\mathbb{P}_2\in\mathcal{P}_2} \mathbb{E}_{\mathbb{P}_2}\left[Q_2(x_1,\xi_2)\right]\\ \text{s.t.} \quad & A_1x_1 = B_1x_0 + \xi_1,\\ & x_1 \geq 0,\end{aligned} \tag{2.1}$$

where x_0 is given as input, and where the cost-to-go function for $t = 2,\dots,T$ is

$$\begin{aligned}Q_t(x_{t-1},\xi_t) = \quad \min_{x_t} \quad & c_tx_t + \max_{\mathbb{P}_{t+1}\in\mathcal{P}_{t+1}} \mathbb{E}_{\mathbb{P}_{t+1}}\left[Q_{t+1}(x_t,\xi_{t+1})\right]\\ \text{s.t.} \quad & A_tx_t = B_tx_{t-1} + \xi_t,\\ & x_t \geq 0.\end{aligned} \tag{2.2}$$

For simplicity we assume $Q_{T+1} \equiv 0$, although it can be a different piecewise linear convex terminal function. We assume that there are n_t realizations of the random vector ξ_t, ξ_t^i, $i \in [n_t]$, so that the support is $\Xi_t^{n_t} = \{\xi_t^1,\dots,\xi_t^{n_t}\}$. Thus, we consider randomness only in the right-hand side of the constraints in (2.2). For notational convenience, ξ_1 is sometimes referenced as a degenerate random vector, i.e., $n_1 = 1$. The set of scenarios is indexed by $\Omega = \bigtimes_{t=1}^{T}[n_t]$.

We let $\mathcal{X}_t(x_{t-1},\xi_t)$ denote the feasible regions of models (2.1)–(2.2) for $t = 1,2,\dots,T$. Adapting the notation in [28], let $\Xi_t \subseteq \mathbb{R}^{m_t}$ be the support of the random vector ξ_t and $\mathcal{M}(\Xi_t)$ be a class of probability distributions supported on Ξ_t. Furthermore, let $d : \mathcal{M}(\Xi_t) \times \mathcal{M}(\Xi_t) \to \mathbb{R}$ be a function that measures the distance between two probability distributions in $\mathcal{M}(\Xi_t)$. Then, for a given radius, $r \geq 0$, and nominal distribution, $\mathbb{Q}_t^n$, we can define a DUS as

$$\mathcal{P}_t = \{\mathbb{P}_t \in \mathcal{M}(\Xi_t) : d(\mathbb{Q}_t^n, \mathbb{P}_t) \leq r\}.$$

Under the above definition, different uncertainty sets can be defined depending on the choice of d, and the class of distributions that are admitted. In what follows we focus on the Wasserstein metric as the distance measure. And, in the bulk of what we present we restrict attention to probability distributions with finite and prespecified support in specifying $\mathcal{M}(\Xi_t^{n_t})$, although we describe extensions in section 5.

Throughout the paper we make the following assumptions:

(A.1) The DUS at stage t is given by the family of distributions on prespecified finite support, $\Xi_t^{n_t} = \{\xi_t^1,\dots,\xi_t^{n_t}\}$, with a Wasserstein distance of at most $r \geq 0$, i.e., $\mathcal{P}_t = \{\mathbb{P}_t \in \mathcal{M}(\Xi_t^{n_t}) : d_W(\mathbb{Q}_t^{n_t}, \mathbb{P}_t) \leq r\}$, where $\mathbb{Q}_t^{n_t} = (q_t^j > 0 : j \in [n_t])$ is the prespecified probability mass function on $\Xi_t^{n_t}$, and d_W is the Wasserstein metric (section 3.1).

(A.2) The polyhedral feasible region $\mathcal{X}_t(x_{t-1}, \xi_t)$ in models (2.1) and (2.2) is nonempty and compact for any feasible x_{t-1} and any $\xi_t \in \Xi_t^{n_t}$, $t = 1, \dots, T$.

Assumption (A.1) requires that the nominal distribution exhibit interstage independence as is typical in MSPs amenable to solution by SDDP, although extensions to, e.g., additive dependence models are possible [13, 22]. After we have more context on the algorithm we propose, we discuss implications of assumption (A.1) in section 5.4. Assumption (A.2) includes the requirement of relatively complete recourse. This, coupled with the fact that $\mathbb{P}_t$ and $\mathbb{Q}_t^{n_t}$ are defined on the same support $\Xi_t^{n_t}$ in (A.1) ensures feasibility of subproblems we encounter during the decomposition algorithms we describe.

3. Distributionally robust SDDP. We first formalize the definition of our DUSs under the Wasserstein metric. Next, we dualize the inner problem in order to obtain a single-level formulation at each stage and then show how to tackle the reformulation via SDDP. This section closes with discussions of sampling schemes and connections to the work of [32].

3.1. Wasserstein-based uncertainty set. For each stage, $t = 2, \dots, T$, the set $\mathcal{M}(\Xi_t^{n_t})$, denotes all probability distributions with support $\Xi_t^{n_t} = \{\xi_t^1, \dots, \xi_t^{n_t}\}$, and $\mathcal{P}_t$ restricts those distributions to be close to the nominal distribution, $\mathbb{Q}_t^{n_t}$, in a sense that we now make precise. For any probability mass function of the form $\mathbb{P}_t = (p_t^i : i \in [n_t])$, nominal distribution $\mathbb{Q}_t^{n_t} = (q_t^i : i \in [n_t])$, and distance $d_t^{i,j} = \|\xi_t^i - \xi_t^j\|_\eta$ (where, e.g., $\eta \in \{1, 2, \infty\}$), we can define the Wasserstein metric as

$$
\begin{aligned}
d_W(\mathbb{Q}_t^{n_t}, \mathbb{P}_t) = \min_z \; & \sum_{i \in [n_t]} \sum_{j \in [n_t]} d_t^{i,j} z^{i,j} \\
\text{s.t.} \; & \sum_{j \in [n_t]} z^{i,j} = q_t^i \quad \forall i \in [n_t], \\
& \sum_{i \in [n_t]} z^{i,j} = p_t^j \quad \forall j \in [n_t], \\
& z^{i,j} \geq 0 \quad \forall i, j \in [n_t].
\end{aligned}
$$

The corresponding uncertainty set is given by

$$
\mathcal{P}_t = \{\mathbb{P}_t \in \mathcal{M}(\Xi_t^{n_t}) : d_W(\mathbb{Q}_t^{n_t}, \mathbb{P}_t) \leq r\}. \tag{3.2}
$$

3.2. Single-level reformulation. For a given value of x_t, the inner maximization problem in (2.1) or (2.2) at stage t can be expressed as

$$
\max_{z_t, p_{t+1}} \sum_{j \in [n_{t+1}]} p_{t+1}^j Q_{t+1}(x_t, \xi_{t+1}^j) \tag{3.3a}
$$

$$
\text{s.t.} \quad \sum_{i \in [n_{t+1}]} \sum_{j \in [n_{t+1}]} d_{t+1}^{i,j} z_t^{i,j} \leq r, \tag{3.3b}
$$

$$
\sum_{j \in [n_{t+1}]} z_t^{i,j} = q_{t+1}^i \quad \forall i \in [n_{t+1}], \tag{3.3c}
$$

$$
\sum_{i \in [n_{t+1}]} z_t^{i,j} - p_{t+1}^j = 0 \quad \forall j \in [n_{t+1}], \tag{3.3d}
$$

$$
z_t^{i,j} \geq 0 \quad \forall i, j \in [n_{t+1}]. \tag{3.3e}
$$

Letting $\gamma_t \in \mathbb{R}$ and $\nu_t \in \mathbb{R}^{n_{t+1}}$ be dual variables for (3.3b) and (3.3c), respectively, the dual of (3.3) can be written

$$\min_{\gamma_t,\nu_t} \quad r\gamma_t + \sum_{i\in[n_{t+1}]} q^i_{t+1}\nu^i_t \tag{3.4a}$$

$$\text{s.t.} \quad d^{i,j}_{t+1}\gamma_t + \nu^i_t \geq Q_{t+1}(x_t,\xi^j_{t+1}) \quad \forall i,j \in [n_{t+1}], \tag{3.4b}$$

$$\gamma_t \geq 0. \tag{3.4c}$$

Note that in model (3.4) we substitute out the dual variables associated with (3.3d) via the constraints associated with p_{t+1}, and combine the result in (3.4b). Using strong duality for problems (3.3) and (3.4), we can reformulate (2.1) and (2.2) into single-level optimization problems. The problem for stages $t = 1,\ldots,T-1$ is therefore

$$\begin{aligned} Q_t(x_{t-1},\xi_t) = \quad & \min_{x_t,\gamma_t,\nu_t} && c_t x_t + r\gamma_t + \sum_{i\in[n_{t+1}]} q^i_{t+1}\nu^i_t \\ & \quad \text{s.t.} && A_t x_t = B_t x_{t-1} + \xi_t, \\ & && d^{i,j}_{t+1}\gamma_t + \nu^i_t \geq Q_{t+1}(x_t,\xi^j_{t+1}) \ \forall i,j \in [n_{t+1}], \\ & && x_t,\gamma_t \geq 0. \end{aligned} \tag{3.5}$$

To reformulate model (2.2) as a linear program, we first need to show that $Q_{t+1}(x_t,\xi_{t+1})$ is a piecewise linear convex function in x_t for fixed $\xi_{t+1} \in \Xi^{n_{t+1}}_{t+1}$.

LEMMA 3.1. *Assume* (A.1)–(A.2) *and let* $t \in \{1,2,\ldots,T-1\}$. *The function* $Q_{t+1}(\cdot,\xi_{t+1})$ *is a piecewise linear convex function on* $\mathcal{X}_t(x_{t-1},\xi_t)$ *with a finite number of pieces for any* $\xi_t \in \Xi^{n_t}_t$, $\xi_{t+1} \in \Xi^{n_{t+1}}_{t+1}$, *and feasible* x_{t-1}.

Proof. We proceed by induction backwards in the stages. For $t = T$,

$$\begin{aligned} Q_T(x_{T-1},\xi_T) &= \min_{x_T}\{c_T x_T : A_T x_T = B_T x_{T-1} + \xi_T; x_T \geq 0\} \\ &= \max_{\pi_T}\{\pi_T(B_T x_{T-1} + \xi_T) : \pi_T A_T \leq c_T\} \\ &= \max_{\kappa\in[\mathfrak{d}_T]}\{\pi_{T,\kappa}(B_T x_{T-1} + \xi_T)\}, \end{aligned}$$

where the second equality comes from strong duality and relatively complete recourse implied by assumption (A.2). Here, $\pi_{T,\kappa}$, $\kappa \in [\mathfrak{d}_T]$, denotes the extreme points of the dual problem for stage T, where $\mathfrak{d}_T$ is finite. Hence, the desired result holds for $t = T$. By induction, suppose that $Q_{t+1}(\cdot,\xi_{t+1})$ is piecewise linear and convex, with a finite number of pieces, for fixed ξ_{t+1}. For a particular realization indexed by $j \in [n_{t+1}]$, let $\mathcal{K}^j_{t+1}$ index all the pieces that describe $Q_{t+1}(x_t,\xi^j_{t+1})$ as a function of x_t with gradients $G^j_{t,\kappa}$ and intercepts $g^j_{t,\kappa}$ for all $\kappa \in \mathcal{K}^j_{t+1}$. Rewriting model (3.5),

$$Q_t(x_{t-1},\xi_t) = \min_{x_t,\gamma_t,\nu_t} c_t x_t + r\gamma_t + \sum_{i\in[n_{t+1}]} q^i_{t+1}\nu^i_t \tag{3.6a}$$

$$\text{s.t.} \quad A_t x_t = B_t x_{t-1} + \xi_t \ [\pi_t], \tag{3.6b}$$

$$d^{i,j}_{t+1}\gamma_t + \nu^i_t \geq G^j_{t,\kappa}x_t + g^j_{t,\kappa} \quad \forall i,j \in [n_{t+1}], \kappa \in \mathcal{K}^j_{t+1} \ [z^{i,j}_{t,\kappa}], \tag{3.6c}$$

$$x_t,\gamma_t \geq 0. \tag{3.6d}$$

Model (3.6a) has a finite optimal solution by (A.1) and (A.2). Taking its dual yields

$$Q_t(x_{t-1},\xi_t) = \max_{\pi_t, z_t} \pi_t\left(B_t x_{t-1} + \xi_t\right) + \sum_{i\in[n_{t+1}]}\sum_{j\in[n_{t+1}]}\sum_{\kappa\in\mathcal{K}^j_{t+1}} g^j_{t,\kappa} z^{i,j}_{t,\kappa} \tag{3.7a}$$

$$\text{s.t. } \pi_t A_t - \sum_{i\in[n_{t+1}]}\sum_{j\in[n_{t+1}]}\sum_{\kappa\in\mathcal{K}^j_{t+1}} G^j_{t,\kappa} z^{i,j}_{t,\kappa} \le c_t \quad [x_t], \tag{3.7b}$$

$$\sum_{i\in[n_{t+1}]}\sum_{j\in[n_{t+1}]}\sum_{\kappa\in\mathcal{K}^j_{t+1}} d^{i,j}_{t+1} z^{i,j}_{t,\kappa} \le r \quad [\gamma_t], \tag{3.7c}$$

$$\sum_{j\in[n_{t+1}]}\sum_{\kappa\in\mathcal{K}^j_{t+1}} z^{i,j}_{t,\kappa} = q^i_{t+1} \quad \forall i \in [n_{t+1}] \quad [\nu_t], \tag{3.7d}$$

$$z_t \ge 0. \tag{3.7e}$$

Writing model (3.7) as the maximum over the finite number, $\mathfrak{d}_t$, of extreme points of (3.7b)–(3.7e)'s polyhedron yields the desired result:

$$Q_t(x_{t-1},\xi_t) = \max_{k\in[\mathfrak{d}_t]} \pi_{t,k}\left(B_t x_{t-1} + \xi_t\right) + \sum_{i\in[n_{t+1}]}\sum_{j\in[n_{t+1}]}\sum_{\kappa\in\mathcal{K}^j_{t+1}} g^j_{t,\kappa}\left(z^{i,j}_{t,\kappa}\right)_k,$$

i.e., $Q_t(x_{t-1},\xi_t)$ is a piecewise linear convex function of x_{t-1} with a finite number of pieces. ∎

Lemma 3.1's proof gives us the equations for gradients and intercepts of the pieces that define $Q_t(x_{t-1},\xi_t)$. For a fixed realization $\xi_t = \xi^j_t$ and fixed $x_{t-1} = \hat{x}_{t-1}$, let $\pi^j_{t,k}$ be a subvector in a maximizer of (3.7) for such input. Then, the piece indexed by $k \in \mathcal{K}^j_t$ supporting $Q_t(x_{t-1},\xi^j_t)$ at $\hat{x}_{t-1}$ is determined by the cut gradient $G^j_{t-1,k}$ and cut intercept $g^j_{t-1,k}$ with formulas

$$G^j_{t-1,k} = \pi^j_{t,k} B_t, \tag{3.8a}$$

$$g^j_{t-1,k} = Q_t(\hat{x}_{t-1},\xi^j_t) - \pi^j_{t,k} B_t \hat{x}_{t-1}. \tag{3.8b}$$

Equations (3.8) suggest an algorithm in which piecewise linear approximations of $Q_t(x_{t-1},\xi^j_t)$ are iteratively refined. In what follows we present such an algorithm to solve model (2.1) and show its convergence properties.

3.3. SDDP algorithm for DR-MSP. To solve model (2.1) we use a multicut version of SDDP. Instead of solving model (3.6a) directly, we approximate the cost-to-go function $Q_t(\cdot,\xi^j_t)$ for each $j \in [n_t]$. Above we use $\mathcal{K}^j_t$ to index *all* of the finitely many pieces that characterize $Q_t(\cdot,\xi^j_t)$. Here, we let $\mathcal{K}^{j,k}_t$ index the cuts that have been generated to approximate the function $Q_t(\cdot,\xi^j_t)$ after k iterations of the algorithm. We denote the optimal value of these (relaxed, as we subsequently prove) subproblems at the kth iteration by $\mathfrak{Q}^k_t(x_{t-1},\xi_t)$, where

$$\mathfrak{Q}^k_t\left(x_{t-1},\xi_t\right) = \min_{x_t,\gamma_t,\nu_t} c_t x_t + r\gamma_t + \sum_{i\in[n_{t+1}]} q^i_{t+1}\nu^i_t \tag{3.9a}$$

$$\text{s.t. } A_t x_t = B_t x_{t-1} + \xi_t \quad [\pi_t], \tag{3.9b}$$

$$d^{i,j}_{t+1}\gamma_t + \nu^i_t \ge G^j_{t,\kappa} x_t + g^j_{t,\kappa} \quad \forall i,j \in [n_{t+1}], \kappa \in \mathcal{K}^{j,k}_{t+1} \quad [z^{i,j}_{t,\kappa}], \tag{3.9c}$$

$$x_t, \gamma_t \ge 0. \tag{3.9d}$$

At iteration k, the algorithm samples a forward path and solves the subproblems (3.9) to obtain $\hat{x}_t$ for all $t = 1, \dots, T-1$. In the backward pass, a new cut is computed based on dual variables from $t = T, \dots, 2$, for each realization indexed by $j \in [n_t]$, refining an outer approximation of $Q_t(\cdot, \xi_t^j)$ for each $j \in [n_t]$. Cut gradients and intercepts for each $j \in [n_t]$ are added to the stage $t-1$ version of subproblem (3.9) using

$$G_{t-1,k}^j = \pi_{t,k}^j B_t, \tag{3.10a}$$

$$g_{t-1,k}^j = \mathfrak{Q}_t^k(\hat{x}_{t-1}, \xi_t^j) - \pi_{t,k}^j B_t \hat{x}_{t-1}. \tag{3.10b}$$

Algorithm 3.1 shows the procedure to solve model (2.1) using subproblems of the form (3.9). We denote subproblem (3.9) after k iterations by $P_t^k(x_{t-1}, \xi_t)$.

Algorithm 3.1 SDDP variant for DR-MSP.

Require: $P_t^0,\ t = 1, \dots, T$, subproblems with initial lower-bounding cuts; K, maximum number of iterations; Ω, set of scenarios; $\hat{x}_0 = x_0$ for $P_1^0(\hat{x}_0, \xi_1)$.
Ensure: $P_t^K,\ t = 1, \dots, T$, subproblems with cuts accumulated through K iterations.
1: `termination=False`, $k = 0$
2: **while** `termination` $=$ `False` **do**
3:     **for** $t = 1, \dots, T-1$ **do** ▷ Forward pass
4:         Sample $\hat{\xi}_t$ from sample space $[n_t]$ with probability $\mathbb{Q}_t^{n_t}$
5:         Solve $P_t^k(\hat{x}_{t-1}, \hat{\xi}_t)$ and obtain $\hat{x}_t$
6:     **for** $t = T, \dots, 2$ **do**
7:         **for** $j \in [n_t]$ **do** ▷ Backward pass
8:             Solve $P_t^k(\hat{x}_{t-1}, \xi_t^j)$ and obtain $\pi_{t,k}^j$ and $\mathfrak{Q}_t(\hat{x}_{t-1}, \xi_t^j)$
9:         Add cuts to stage $t-1$ collection using (3.10)
10:     `termination=check_termination(`K`)`
11:     $k = k + 1$
12: **return** subproblems $P_t^K \quad \forall t = 1, \dots, T-1$

As part of the input, we require initial lower-bounding cuts to preclude subproblem (3.9) from being unbounded. One way to produce such cuts is to run several iterations of a standard version of SDDP on the expected-value form of model (2.1) under the nominal distribution, i.e., with $r = 0$. In its forward pass, Algorithm 3.1 samples from the nominal distribution, including the degenerate stage 1 realization for which $[n_1]$ is a singleton. We have some latitude to modify Algorithm 3.1 regarding how to carry out the sampling in step 4, and we discuss this further in the next section. As part of the backward pass, in implementation we add *up to* n_t cuts to the set accumulated in stage $t-1$ at each iteration because we only add cuts that are violated at the current stage $t-1$ solution. As in other SDDP algorithmic settings, there are challenges in computing upper bounds when certain types of risk measures or distributionally robust formulations are employed, although we point to [23, 26, 31]. For this reason, in step 10 we simply terminate the algorithm after a prespecified number of iterations, K, or after a given computational budget is consumed. The output of Algorithm 3.1 is a set of cuts at each stage $t = 1, \dots, T-1$. These cuts yield a policy that can be employed on out-of-sample realizations that the algorithm did not see in constructing these cuts. For reasons sketched in the paper's introduction—regarding our motivation for considering models with distributional robustness—we return to this in our computational results in section 6. As we formalize below, like other SDDP-style algorithms, the cuts also yield deterministically valid outer linearizations of each $Q_t(\cdot, \xi_t^j)$ and, hence, solving model (3.9) for $t = 1$ yields a deterministically valid lower bound on the optimal value of model (2.1).

3.4. Dynamic sampling algorithm. Variants of Algorithm 3.1 permit different sampling procedures. Step 4 of the algorithm samples scenarios using the nominal distributions, $\mathbb{Q}_t^{n_t}$, $t = 2, \dots, T$. As an alternative, we could instead sample according to the worst-case distribution at each stage, with the notion that this might accelerate growth of the algorithm's lower bound as relevant scenarios would be sampled more frequently. Since the worst-case distribution is unknown, we could use as a proxy the current worst-case distribution in order to sample scenarios dynamically. The dual variables z_t of subproblem (3.9) are related to the transportation variables in model (3.3), and satisfy an analog of constraint (3.3d) at iteration k:

$$p_{t+1}^{j,k} = \sum_{i \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} z_{t,\kappa}^{i,j} \quad \forall j \in [n_{t+1}]. \tag{3.11}$$

Using (3.11) and $\mathbb{Q}_t^{n_t}$ we construct the following distribution to sample scenarios for a given parameter $\beta \in [0, 1)$:

$$\hat{\mathbb{P}}_t^k \equiv \left(\hat{p}_t^{j,k} = \beta p_t^{j,k} + (1 - \beta) q_t^j : j \in [n_t] \right). \tag{3.12}$$

If $\beta = 0$ then we sample the nominal distribution, and if $\beta = 1$ we sample the current estimate of the worst-case distribution. Other values of β allow a balance between these extremes. As we discuss in section 4, in order to ensure convergence we require $\beta < 1$ so that assumption (A.1) ensures $\hat{p}_t^{j,k}$ is (uniformly) bounded away from zero for all $j \in [n_t]$. The only modification in the algorithm is in the sampling procedure, which updates the sampling distribution during the forward pass. Algorithm 3.2 summarizes the changes to steps 3–5 in Algorithm 3.1's forward pass.

Algorithm 3.2 SDDP forward pass under dynamic sampling.

Require: $\beta \in [0, 1)$, parameter for sampling distribution; $\hat{\mathbb{P}}_1^k = (1)$, degenerate probability mass function for stage 1

1: **for** $t = 1, \dots, T - 1$ **do**
2: Sample $\hat{\xi}_t$ from sample space $[n_t]$ with distribution $\hat{\mathbb{P}}_t^k$
3: Solve $P_t^k(\hat{x}_{t-1}, \hat{\xi}_t)$ and obtain $\hat{x}_t$ and dual variables $\hat{z}_t$
4: Compute worst-case distribution proxy $\hat{\mathbb{P}}_{t+1}^k$ with $\hat{z}_t$ and β using (3.11)–(3.12)

We assume the following regarding solutions to models $P_t^k(\hat{x}_{t-1}, \cdot)$ obtained in Algorithms 3.1 and 3.2.

(A.3) When solving subproblems (3.9) in Algorithm 3.1 and 3.2, we obtain extreme point solutions and multiple optimal solutions in the primal (step 5) and the dual (step 8) are, respectively, resolved using a consistent tie-breaking rule.

In addition to obtaining extreme point solutions, assumption (A.3) ensures that if we solve an identical subproblem multiple times, we obtain the same solution.

3.5. Single-cut and multicut algorithms. Both multicut and single-cut versions of the L-shaped method and, hence, SDDP, are possible [7]. A second choice in designing an SDDP algorithm for DR-MSP is whether the inner "$\max_{\mathbb{P}_{t+1} \in \mathcal{P}_{t+1}}$"

is handled via duality or by using a side computation. Together these choices suggest four possible SDDP-style algorithms for DR-MSP. In the algorithm we develop above, we use a multicut version and employ duality to handle the inner maximization. Philpott, de Matos, and Kapelevich [32] instead use a single-cut algorithm and rather than using duality to handle the DRO aspect of the problem, for a given $\hat{x}_{t-1}$ in the backward pass, the inner maximization is computed in an auxiliary computation, and the worst-case distribution is computed for that specific $\hat{x}_{t-1}$ at that specific iteration, k. (In contrast, via duality our worst-case distribution is implicitly recomputed at each iteration for all cuts.) Moreover, in [32] the quadratic nature of the modified χ^2 distance measure is exploited so that the side computation can be performed analytically, although their algorithm easily adapts instead to solve numerically an optimization problem, for example, of the form (3.3), with $Q_{t+1}(\hat{x}_t, \xi^j_{t+1})$ replaced by $\mathfrak{Q}^k_{t+1}(\hat{x}_t, \xi^j_{t+1})$ in the objective function.

Our attempt to derive a single-cut algorithm using duality led to a formulation identical to (3.9). The multicut variables, ν^i_t, and associated cut constraints (3.9c), seem to arise naturally given constraints (3.3c) defining the Wasserstein set. However, the possibility of using a multicut algorithm while computing cuts in a side computation is viable, as we now sketch.

We need to approximate both the function, $Q_{t+1}(x_t, \xi^j_{t+1})$ for each $j \in [n_{t+1}]$, and the corresponding worst-case expectation, $\max_{\mathbb{P}_{t+1} \in \mathcal{P}_{t+1}} \mathbb{E}_{\mathbb{P}_{t+1}} [Q_{t+1}(x_t, \xi_{t+1})]$, to account for the fact that the worst-case probability distribution changes from one iteration to the next. Let θ^j_t be a decision variable that approximates $Q_{t+1}(\cdot, \xi^j_{t+1})$, and further define $\theta^M_t = \max_{\kappa \in [k]} \{\sum_{j \in [n_{t+1}]} p^{j,\kappa}_{t+1} \theta^j_t\}$ as the approximation of the worst-case expectation after k iterations. Then, we can approximate (2.1) and (2.2) with the following problem:

$$
\begin{aligned}
&(3.13a) \qquad && \underline{\mathfrak{Q}}^k_t(x_{t-1}, \xi_t) = \min_{x_t, \theta_t, \theta^M_t} \ c_t x_t + \theta^M_t \\
&(3.13b) && \text{s.t.} \quad A_t x_t = B_t x_{t-1} + \xi_t, \\
&(3.13c) && \theta^M_t \geq \sum_{j \in [n_{t+1}]} p^{j,\kappa}_{t+1} \theta^j_t \quad \forall \kappa \in [k], \\
&(3.13d) && \theta^j_t \geq G^j_{t,\kappa} x_t + g^j_{t,\kappa} \quad \forall j \in [n_{t+1}], \kappa \in \mathcal{K}^{j,k}_{t+1}, \\
&(3.13e) && x_t \geq 0.
\end{aligned}
$$

In a typical multicut master problem, variables θ^j_t are weighted by their probabilities in the objective function. Problem (3.13), instead, has the additional constraints (3.13c) because in our setting, we might find a distinct distribution, $p^{j,\kappa}_{t+1}$, $j \in [n_{t+1}]$, in a side computation at each iteration κ, as done in [32] in the single-cut case, and these probabilities are henceforth fixed.

Algorithm 3.1 differs from [32] in two key ways: (i) we use a multicut rather than single-cut procedure, and (ii) we use duality rather than an iteration-by-iteration side computation for DRO. The third variant of DRO-SDDP algorithms just sketched allows us to isolate the effect of these two differences. The following proposition compares the lower bounds obtained by our two multicut variants of the algorithm.

PROPOSITION 3.2. *Assume* (A.1)–(A.2) *and let* $t \in \{1, 2, \ldots, T-1\}$. *Let* x_{t-1} *be feasible and* $\xi_t \in \Xi^{n_t}_t$ *be given. Assume problems* (3.9) *and* (3.13) *have the same set of cuts, i.e.,* $\mathcal{K}^{j,k}_{t+1}$ *indexes the same set of cuts in* (3.9c) *and* (3.13d)*, respectively. Then,* $\mathfrak{Q}^k_t(x_{t-1}, \xi_t) \geq \underline{\mathfrak{Q}}^k_t(x_{t-1}, \xi_t)$.

Proof. We proceed by forming the duals of (3.9) and (3.13). To facilitate our argument, we introduce auxiliary variables for all $j \in [n_{t+1}]$ and $\kappa \in \mathcal{K}_{t+1}^{j,k}$:

$$\alpha_t^{j,\kappa} = \sum_{i \in [n_{t+1}]} z_{t,\kappa}^{i,j}.$$

Hence, we can write the dual of (3.9):

$$\mathfrak{Q}_t^k(x_{t-1}, \xi_t) = \max_{\pi_t, z_t, \alpha_t} \pi_t \left(B_t x_{t-1} + \xi_t\right) + \sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} g_{t,\kappa}^j \alpha_t^{j,\kappa} \tag{3.14a}$$

$$\text{s.t.} \quad \pi_t A_t - \sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} G_{t,\kappa}^j \alpha_t^{j,\kappa} \le c_t, \tag{3.14b}$$

$$\sum_{i \in [n_{t+1}]} \sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} d_{t+1}^{i,j} z_{t,\kappa}^{i,j} \le r, \tag{3.14c}$$

$$\sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} z_{t,\kappa}^{i,j} = q_{t+1}^i \quad \forall i \in [n_{t+1}], \tag{3.14d}$$

$$\sum_{i \in [n_{t+1}]} z_{t,\kappa}^{i,j} - \alpha_t^{j,\kappa} = 0 \quad \forall j \in [n_{t+1}], \kappa \in \mathcal{K}_{t+1}^{j,k}, \tag{3.14e}$$

$$z_t \ge 0. \tag{3.14f}$$

To write the dual of (3.13), let π_t, λ_t^κ, and $\alpha_t^{j,\kappa}$ be the the dual variables for constraints (3.13b)–(3.13d), respectively. Hence, we can write the dual problem of (3.13):

$$\underline{\mathfrak{Q}}_t^k(x_{t-1}, \xi_t) = \max_{\pi_t, z_t, \alpha_t} \pi_t \left(B_t x_{t-1} + \xi_t\right) + \sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} g_{t,\kappa}^j \alpha_t^{j,\kappa} \tag{3.15a}$$

$$\text{s.t.} \quad \pi_t A_t - \sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} G_{t,\kappa}^j \alpha_t^{j,\kappa} \le c_t, \tag{3.15b}$$

$$\sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} \alpha_t^{j,\kappa} - \sum_{\kappa \in [k]} p_{t+1}^{j,\kappa} \lambda_t^\kappa = 0 \qquad \forall j \in [n_{t+1}], \tag{3.15c}$$

$$\sum_{\kappa \in [k]} \lambda_t^\kappa = 1, \tag{3.15d}$$

$$\alpha_t, \lambda_t \ge 0. \tag{3.15e}$$

The projection of the set defined by constraints (3.3b)–(3.3e) onto $p_{t+1} = (p_{t+1}^j : j \in [n_{t+1}])$ characterizes the uncertainty set, $\mathcal{P}_{t+1}$. Constraints (3.14c)–(3.14f) play the same role in characterizing $\mathcal{P}_{t+1}$, where $z_t^{i,j} = \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} z_{t,\kappa}^{i,j}$ and where p_t^j is defined by (3.11). The parameters, $p_{t+1}^{j,\kappa}$, $j \in [n_{t+1}], \kappa \in \mathcal{K}_{t+1}^{j,k}$, in subproblem (3.15) are computed at each iteration by optimizing an auxiliary problem over $\mathcal{P}_{t+1}$. After k iterations, in the stage-t subproblem we have $(p_{t+1})^\kappa, \kappa = 1, 2, \ldots, k$, and the constraints (3.15c)–(3.15e) characterize the convex hull of these k vectors, i.e., they describe only a subset of $\mathcal{P}_{t+1}$. The feasible region of (3.14) captures all of $\mathcal{P}_{t+1}$, and that of (3.15) only captures a subset and, hence, we have $\mathfrak{Q}_t^k(x_{t-1}, \xi_t) \ge \underline{\mathfrak{Q}}_t^k(x_{t-1}, \xi_t)$. ∎

4. Convergence of DR-SDDP algorithm. The correctness of Algorithm 3.1 depends on generating valid lower-bounding cuts and, hence, that $\mathfrak{Q}_t^k(x_{t-1}, \xi_t) \leq Q_t(x_{t-1}, \xi_t)$ for all $t = 2, \ldots, T$ and for all $k \geq 1$. We make these statements concrete in the following lemma.

LEMMA 4.1. *Assume* (A.1)–(A.3)*, let* $t \in \{1, 2, \ldots, T-1\}$*, let* x_{t-1} *be feasible, and let* $\xi_t, \xi_t^j \in \Xi_t^{n_t}$*. Consider problem* (3.9)*, and assume that the cuts in constraint* (3.9c) *are valid, i.e.,* $\max_{\kappa \in \mathcal{K}_{t+1}^{j,k}} [G_{t,\kappa}^j x_t + g_{t,\kappa}^j] \leq Q_{t+1}(x_t, \xi_{t+1}^j)$ *for all* $x_t \in \mathcal{X}_t(x_{t-1}, \xi_t)$ *and* $j \in [n_{t+1}]$*. Let* $(\hat{\pi}_t, \hat{z}_t)$ *be dual feasible to problem* (3.9) *with input* $(\hat{x}_{t-1}, \xi_t^j)$*. Then, the cuts derived from* (3.10) *are valid, i.e.,* $G_{t-1}^j x_{t-1} + g_{t-1}^j \leq Q_t(x_{t-1}, \xi_t^j)$ *for all* $j \in [n_t]$*. Furthermore,* $\mathfrak{Q}_t^k(x_{t-1}, \xi_t) \leq Q_t(x_{t-1}, \xi_t)$*.*

Proof. Let $\hat{\pi}_T$ be a dual feasible extreme point to problem (3.9) for $t = T$, and let $\mathcal{K}_T^{j,k}$ denote the corresponding index set. (Note that at stage T there are no cuts and, therefore, dual variables only involve π_T.) For any $j \in [n_T]$ and any feasible x_{T-1}, we have

$$\begin{aligned}
G_{T-1}^j x_{T-1} + g_{T-1}^j &= \hat{\pi}_T B_T x_{T-1} + \hat{\pi}_T \xi_T^j \\
&\leq \max_{\pi_T} \{\pi_T (B_T x_{T-1} + \xi_T^j) : \pi_T A_T \leq c_T\} \\
&= \min_{x_T} \{c_T x_T : A_T x_T = B_T x_{T-1} + \xi_T^j; x_T \geq 0\} \\
&= Q_T(x_{T-1}, \xi_T^j),
\end{aligned}$$

where the first equality holds by (3.10), and the penultimate equality holds by assumption (A.2). Thus, the cut is valid for stage $T-1$.

Proceeding by induction, now consider problem (3.9) for $t \leq T-1$ with input $(\hat{x}_{t-1}, \xi_t^j)$; assume that the cuts on the right-hand side of inequality (3.9c) are valid and let $(\hat{\pi}_t, \hat{z}_t)$ be a dual feasible solution. Then, by (3.10)

$$G_{t-1}^j = \hat{\pi}_t B_t \text{ and} \tag{4.1a}$$

$$\begin{aligned}
g_{t-1}^j &= \mathfrak{Q}_t^k(\hat{x}_{t-1}, \xi_t^j) - \hat{\pi}_t B_t \hat{x}_{t-1} \\
&= \hat{\pi}_t \xi_t^j + \sum_{i \in [n_{t+1}]} \sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} g_{t,\kappa}^j \hat{z}_{t,\kappa}^{i,j}.
\end{aligned} \tag{4.1b}$$

If problem (3.9) had cuts in (3.9c) indexed by $\mathcal{K}_{t+1}^j$, $j \in [n_{t+1}]$, rather than $\mathcal{K}_{t+1}^{j,k}$, $j \in [n_{t+1}]$, then by duality its optimal value would be identical to that of problem (3.7). Let $\Pi_t = \{(\pi_t, z_t) : (\pi_t, z_t)$ satisfies (3.7b)–(3.7e)$\}$, and let Π_t^k denote the analogous set when the cuts indexed by $\mathcal{K}_{t+1}^j$, $j \in [n_{t+1}]$, are replaced by those of $\mathcal{K}_{t+1}^{j,k}$, $j \in [n_{t+1}]$. We then have that

$$G_{t-1}^j x_{t-1} + g_{t-1}^j = \hat{\pi}_t (B_t x_{t-1} + \xi_t^j) + \sum_{i \in [n_{t+1}]} \sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_t^{j,k}} g_{t,\kappa}^j \hat{z}_{t,\kappa}^{i,j} \tag{4.2}$$

$$\leq \max_{(\pi_t, z_t) \in \Pi_t^k} \pi_t \left(B_t x_{t-1} + \xi_t^j \right) + \sum_{i \in [n_{t+1}]} \sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j,k}} g_{t,\kappa}^j z_{t,\kappa}^{i,j} \tag{4.3}$$

$$\leq \max_{(\pi_t, z_t) \in \Pi_t} \pi_t \left(B_t x_{t-1} + \xi_t^j \right) + \sum_{i \in [n_{t+1}]} \sum_{j \in [n_{t+1}]} \sum_{\kappa \in \mathcal{K}_{t+1}^{j}} g_{t,\kappa}^j z_{t,\kappa}^{i,j} \tag{4.4}$$

$$= Q_t(x_{t-1}, \xi_t^j). \tag{4.5}$$

Equation (4.2) simply applies (4.1), which defines the cut gradient and intercept, and inequality (4.3) holds because $(\hat{\pi}_t, \hat{z}_t) \in \Pi_t^k$. Inequality (4.4) holds because $\mathcal{K}_{t+1}^j$ indexes cuts which define $Q_{t+1}(x_t, \xi_t^j)$ while by hypothesis $\mathcal{K}_{t+1}^{j,k}$ indexes valid, i.e., lower bounding, cuts. Equation (4.5) follows from strong duality of problems (3.6a) and (3.7). This establishes the first claim that all cuts are valid, and hence $\mathfrak{Q}_t^k(x_{t-1}, \xi_t) \le Q_t(x_{t-1}, \xi_t)$ follows immediately. ∎

The next lemma states that when solving problem (3.9) at stage $t = T-1$, we either obtain an optimal solution to problem (3.6a), or the stage T descendants identify a new cut.

LEMMA 4.2. *Assume* (A.1)–(A.3). *Let* $(x_{T-1}^k, \gamma_{T-1}^k, \nu_{T-1}^k)$ *be an optimal solution to problem* (3.9) *for* $t = T-1$, *where* $\mathcal{K}_T^{j,k}$ *indexes the cuts on the right-hand side of constraint* (3.9c). *Then exactly one of the following holds:*

(i) $\exists\ \hat{\pi}_T \in \{\pi_T : \pi_T A_T \le c_T\}$ *and* $i, j \in [n_T]$ *such that* $d_T^{i,j}\gamma_{T-1}^k + \nu_{T-1}^{i,k} < \hat{\pi}_T(B_T x_{T-1}^k + \xi_T^j)$.

(ii) $(x_{T-1}^k, \gamma_{T-1}^k, \nu_{T-1}^k)$ *solves problem* (3.6a) *for* $t = T-1$.

Proof. The solution $(x_{T-1}^k, \gamma_{T-1}^k, \nu_{T-1}^k)$ satisfies constraints (3.9b) and (3.9d) and, hence, also satisfies the identical constraints (3.6b) and (3.6d). Examining constraint (3.6c), we are in one of the following two cases:

$$\text{(4.6a)}\quad d_T^{i,j}\gamma_{T-1}^k + \nu_{T-1}^{i,k} \ge \max_{\kappa \in \mathcal{K}_T^j} \left\{ G_{T-1,\kappa}^j x_{T-1}^k + g_{T-1,\kappa}^j \right\} \quad \forall i, j \in [n_T],$$

$$\text{(4.6b)}\quad \exists\ i, j \in [n_T] \text{ such that } d_T^{i,j}\gamma_{T-1}^k + \nu_{T-1}^{i,k} < \max_{\kappa \in \mathcal{K}_T^j} \left\{ G_{T-1,\kappa}^j x_{T-1}^k + g_{T-1,\kappa}^j \right\}.$$

Suppose case (4.6a) holds. By Lemma 4.1 for any feasible x_{T-1} and $j \in [n_T]$

$$\max_{\kappa \in \mathcal{K}_T^j} \left\{ G_{T-1,\kappa}^j x_{T-1} + g_{T-1,\kappa}^j \right\} \ge \max_{\kappa \in \mathcal{K}_T^{j,k}} \left\{ G_{T-1,\kappa}^j x_{T-1} + g_{T-1,\kappa}^j \right\}.$$

As a result, (ii) holds because problem (3.9) is a relaxation of problem (3.6a), and an optimal solution to the former is feasible for the latter. In addition, by (3.10) and (4.6a) for each $i, j \in [n_T]$,

$$\begin{aligned} d_T^{i,j}\gamma_{T-1}^k + \nu_{T-1}^{i,k} &\ge \max_{\kappa \in \mathcal{K}_T^j} \{\pi_{T,\kappa}(B_T x_{T-1}^k + \xi_T^j)\} \\ &\ge \hat{\pi}_T(B_T x_{T-1}^k + \xi_T^j) \quad \forall \hat{\pi}_T \in \{\pi_T : \pi_T A_T \le c_T\}, \end{aligned}$$

and so (i) does not hold.

Suppose case (4.6b) holds. Solution $(x_{T-1}^k, \gamma_{T-1}^k, \nu_{T-1}^k)$ violates at least one constraint of form (3.6c) and, hence, (ii) does not hold. Using (3.10) and (4.6b) there exists $i, j \in [n_T]$ and an extreme point $\hat{\pi}_T \in \{\pi_T : \pi_T A_T \le c_T\}$ with $G_{T-1,\kappa}^j = \hat{\pi}_T B_T$ and $g_{T-1,\kappa}^j = \hat{\pi}_T \xi_T^j$ such that (i) holds. ∎

The previous lemma establishes that, given its current input, either a stage $T-1$ approximating problem is exact or we generate a new cut. The following lemma establishes an analog for stages $t = 1, 2, \ldots, T-2$.

LEMMA 4.3. *Assume* (A.1)–(A.3) *and let* $t \in \{1, 2, \ldots, T-2\}$. *Let* $(x_t^k, \gamma_t^k, \nu_t^k)$ *be an optimal solution to problem* (3.9), *where* $\mathcal{K}_{t+1}^{j,k}$ *indexes the cuts on the right-hand side of constraint* (3.9c). *Assume*

$$\text{(4.7)}\qquad \max_{\kappa \in \mathcal{K}_{t+2}^{j',k}} \left\{ G_{t+1,\kappa}^{j'} x_{t+1}^{k,j} + g_{t+1,\kappa}^{j'} \right\} = Q_{t+2}(x_{t+1}^{k,j}, \xi_{t+2}^{j'})\ \forall j \in [n_{t+1}],\ j' \in [n_{t+2}],$$

where $x_{t+1}^{k,j}$ *is a subvector of an optimal solution to the descendant subproblems* (3.9) *for stage* $t+1$ *with input* (x_t^k, ξ_{t+1}^j)*. Then exactly one of the following holds:*

(i) $\exists(\hat{\pi}_{t+1}, \hat{z}_{t+1})$ *dual feasible to* (3.9) *and* $i, j \in [n_{t+1}]$ *such that* $d_{t+1}^{i,j}\gamma_t^k + \nu_t^{i,k} < \hat{G}_t x_t^k + \hat{g}_t$*, where* $\hat{G}_t = \hat{\pi}_{t+1}B_{t+1}$ *and* $\hat{g}_t = \hat{\pi}_{t+1}\xi_{t+1}^j + \sum_{i',j'\in[n_{t+2}]}\sum_{\kappa\in\mathcal{K}_{t+2}^{j',k}} g_{t+1,\kappa}^{j'}\,\hat{z}_{t+1,\kappa}^{i',j'}$.

(ii) $(x_t^k, \gamma_t^k, \nu_t^k)$ *solves problem* (3.6a).

Proof. The proof proceeds in a fashion parallel to that of Lemma 4.2: Problems (3.6a) and (3.9) are identical except for constraints (3.6c) and (3.9c). By Lemma 4.1 problem (3.9) is a relaxation of problem (3.6a). Solution $(x_t^k, \gamma_t^k, \nu_t^k)$ either satisfies constraint (3.6c), i.e., satisfies the inequality for all $i, j \in [n_{t+1}]$ and $\kappa \in \mathcal{K}_{t+1}^j$, or the constraint is violated for some $i, j \in [n_{t+1}]$ and $\kappa \in \mathcal{K}_{t+1}^j$. In the former case, it is immediate that (ii) holds and (i) does not. In the latter case, again analogous to the proof of Lemma 4.2, it is immediate that (ii) does not hold. It remains to show that (i) holds in the latter case. The descendant subproblems (3.9) can be written

$$
\begin{aligned}
&\mathfrak{Q}_{t+1}^k(x_t^k, \xi_{t+1}^j)\\
&\quad= \min_{x_{t+1},\gamma_{t+1},\nu_{t+1}} c_{t+1}x_{t+1} + r\gamma_{t+1} + \sum_{i'\in[n_{t+2}]} q_{t+2}^{i'}\nu_{t+1}^{i'}\\
&\qquad \text{s.t.}\quad A_{t+1}x_{t+1} = B_{t+1}x_t^k + \xi_{t+1}^j,\\
&\qquad\qquad d_{t+2}^{i',j'}\gamma_{t+1} + \nu_{t+1}^{i'} \geq G_{t+1,\kappa}^{j'}x_{t+1} + g_{t+1,\kappa}^{j'}\ \forall i', j' \in [n_{t+2}], \kappa \in \mathcal{K}_{t+2}^{j',k},\\
&\qquad\qquad x_{t+1}, \gamma_{t+1} \geq 0,
\end{aligned}
$$

and hypothesis (4.7) allows for equivalently indexing the cut constraints over $\mathcal{K}_{t+2}^{j',k}$ or over $\mathcal{K}_{t+2}^{j'}$. As a result, $\mathfrak{Q}_{t+1}^k(x_t^k, \xi_{t+1}^j) = Q_{t+1}(x_t^k, \xi_{t+1}^j)$. With corresponding dual variables, $(\hat{\pi}_{t+1}, \hat{z}_{t+1})$, and the equations for $\hat{G}_t$ and $\hat{g}_t$ indicated in (i) we therefore have, by strong duality, that

$$
\begin{aligned}
&\hat{\pi}_{t+1}B_{t+1}x_t^k + \hat{\pi}_{t+1}\xi_{t+1}^j + \sum_{i',j'\in[n_{t+2}]}\sum_{\kappa\in\mathcal{K}_{t+2}^{j',k}} g_{t+1,\kappa}^{j'}\,\hat{z}_{t+1,\kappa}^{i',j'}\\
&\quad= \hat{G}_t x_t^k + \hat{g}_t\\
&\quad= Q_{t+1}(x_t^k, \xi_{t+1}^j)\\
&\quad= \max_{\kappa\in\mathcal{K}_{t+1}^j}\,[G_{t,\kappa}^j x_t^k + g_{t,\kappa}^j]\\
&\quad> d_{t+1}^{i,j}\gamma_t^k + \nu_t^{i,k},
\end{aligned}
$$

where the inequality holds for some $i, j \in [n_{t+1}]$ by the constraint-violation hypothesis. Thus (i) holds in the latter case. ∎

The remainder of the proof relies on a line of argument developed in Philpott and Guan [30].

Lemma 4.4. *Assume* (A.1)–(A.3)*. Consider a variant of Algorithm* 3.1 *in which we repeatedly sample all* $\prod_{t=2}^T n_t$ *scenarios in the forward pass in a deterministic order. Then, the algorithm converges to an optimal solution of* (2.1) *in a finite number of iterations.*

Proof. By Lemma 3.1, we have that for all $t = 2, \ldots, T$, $Q_t(\cdot, \xi_t)$ is a piecewise linear convex function in x_{t-1} with a finite number of pieces. For any sequence of stage $T-1$ solutions x^k_{T-1} generated by the deterministic variant of Algorithm 3.1, the fact that the backward pass uses extreme-point dual solutions implies there exists $\bar{k}_{T-1}$ such that no new cuts are generated for stage $T-1$ for $k > \bar{k}_{T-1}$. Lemma 4.2, coupled with the consistent primal tie-breaking assumption (A.3), then implies that $\mathfrak{Q}^k_T(x^k_{T-1}, \xi^j_T) = Q_T(x^k_{T-1}, \xi^j_T)$ for all $j \in [n_T]$ and all $k > \bar{k}_{T-1}$.

We now similarly employ Lemmas 3.1 and 4.3 in inductive fashion to argue that after a finite number of iterations, $\bar{k}_t$, we have $\mathfrak{Q}^k_{t+1}(x^k_t, \xi^j_{t+1}) = Q_{t+1}(x^k_t, \xi^j_{t+1})$ for all $j \in [n_{t+1}]$ and $k > \bar{k}_t$ for $t = T-2, T-3, \ldots, 1$. Thus for $k > \bar{k}_t$, $(x^k_t, \gamma^k_t, \nu^k_t)$ solves problem (3.6a), given its input, (x^k_{t-1}, ξ^j_t). In particular, $(x^k_1, \gamma^k_1, \nu^k_1)$ solves problem (3.6a) for $t = 1$, given its input (x^k_0, ξ_1) for all $k > \bar{k}_1$. The desired result follows given that under (A.1)–(A.2), problem (3.6a) is an equivalent reformulation of models (2.1) for $t = 1$ and (2.2) for $t = 2, \ldots, T-1$. ∎

The proof of the following result again follows in the vein of [15, 30].

THEOREM 4.5. *Assume* (A.1)–(A.3). *Consider Algorithm* 3.1 *except that we sample forward passes according to Algorithm* 3.2 *with* $\beta \in [0, 1)$*, and we remove the termination condition of step* 10. *Then, the algorithm converges to an optimal solution of model* (2.1) *in a finite number of iterations with probability one.*

Proof. First, we note that Lemma 4.4 again holds if the stage $t = 1, 2, \ldots, T-1$ problems (3.9) are initialized with any set of valid lower-bounding cuts, even if the specific values of $\bar{k}_t$, $t = 1, 2, \ldots, T-1$, may differ. We know by Lemma 4.1 that the algorithm produces such cuts.

Given assumptions (A.1) and (A.3) there are finitely many values of (x^k_{t-1}, ξ^j_t) that problems (3.9) take as input at each stage. With probability one, in the course of running the algorithm there is an iteration after which no more cuts are added.

By Lemma 4.4, if Algorithm 3.1 follows a prespecified deterministic order of the $\prod_{t=2}^T n_t$ scenarios a sufficient number of times, we obtain an optimal solution. By the second Borel–Cantelli lemma (e.g., [19]), Algorithm 3.1, under the sampling procedure of Algorithm 3.2 with $\beta < 1$ and assumption (A.1), will almost surely sample this sequence after the last iteration in which a cut has been added. Hence, we obtain the desired result. ∎

We note that while Theorem 4.5 establishes that Algorithm 3.1 converges, it does not otherwise characterize the quality of the algorithm or suggest criteria to terminate the algorithm.

5. Extensions.

The uncertainty set in (3.2) can be extended to include a broader family of distributions as well as alternatives to the Wasserstein distance. In this section we outline some extensions based on the Wasserstein metric as well as extensions involving phi-divergence.

5.1. Hedging against additional points of support.

The uncertainty set (3.2) requires as input $\Xi^{n_t}_t$ and the corresponding probability mass function $q^i_t > 0$, $i \in [n_t]$. Here, we allow nature to redistribute mass to a finite superset $\hat{\Xi}_t \supseteq \Xi^{n_t}_t$. Choosing $\hat{\Xi}_t$ as a superset ensures feasibility of model (3.3) for all values of the radius r. If the set $\Xi^{n_t}_t$ is specified by observed data then $\hat{\Xi}_t$ can include additional plausible realizations, and in particular we may wish to include "tail" events that have not been observed but about which we have concern. Such points could arise via expert elicitation, by fitting a plausible distribution, and sampling points outside of $\Xi^{n_t}_t$, etc.

Let $[\hat{n}_t]$ index the realizations of $\hat{\Xi}_t$. Then, the Wasserstein metric transports probability mass from $\Xi_t^{n_t}$ to $\hat{\Xi}_t$ and is defined as

$$d_W(\mathbb{Q}_t^{n_t}, \mathbb{P}_t) = \min_z \sum_{i\in[n_t]} \sum_{j\in[\hat{n}_t]} d_t^{i,j} z^{i,j} \tag{5.1a}$$

$$\text{s.t.} \sum_{j\in[\hat{n}_t]} z^{i,j} = q_t^i \quad \forall i \in [n_t], \tag{5.1b}$$

$$\sum_{i\in[n_t]} z^{i,j} = p_t^j \quad \forall j \in [\hat{n}_t], \tag{5.1c}$$

$$z^{i,j} \geq 0 \quad \forall i \in [n_t], j \in [\hat{n}_t]. \tag{5.1d}$$

We extend the inner maximization over $\mathbb{P}_t \in \mathcal{P}_t$ via

$$\mathcal{P}_t = \{\mathbb{P}_t \in \mathcal{M}(\hat{\Xi}_t) : d_W(\mathbb{Q}_t^{n_t}, \mathbb{P}_t) \leq r\}.$$

With this modification the analog of the SDDP subproblem (3.9) becomes

$$\mathfrak{Q}_t^k(x_{t-1}, \xi_t) = \min_{x_t,\gamma_t,\nu_t} c_t x_t + r\gamma_t + \sum_{i\in[n_{t+1}]} q_{t+1}^i \nu_t^i \tag{5.2a}$$

$$\text{s.t. } A_t x_t = B_t x_{t-1} + \xi_t, \tag{5.2b}$$

$$d_{t+1}^{i,j}\gamma_t + \nu_t^i \geq G_{t,\kappa}^j x_t + g_{t,\kappa}^j \quad \forall i \in [n_{t+1}], j \in [\hat{n}_{t+1}], \kappa \in \mathcal{K}_{t+1}^{j,k}, \tag{5.2c}$$

$$x_t \geq 0, \tag{5.2d}$$

$$\gamma_t \geq 0. \tag{5.2e}$$

An analog of Algorithm 3.1 follows immediately. To ensure convergence, a dynamic sampling algorithm would require positive weight that is uniformly bounded away from zero, on all realizations of the superset $\hat{\Xi}_t$ at each stage, t.

5.2. Incorporating bounds on moments. Let $\xi_t(\ell), \ell = 1, \ldots, d_{\xi,t}$, denote the components of the stage-t random vector ξ_t. We can include in the uncertainty set, bounds on the moments of ξ_t, for example, as follows:

$$\underline{m}_t^1(\ell) \leq \mathbb{E}_{\mathbb{P}_t}\xi_t(\ell) \leq \bar{m}_t^1(\ell),\ \ell \in [d_{\xi,t}],$$
$$\underline{m}_t^2(\ell,\ell') \leq \mathbb{E}_{\mathbb{P}_t}\left[\xi_t(\ell)\xi_t(\ell')\right] \leq \bar{m}_t^2(\ell,\ell'),\ \ell \leq \ell' \in [d_{\xi,t}],$$

where $\underline{m}_t^1(\ell), \bar{m}_t^1(\ell), \underline{m}_t^2(\ell,\ell'), \bar{m}_t^2(\ell,\ell')$, $\ell \leq \ell' \in [d_{\xi,t}]$, are user-specified bounds. In our setting these correspond to

$$\underline{m}_t^1(\ell) \leq \sum_{j\in[n_t]} p_t^j \xi_t^j(\ell) \leq \bar{m}_t^1(\ell),\ \ell \in [d_{\xi,t}], \tag{5.3a}$$

$$\underline{m}_t^2(\ell,\ell') \leq \sum_{j\in[n_t]} p_t^j \xi_t^j(\ell)\xi_t^j(\ell') \leq \bar{m}_t^2(\ell,\ell'),\ \ell \leq \ell' \in [d_{\xi,t}]. \tag{5.3b}$$

We can extend the definition of the uncertainty set to

$$\mathcal{P}_t = \{\mathbb{P}_t \in \mathcal{M}(\Xi_t) : d_W(\mathbb{Q}_t^{n_t}, \mathbb{P}_t) \leq r,\ (5.3a)\text{-}(5.3b)\}.$$

The form of (5.3a)–(5.3b) allows us to derive a linear programming analog of subproblem (3.9), which accounts for the bounds on these moments. Of course, if the bounds are too large, we could violate positive semidefiniteness of the would-be covariance matrix. Upper bounding a covariance matrix, rather than component-by-component bounds, can lead to various conic reformulations; see, e.g., [11, 12, 14, 20].

5.3. Other distance metrics. In addition to the Wasserstein metric, $\mathcal{P}_t$ can be defined in terms of other distance metrics or proxies thereof. Total variation is defined as $d_{TV}(\mathbb{Q}_t^{n_t}, \mathbb{P}_t) = \sum_{j\in[n_t]} |q_t^j - p_t^j|$, and the modified χ^2 distance is $d_{\chi^2}(\mathbb{Q}_t^{n_t}, \mathbb{P}_t) = \sum_{j\in[n_t]} (q_t^j - p_t^j)^2/q_t^j$ (see, e.g., [32]). The former can be captured directly within our framework with linear programming constructs. For the latter the analog of (3.3) is

$$\max_{p_{t+1}} \sum_{j\in[n_{t+1}]} p_{t+1}^j Q_{t+1}(x_t, \xi_{t+1}^j) \tag{5.4a}$$

$$\text{s.t.} \sum_{j\in[n_{t+1}]} (q_{t+1}^j - p_{t+1}^j)^2/q_{t+1}^j \leq r, \tag{5.4b}$$

$$\sum_{j\in[n_{t+1}]} p_{t+1}^j = 1, \tag{5.4c}$$

$$p_{t+1}^j \geq 0 \quad \forall j \in [n_{t+1}]. \tag{5.4d}$$

Model (5.4) can be expressed as a second-order conic program (SOCP). Hence, the analogs of model (3.3) and SDDP subproblem (3.9) are, in turn, SOCPs also facilitating a variant of Algorithm 3.1 in which we solve SOCPs in place of linear programs.

5.4. Interstage independence and dependence. Assumption (A.1) requires that ξ_t, $t = 2, \ldots, T$, be interstage independent. After having solved a multistage problem, there is a stochastic process that corresponds to nature's constrained worst-case choice of distribution at each node in the scenario tree. Ironically, that stochastic process is *not* interstage independent. This occurs because nature can adapt to the path dependent x_{t-1} in the inner maximization problem (3.3). This structure arises because of the nested form of problem (2.1)–(2.2) coupled with the "rectangular" nature of the DUSs in assumption (A.1). An interesting variant of the problem would enforce interstage independence on $\mathbb{P}_t$ for $t = 2, \ldots, T$. Doing so would appear to require coupling nature's "$\max_{\mathbb{P}_{t+1}\in\mathcal{P}_{t+1}}$" across all stage t nodes, but the decoupled form of that maximization is key to the validity of Algorithm 3.1. Restated, even though the nominal stochastic process for ξ_t, $t = 2, \ldots, T$, is interstage independent, our formulation yields a distributionally robust stochastic process that is interstage dependent. Our algorithm is unable to handle the case in which we require the distributionally robust stochastic process to be interstage independent.

6. Numerical experiments.

6.1. Problem statement. We consider a multistage hydro-thermal generation problem with R reservoirs in series, in which the randomness comes from the exogenous inflow of water into some of the reservoirs. Our model is based on that in Wahid [40], but we incorporate a cost function for using thermal generation as opposed to maximizing revenue. Here, we consider instances of model (2.1) under the Wasserstein-based uncertainty set, and we employ variants of Algorithm 3.1 as we describe below. The parameters and variables of the model are summarized in Table 1.

Table 1
Notation for hydro-generation model.

Sets	
$r \in \mathcal{R} = \{1, \dots, R\}$	set of reservoirs
$l \in \mathcal{L}_r$	set of efficiency levels for hydroelectric generation for reservoir r
Parameters	
$c_t(\cdot)$	piecewise linear, convex thermal generation cost function in stage t
d_t	demand for electricity
$p_{r,l}$	power generation at efficiency level l for reservoir r
$f_{r,l}$	water flow at efficiency level l from reservoir r
l_r	minimum inventory of water for reservoir r
u_r	maximum inventory of water for reservoir r
$\xi_{t,r}$	random inflow for reservoir r in stage t
Decision variables	
$x_{t,r}$	inventory of water in reservoir r at the end of stage t
$o_{t,r}$	outflow of water for hydropower from reservoir r in stage t
$s_{t,r}$	spillage from reservoir r in stage t
h_t	hydropower generation in stage t
g_t	thermal power generation in stage t
$z_{t,r,l}$	usage (%) of efficiency level l from reservoir r in stage t
Boundary conditions	
$o_{t,0} = s_{t,0} = 0$; $x_{0,r}$ given	

The model is specified by the following cost-to-go function for $t = 1, \dots, T$ with $Q_{T+1} = 0$:

$$Q_t(x_{t-1}, \xi_t) = \min c_t(g_t) + \max_{\mathbb{P}_{t+1} \in \mathcal{P}_{t+1}} \mathbb{E}_{\mathbb{P}_{t+1}} \left[Q_{t+1}(x_t), \xi_{t+1}) \right] \tag{6.1a}$$

$$\text{s.t.} \quad x_{t,r} = x_{t-1,r} + \xi_{t,r} - o_{t,r} - s_{t,r} + o_{t,r-1} + s_{t,r-1} \quad \forall r \in \mathcal{R}, \tag{6.1b}$$

$$h_t + g_t = d_t, \tag{6.1c}$$

$$h_t = \sum_{r \in \mathcal{R}} \sum_{l \in \mathcal{L}_r} p_{r,l} z_{t,r,l}, \tag{6.1d}$$

$$o_{t,r} = \sum_{l \in \mathcal{L}_r} f_{r,l} z_{t,r,l} \quad \forall r \in \mathcal{R}, \tag{6.1e}$$

$$\sum_{l \in \mathcal{L}_r} z_{t,r,l} \le 1 \quad \forall r \in \mathcal{R}, \tag{6.1f}$$

$$l_r \le x_{t,r} \le u_r \quad \forall r \in \mathcal{R}, \tag{6.1g}$$

$$0 \le z_{t,r,l} \le 1 \quad \forall r \in \mathcal{R}, l \in \mathcal{L}_r, \tag{6.1h}$$

$$o_{t,r}, s_{t,r} \ge 0 \quad \forall r \in \mathcal{R}, \tag{6.1i}$$

$$g_t \ge 0. \tag{6.1j}$$

Equation (6.1a) models the cost incurred by thermal generation plus the cost of future stages. Equation (6.1b) conserves flow of water for each reservoir, updating the water storage according to exogenous and upstream inflows as well as outflows due to hydrogeneration and spills. Equation (6.1c) requires we satisfy demand with hydro and/or thermal generation. Equations (6.1d)–(6.1f) model the power generation of each reservoir as a piecewise linear concave function of the outflow of water; i.e., (f_{rl}, p_{rl}), $l \in \mathcal{L}_r$, are the breakpoints of such a function for reservoir r. Equations (6.1g)–(6.1j) provide simple bounds on the variables.

For our experimental setup, the inflows $\xi_{t,r}$ are interstage independent, but have dependence across reservoirs within each stage. We assume that $\xi_{t,r}$ has a transformed

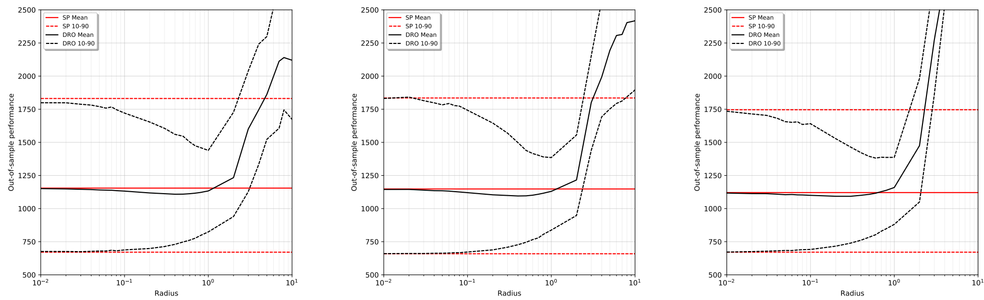
TABLE 2
Numerical values for the parameters in model (6.1).

Energy demand	$d_{\cdot} = (117, 117, 117, 117, 117, 176, 293, 176, 117, 117, 117, 117)$
Random inflows	$b_{\cdot,r} = (5, 5, 5, 15, 5, 5, 5, 5, 5, 10, 5, 5)\ \forall r \in \{1, 4, 7\}$, 0 otherwise. $\mu_{\cdot,r} = (0.6, 0.6, 0.6, 1.5, 0.6, 0.6, 0.6, 0.6, 0.6, 1.0, 0.6, 0.5)\ \forall r \in \{1, 4, 7\}$, 0 otherwise. $\sigma_{\cdot,r} = (0.3, 0.3, 0.3, 0.5, 0.3, 0.3, 0.3, 0.3, 0.3, 0.5, 0.3, 0.4)\ \forall r \in \{1, 4, 7\}$, 0 otherwise. $\rho_{t,r,r'} = 0.9\ \forall r, r' \in \{1, 4, 7\}, r \neq r'$
Reservoir bounds	$u_r = 120$ and $l_r = 20\ \forall r \in \mathcal{R}$
Hydropower generation	$\mathcal{L}_r = \{1, 2, 3\}\ \forall r \in \mathcal{R}$ $p_{r,1} = 11, p_{r,2} = 26,$ and $p_{r,3} = 50\ \forall r \in \mathcal{R}$ $f_{r,1} = 10, f_{r,2} = 25,$ and $f_{r,3} = 50\ \forall r \in \mathcal{R}$
Thermal cost	$c_t(g_t) = \max\{g_t, 2g_t - 10, 8g_t - 130\}$

lognormal distribution of the form $\max\{b_{t,r} - \exp(\psi_{t,r}), 0\}$, where $b_{t,s} \in \mathbb{R}_+$ and $\psi_{t,r}$, $r \in \mathcal{R}$, has a multivariate normal distribution with mean $\mu_{r,t}$, variance $\sigma^2_{r,t}$, and correlation coefficient $\rho_{t,r,r'}$. That said, we again assume we only have access to n_t realizations of ξ_t for each stage t. For out-of-sample tests we draw additional realizations of the random vectors from the transformed lognormal distribution. In the computation that follows we use $R = 10$, $T = 48$ months, exogenous inflow into three of the 10 reservoirs, and all out-of-sample tests use the same 1000 sample paths through the 48-stage tree. Table 2 summarizes the values of the parameters we use in our model. Temporal parameters are presented for the first 12 months, and it is assumed they repeat in the following years. Aside from the model parameters, we establish a maximum number of 110 cuts per outcome ($j \in [n_t]$) for each subproblem in the multicut implementations, and $110 \times n_t$ for the single-cut implementation. When the cut limit is reached, 10% of the oldest cuts are removed.

6.2. Effect of robustness. To assess the potential value of our DRO, we conduct an out-of-sample test. As we discuss in section 3.3, the cuts obtained when running variants of Algorithm 3.1 provide a policy that can be used on new realizations of inflows. For a given value of the uncertainty radius, r, we run the algorithm for a fixed budget of computational time, namely, 3600 seconds. The cost incurred with the policy obtained after that amount of computational effort is then simulated in an out-of-sample test. Figure 1 shows the out-of-sample performance for different numbers of realizations per stage (i.e., $n_t = 5$, 10, and 40) and for different values of the Wasserstein radius r (i.e., $r \in \{a10^b : a = 1, \ldots, 9; b = -2, \ldots, 1\}$). For each plot, we show in black the results of the expected cost as well as the 10th percentile and 90th percentile of the cost over the 1000 realizations of the out-of-sample simulation. As a benchmark, we plot in red the same performance metrics for the policy obtained from the nominal SP, i.e., from solving the DRO-based model with $r = 0$. We note that while $n_t = 5$ may seem small, there are 47 sets of such realizations across the stages, and 5^{47} scenarios in the resulting multistage scenario tree.

From Figure 1 we see that for a range of values of r for each n_t, the distributionally robust policy outperforms the policy of the nominal SP for the mean cost. Moreover, the robust policy tends to reduce the 90th percentile of cost more significantly compared with the nominal SP's policy. The overall out-of-sample performance improves as n_t grows. We attribute the fact that the robust policy exhibits better out-of-sample performance to "overtraining" of the stochastic programming policy to the specific outcomes in $\Xi_t^{n_t}$ and, consistent with this, the improvement of the DRO approach over the nominal approach decreases as n_t grows. Empirically, improved out-of-sample performance depends on our choice of the distribution governing inflows. Our truncated lognormal has a left skew, and low inflows lead to high costs.



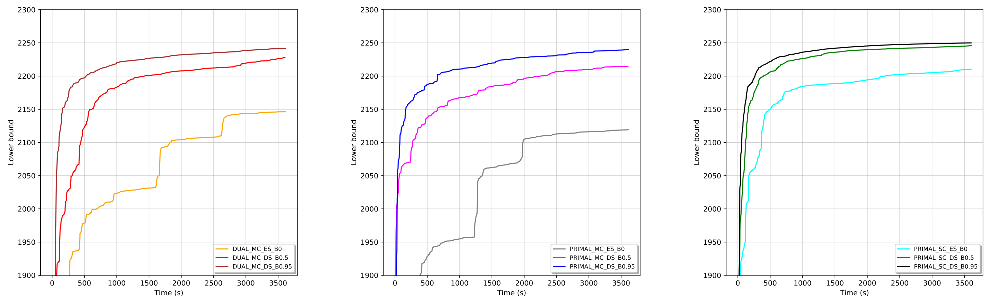
(a) n_t=5 (b) n_t=10 (c) n_t=40

FIG. 1. *Out-of-sample assessment of policies obtained for different values of* n_t.

Separate experiments with a (symmetric) uniform distribution show no benefit for DRO in out-of-sample tests. These results align with conclusions of Anderson and Philpott [1] regarding skewed distributions with heavy tails, even though the mathematical form of our recourse function does not coincide with assumptions made in their formal results.

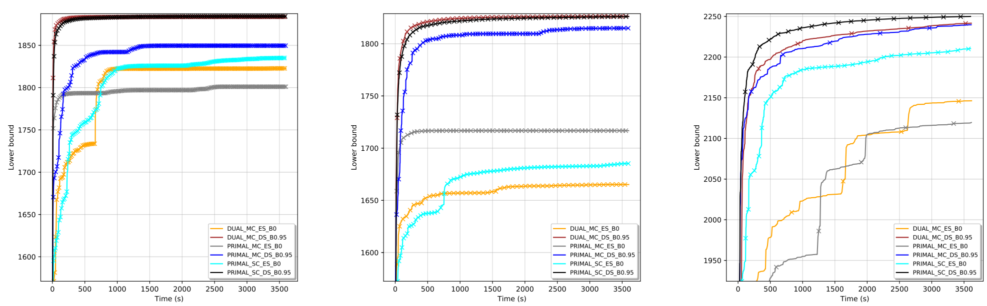
6.3. Algorithmic performance. In section 3.5 we discuss four possible algorithms involving either a multicut or single-cut procedure coupled with taking a dual or using a side computation (henceforth we call the latter the "primal" approach, for brevity) to handle the DRO aspect of the problem. For reasons we discuss in section 3.5 we do not consider the single-cut/dual combination, leaving three algorithms. These can be run by sampling under the empirical (nominal) distribution or by the dynamic sampling procedure we discuss in section 3.4. This leads to six algorithms, whose performance we assess here. In all cases we use a Wasserstein-based uncertainty set specified by constraints (3.3b)–(3.3e), without any of the extensions of section 5. In our implementation, worst-case probabilities are computed in the backward pass of SDDP to form a cut as we describe in Algorithm 3.1. For both the single-cut and multicut versions of the primal algorithm, to employ dynamic sampling we require the worst-case distribution from (3.11). We compute this by solving model (3.3) in the forward pass, with $\mathfrak{Q}^k_{t+1}(\hat{x}_t, \xi^j_{t+1})$ replacing $Q_{t+1}(\hat{x}_t, \xi^j_{t+1})$ for all $j \in [n_{t+1}]$. Note that $\mathfrak{Q}^k_{t+1}(\hat{x}_t, \xi^j_{t+1})$ is readily available in the forward pass for the multicut variant through the epigraph variables; however, for the single-cut variant we need to solve the descendants of a sample path to compute $\mathfrak{Q}^k_{t+1}(\hat{x}_t, \xi^j_{t+1})$. In addition to the primal and dual designations just discussed, in what follows we abbreviate dynamic sampling as DS, empirical sampling as ES, multicut as MC, and single-cut as SC. Throughout the DS experiments in this section, we use a mixture parameter of $\beta \in \{0.5, 0.95\}$ in (3.12), and note that ES corresponds to $\beta = 0$.

Lower bound. We compare the growth of the lower bounds obtained by each of the algorithmic variants over a fixed computational budget of 3600 seconds. For this experiment we use $n_t = 40$ to evaluate the performance, and choose $r = 1$ which is, to order of magnitude, the value that we would choose from the out-of-sample experiment. Figure 2 shows the evolution of the lower bound for different variants of the algorithm, and within the DS variants $\beta = 0.5$ and 0.95 are annotated in the legends as `B0.5` and `B0.95`, respectively. We emphasize that the "lower bound" reported here is a lower bound on the distributionally robust model, and the out-of-sample cost estimates in section 6.2 are for the risk-neutral expected cost.



(a) Dual multicut (b) Primal multicut (c) Primal single-cut

FIG. 2. *Lower bound evolution of different algorithmic variants.*



(a) $n_t = 5$ (b) $n_t = 10$ (c) $n_t = 40$

FIG. 3. *Lower bound evolution for different values of* n_t.

There are multiple trade-offs at play in the algorithms we consider. First, there is a trade-off between tighter lower bounds typically afforded by multicut algorithms, relative to their single-cut counterparts, after the same number of iterations, and the increased computational effort associated with larger multicut subproblems. Second, there is a similar trade-off between the tighter lower bounds provided by the dual approach, relative to the primal approach, versus the larger size of their subproblems. Third, under the primal single-cut approach, dynamic sampling requires an auxiliary computation in the forward pass to obtain the worst-case probabilities. The most striking aspect of Figure 2 is the benefit provided by dynamic sampling across the three algorithmic variants. We also note that in all three cases, $\beta = 0.95$ dominates $\beta = 0.5$, but the relative merit of this is dampened in the primal single-cut variant.

We now compare the algorithms for different choices of n_t. Figure 3 shows the lower bound for $n_t = 5, 10$, and 40. We drop all the experiments with $\beta = 0.5$ for clarity in the plots and added markers to identify every 50 iterations of SDDP, which are most easily visible when $n_t = 40$.

Figure 3 shows that using dynamic sampling in any of the variants is better than the empirical sampling counterpart across all three values of n_t. For smaller values of n_t, the dual variant with dynamic sampling performs better than the other variants, but this changes as n_t grows to 40. For small values of n_t, our dual DRO reformulation has subproblems of relatively modest size, which do not hinder the algorithm's performance. However, the single-level dual reformulation in (3.9) has a quadratic number of cut constraints in n_t, and so relative performance does deteriorate as n_t

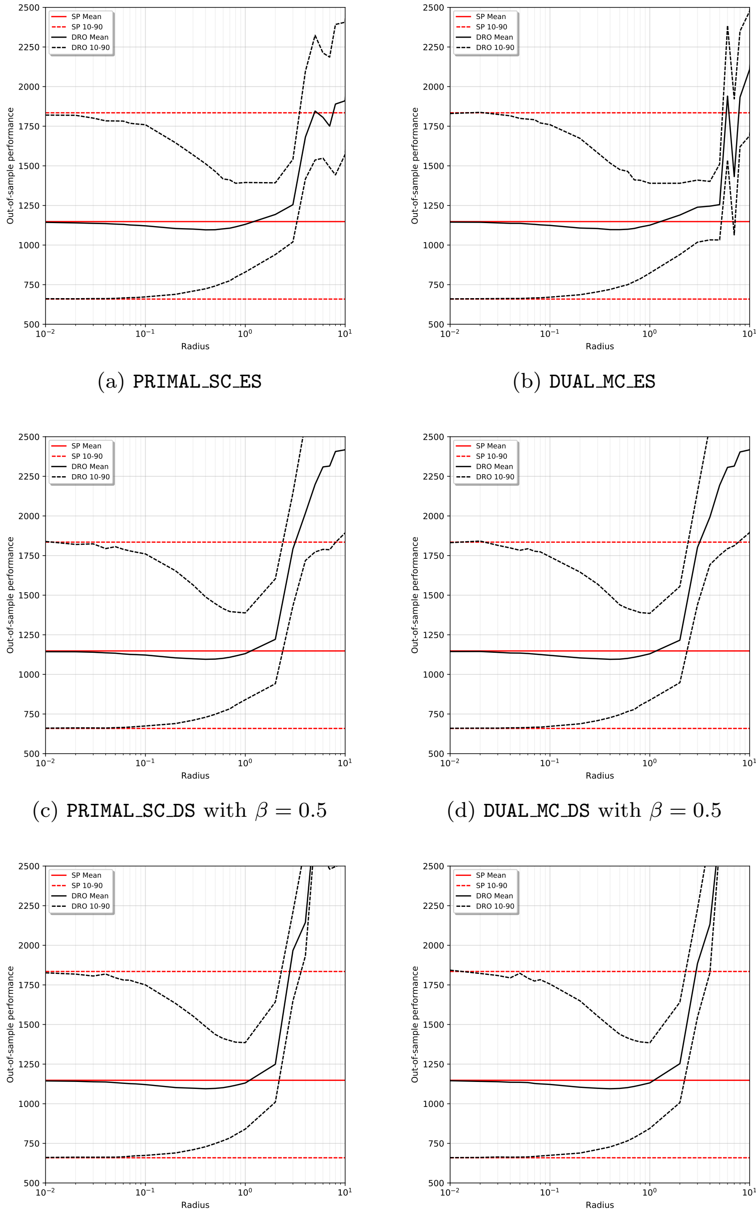
grows. Empirically, the primal multicut variant results in a dominated compromise: Its bound is not as tight as the dual multicut counterpart, and it performs iterations more slowly than the primal single-cut algorithm. Interestingly, the primal single-cut variant performs well when using dynamic sampling even though the forward pass is computationally more intensive. That said, the size of its subproblems is smaller, allowing more iterations in the same amount of time, which can yield a better approximation of the cost-to-go function. It seems clear that the preferred approach will depend on the problem instance as well as n_t and, hence, there is value in having access to a family of algorithms. The trade-off between primal and dual multicut algorithms may change as the number of scenarios grows: The sizes of the subproblems for the dual approach grow quadratically in n_t while those for the multicut primal approach grow linearly in n_t.

Policy evaluation. While obtaining tighter lower bounds quickly is valuable in terms of the algorithm converging in practice, there is a separate question as to whether the policies associated with tighter lower bounds perform better in out-of-sampling testing. Here, we investigate the analog of our tests from section 6.2 except that we compare the out-of-sample performance of the policies obtained by different variants of the algorithm. All variants are solving the same underlying problem, and thus their cut-based policies are identical upon convergence. However, as a practical matter, convergence is difficult to assess and out-of-sample performance is driven by the final set of cuts obtained on termination, which differ across the algorithmic variants and the selection of β. We compare the single-cut primal algorithm, i.e., the method proposed in [32], and our dual multicut approach. For each variant, we plot the out-of-sample performance contrasting the effects of dynamic sampling using $n_t = 10$. Figure 4 summarizes these results.

The algorithmic configurations in Figure 4 offer similar out-of-sample performance for values of the Wasserstein radius up to about $r = 1$. As we further increase this parameter, the performance varies depending on the selection of β, and to a smaller degree, on the selection of the primal versus dual algorithm. If we are interested in improving out-of-sample *expected* performance or if we are risk averse and focused on the most costly 0.10-level tail, there appears to be little value in having $r > 1$ on our instances. That said, there also appears to be some trend that performance at large values of r is better when β is smaller. This is plausible in the sense that the policy is trained more narrowly on the worst-case scenarios when β and r are both large. In other words, smaller values of β lead to policies that are better trained on the entire scenario tree and, hence, appear to be better equipped to hedge against out-of-sample scenarios.

With this caveat in place, for our purposes of being risk neutral or moderately risk averse, we care about the performance in the range of the "best" values of r, even if we do not know this range ahead of time. In light of this, it seems sensible to opt for a configuration of the algorithm that uses dynamic sampling, as better lower bounds do suggest that it focuses on relevant parts of the scenario tree when r is moderate. Both the primal and dual approaches are competitive in the range of values that we experimented with for n_t, and in our view selecting one over the other one should be guided by the amount of data available to build the scenario tree, i.e., the size of n_t.

7. Conclusions. We have developed a family of algorithms, rooted in SDDP, for solving a class of distributionally robust multistage stochastic linear programs under the Wasserstein metric. In particular, we propose a multicut algorithm that dualizes nature's choice of a constrained worst-case probability distribution, and we also pro-



(a) `PRIMAL_SC_ES` (b) `DUAL_MC_ES`

(c) `PRIMAL_SC_DS` with $\beta = 0.5$ (d) `DUAL_MC_DS` with $\beta = 0.5$

(e) `PRIMAL_SC_DS` with $\beta = 0.95$ (f) `DUAL_MC_DS` with $\beta = 0.95$

FIG. 4. *Out-of-sample assessment of policies obtained for different variants of SDDP using* $n_t = 10$.

pose a multicut primal algorithm. In addition to establishing finite convergence with probability one, we compare these algorithms with the single-cut primal algorithm of Philpott, de Matos, and Kapelevich [32], both theoretically in terms of the tightness of the respective lower bounds and computationally. Moreover, we compare results when sampling from the empirical distribution and from a mixture of the empirical distribution and an estimate of the worst-case distribution. Computational results suggest that there can be merit to the latter sampling scheme for important values of the Wasserstein radius in terms of producing tighter lower bounds. For reasonable values of the Wasserstein radius, the alternatives perform similarly in out-of-sample tests.

REFERENCES

[1] E. J. Anderson and A. B. Philpott, *Robust Sample Average Approximation with Small Sample Sizes*, http://www.optimization-online.org/DB_HTML/2019/02/7092.html (2019).

[2] M. Bansal, K.-L. Huang, and S. Mehrotra, *Decomposition algorithms for two-stage distributionally robust mixed binary programs*, SIAM J. Optim., 28 (2018), pp. 2360–2383.

[3] G. Bayraksan and D. Love, *Data-driven stochastic programming using phi-divergences*, in The Operations Research Revolution, Tutorials Oper. Res. 45, D. Aleman and A. Thiele, eds., INFORMS, Hanover, MD, 2015, pp. 1–19.

[4] D. Bertsimas, X. Doan, K. Natarajan, and C.-P. Teo, *Models for minimax stochastic linear optimization problems with risk aversion*, Math. Oper. Res., 35 (2010), pp. 580–602.

[5] D. Bertsimas, V. Gupta, and N. Kallus, *Data-driven robust optimization*, Math. Program., 167 (2018), pp. 235–292.

[6] D. Bertsimas, V. Gupta, and N. Kallus, *Robust sample average approximation*, Math. Program., 171 (2018), pp. 217–282.

[7] J. Birge and F. Louveaux, *A multicut algorithm for two-stage stochastic linear programs*, European J. Oper. Res., 34 (1988), pp. 384–392.

[8] J. Blanchet and K. Murthy, *Quantifying distributional model risk via optimal transport*, Math. Oper. Res., 44 (2019), pp. 565–600.

[9] J. Blanchet, K. Murthy, and F. Zhang, *Optimal Transport Based Distributionally Robust Optimization: Structural Properties and Iterative Schemes*, https://arxiv.org/abs/1810.02403 (2018).

[10] G. Calafiore and L. El-Ghaoui, *On distributionally robust chance-constrainted linear programs*, J. Optim. Theory Appl., 130 (2006), pp. 1–22.

[11] Z. Chen, M. Sim, and P. Xiong, *Robust stochastic optimization made easy with RSOME*, Manag. Sci., 66 (2020), pp. 3329–3339.

[12] Z. Chen, M. Sim, and H. Xu, *Distributionally robust optimization with infinitely constrained ambiguity sets*, Oper. Res., 67 (2019), pp. 1328–1344.

[13] A. R. de Queiroz and D. P. Morton, *Sharing cuts under aggregated forecasts when decomposing multi-stage stochastic programs*, Oper. Res. Lett., 41 (2013), pp. 311–316.

[14] E. Delage and Y. Ye, *Distributionally robust optimization under moment uncertainty with application to data–driven problems*, Oper. Res., 55 (2010), pp. 98–112.

[15] A. Downward, O. Dowson, and R. Baucke, *Stochastic dual dynamic programming with stagewise-dependent objective uncertainty*, Oper. Res. Lett., 48 (2020), pp. 33–39.

[16] J. Duchi, P. Glynn, and H. Namkoong, *Statistics of Robust Optimization: A Generalized Empirical Likelihood Approach*, https://arxiv.org/abs/1610.03425 (2016).

[17] R. Gao and A. Kleywegt, *Distributionally robust stochastic optimization with Wasserstein distance*, preprint, https://arxiv.org/abs/1604.02199 (2016).

[18] A. Georghiou, A. Tsoukalas, and W. Wiesemann, *Robust dual dynamic programming*, Oper. Res., 67 (2016).

[19] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 2nd ed., Oxford University Press, Oxford, 1992.

[20] G. Hanasusanto and D. Kuhn, *Conic programming reformulations of two-stage distributionally robust linear programs over Wasserstein balls*, Oper. Res. 66 (2018), pp. 849–869.

[21] J. Huang, K. Zhou, and Y. Guan, *A study of distributionally robust multistage stochastic optimization*, https://arxiv.org/abs/1708.07930 (2017).

[22] G. Infanger and D. P. Morton, *Cut sharing for multistage stochastic linear programs with interstage dependency*, Math. Program., 75 (1996), pp. 241–256.

[23] V. Kozmík and D. P. Morton, *Evaluating policies in risk-averse multi-stage stochastic programming*, Math. Program., 152 (2015), pp. 275–300.

[24] H. Lam, *Recovering best statistical guarantees via the empirical divergence-base distributionally robust optimization*, Oper. Res., 4 (2019), pp. 1090–1105.

[25] H. Lam and E. Zhou, *The empirical likelihood approach to quantifying uncertainty in sample average approximation*, Oper. Res. Lett., 4 (2017), pp. 301–307.

[26] V. Leclère, P. Carpentier, J.-P. Chancelier, A. Lenoir, and F. Pacaud, *Exact converging bounds for stochastic dual dynamic programming via Fenchel duality*, SIAM J. Optim., 30 (2020), pp. 1223–1250, http://www.optimization-online.org/DB_FILE/2018/04/6575.pdf.

[27] F. Luo and S. Mehrotra, *Decomposition algorithm for distributionally robust optimization using Wasserstein metric*, European J. Oper. Res., 167 (2019), pp. 20–35.
[28] P. Mohajerin Esfahani and D. Kuhn, *Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulation*, Math. Program., 171 (2018), pp. 115–166.
[29] M. V. F. Pereira and L. M. V. G. Pinto, *Multi-stage stochastic optimization applied to energy planning*, Math. Program., 52 (1991), pp. 359–375.
[30] A. Philpott and Z. Guan, *On the convergence of stochastic dual dynamic programming and related methods*, Oper. Res. Lett., 36 (2008), pp. 450–455.
[31] A. B. Philpott, V. de Matos, and E. Finardi, *On solving multistage stochastic programs with coherent risk measures*, Oper. Res., 61 (2013), pp. 957–970.
[32] A. B. Philpott, V. L. de Matos, and L. Kapelevich, *Distributionally robust SDDP*, Comput. Manag. Sci., 15 (2018), pp. 431–454.
[33] H. Rahimian, G. Bayraksan, and T. H. de Mello, *Controlling risk and demand ambiguity in newsvendor models*, European J. Oper. Res., 279 (2019), pp. 854–868.
[34] H. Rahimian, G. Bayraksan, and T. H. de Mello, *Identifying effective scenarios in distributionally robust stochastic programs with total variation distance*, Math. Program., 173 (2019), pp. 393–430.
[35] H. Rahimian and S. Mehrotra, *Distributionally robust optimization: A review*, https://arxiv.org/abs/1908.05659 (2019).
[36] C. Shang and F. You, *Distributionally robust optimization for planning and scheduling under uncertainty*, Comput. Chem. Eng., 110 (2018), pp. 53–68.
[37] A. Shapiro, *On duality theory of conic linear problems*, in Semi-Infinite Programming: Recent Advances, M. Goberna and M. López, eds., Springer, Boston, MA, 2001, pp. 135–165.
[38] A. Shapiro, *Distributionally robust stochastic programming*, SIAM J. Optim., 27 (2017), pp. 2258–2275.
[39] A. Shapiro, *Tutorial on risk neutral, distributionally robust and risk averse multistage stochastic programming*, (2018), http://www.optimization-online.org/DB_HTML/2018/02/6455.html.
[40] F. Wahid, *River Optimization: Short-Term Hydro-Bidding under Uncertainty*, PhD thesis, The University of Auckland, Auckland, New Zealand, 2017.
[41] Z. Wang, P. Glynn, and Y. Ye, *Likelihood robust optimization for data-driven problems*, Comput. Manag. Sci., 13 (2016), pp. 241–261.
[42] W. Wiesemann, D. Kuhn, and M. Sim, *Distributionally robust convex optimization*, Oper. Res., 62 (2014), pp. 1358–1376.
[43] C. Zhao, *Data-Driven Risk-Averse Stochastic Program and Renewable Energy Integration*, PhD thesis, University of Florida, Gainesville, FL, 2014.
[44] C. Zhao and Y. Guan, *Data-driven risk-averse stochastic optimization with Wasserstein metric*, Oper. Res. Lett., 45 (2018), pp. 262–267.