REGULARIZED DECOMPOSITION OF HIGH–DIMENSIONAL MULTISTAGE STOCHASTIC PROGRAMS WITH MARKOV UNCERTAINTY

TSVETAN ASAMOV * AND WARREN B. POWELL†

Abstract. We develop a quadratic regularization approach for the solution of high–dimensional multistage stochastic optimization problems characterized by a potentially large number of time periods/stages (e.g. hundreds), a high-dimensional resource state variable, and a Markov information process. The resulting algorithms are shown to converge to an optimal policy after a finite number of iterations under mild technical assumptions. Computational experiments are conducted using the setting of optimizing energy storage over a large transmission grid, which motivates both the spatial and temporal dimensions of our problem. Our numerical results indicate that the proposed methods exhibit significantly faster convergence than their classical counterparts, with greater gains observed for higher–dimensional problems.

Key words. multistage stochastic optimization, quadratic regularization, nested decomposition, stochastic dual dynamic programming

AMS subject classifications. Programming: Stochastic, Dynamic programming/optimal control: Markov models

1. Introduction.

Multistage stochastic problems arise in a wide variety of real-world applications in fields as diverse as energy, finance, transportation and others. In this paper we consider multistage stochastic linear programs that satisfy the following conditions: *i*) the time horizon length T is finite but potentially large (there may be hundreds of time periods and stages); *ii*) for each time period, the set of sample realizations of the exogenous information process is finite (and relatively small); *iii*) for each stage, the stage cost is a linear function of the decision.

Pereira and Pinto [19] introduced a powerful algorithmic strategy known as Stochastic Dual Dynamic Programming (SDDP) that has received considerable attention for this problem class. Despite its popularity, SDDP can exhibit slow convergence, especially in the setting of high–dimensional resource allocation problems. A separate but important challenge arises when handling problems with long horizons which introduces algorithmic issues for both the setting of intertemporal independence, as well as when there is Markov dependence. Not surprisingly, as practical problems grow in size, improving the rate of convergence of SDDP–type methods becomes an issue of growing importance.

Quadratic regularization has been among the most effective techniques for accelerating the convergence of scenario tree–based decomposition methods (see work by Ruszczyǹski [26, 27, 29]). However, its application to the SDDP framework has not been possible because of the exponential growth of the number of required incumbent solutions. In this work, we propose a new regularization approach which overcomes that challenge. The method can lead to much faster convergence by reducing the oscillation of solutions around distant vertices of the feasible regions where the addition of new cutting hyperplanes might be of little value.

This paper makes the following contributions: *i*) We adopt notation that bridges the gap between dynamic programming and classical stochastic programming, which lays the foundation of our algorithmic strategy by identifying and clarifying the role of the post–decision information state; *ii*) We develop the first quadratic regularization method for the SDDP framework, with or without Markov dependence in the information process, that produces an optimal policy for a sampled model; *iii*) Unlike existing regularization methods on scenario trees, our approach remains computationally tractable even for problems that involve

*Department of Operations Research and Financial Engieering, Princeton University (tasamov@princeton.edu).
† Department of Operations Research and Financial Engieering, Princeton University (powell@princeton.edu).

long time horizons; *iv*) Our numerical results indicate that the proposed approach exhibits faster convergence than classical SDDP and is especially useful for problems with high–dimensional resource states. That makes the work relevant to a wide variety of practical applications.

Our numerical work uses the setting of optimizing energy over a fleet of storage devices for a congested transmission grid. This problem class offers a realistic setting for testing the algorithm with anywhere from 50 to 500 batteries, allowing us to test the performance of the algorithm for resource state variables with widely varying dimensionality. A separate challenge is that these problems exhibit a large number of time periods; our experiments model a day in 5–minute increments, producing problems with 288 time periods.

2. Literature Review. The decomposition approach of Benders [3] and the L–shaped method of Van Slyke and Wets [35] originally focused on the solution of two–stage stochastic optimization problems. Eventually, the idea was extended to the multi–period setting by Birge [5], as well as Donohue and Birge [8] who considered successive Benders–type approximations of the recourse functions in the nested Benders decomposition algorithm for multistage problems on scenario trees. Pereira and Pinto [19] further extended the approach to develop Stochastic Dual Dynamic Programming which has become popular among practitioners. On one hand, the method provides both lower and upper bounds, as well as clear convergence guarantees for many of its different versions as has been discussed Shapiro [32], as well as Linowsky and Philpott [14]. Moreover, it is also very suitable for parallel computing and can be applied to problems with long time horizons. Despite its progress towards overcoming the curse of dimensionality, in its essence SDDP is a cutting plane method, a class of algorithms known to exhibit slow convergence (see [30]), a behavior that is a byproduct of the well–known curse of dimensionality. In general, their computational complexity grows exponentially with the dimension of the problem. In the special case of only two time periods, the SDDP algorithm is equivalent to the well known cutting plane method of Kelley [12] which takes $O\left(\frac{\ln \epsilon^{-1}}{2 \ln 2}\left[\frac{2}{\sqrt{3}}\right]^{n-1}\right)$ iterations to achieve an ϵ–optimal solution on an n–dimensional problem as pointed out by Nesterov and Nesterov [18].

Rockafellar [24] introduced the proximal point algorithm for the minimization of (deterministic) lower semicontinuous proper convex functions. The quadratic regularization of two–stage linear stochastic optimization problems was developed by Ruszczyński [25, 29]. The same idea has also been implemented in the two–stage and multistage versions of the Stochastic Decomposition method developed by Higle and Sen [10, 31], as well as the decomposition work of Morton [17]. All of these methods utilize a scenario tree, either explicitly or implicitly (when indexing regularization terms by the entire history H_t), and consider separate incumbent solutions for every parent node in the scenario tree. That limits their applicability to problems with short time horizons. On the other hand, the method described below is universal and can be applied to problems with a large number of time periods.

3. Problem Formulation. Given a probability space $(\Omega, \mathcal{F}, P)$ with a sigma–algebra $\mathcal{F}$, and a filtration $\{\emptyset, \Omega\} = \mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots \subset \mathcal{F}_T = \mathcal{F}$, we consider a stochastic process $\{W_t,\ t = 1, \ldots, T\}$ adapted to $\{\mathcal{F}_t,\ t = 1, \ldots, T\}$. Throughout our presentation, we adopt the convention that any variable indexed by t is $\mathcal{F}_t$–measurable (surprisingly, this is not a standard assumption). Our goal is to develop new solution methods for the following multistage linear stochastic programming problem:

$$\min_{\substack{A_0 x_0 = b_0 \\ x_0 \geq 0}} \langle c_0, x_0 \rangle + \mathbb{E}_1 \left[\min_{\substack{B_0 x_0 + A_1 x_1 = b_1 \\ x_1 \geq 0}} \langle c_1, x_1 \rangle + \mathbb{E}_2 \left[\cdots + \mathbb{E}_T \left[\min_{\substack{B_{T-1} x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} \langle c_T, x_T \rangle \right] \cdots \right] \right]. \tag{1}$$

The components of the information process $W_t = (A_t, B_t, b_t, c_t), t = 1, \ldots, T$ are the $\mathcal{F}_t$–measurable random matrices A_t, B_t and vectors b_t, c_t, while A_0, B_0, b_0, c_0 are assumed to be deterministic components of the initial state of the system $S_0 = (A_0, B_0, b_0, c_0)$. We denote the sets of possible realizations of W_t with $\Omega_t,\ t = 1, \ldots, T$. Those correspond to nested partitions of Ω given by the filtration $\{\mathcal{F}_t,\ t = 1, \ldots, T\}$, and each $w \in \Omega$ can be represented as $\omega = (\omega_1, \omega_2, \ldots, \omega_T) \in \Omega_1 \times \Omega_2 \times \cdots \times \Omega_T$. We assume that each sample set Ω_t has a finite number of elements that is small enough to be enumerated computationally.

DEFINITION 1. *The information history at time t is $H_t = \{S_0, \omega_1, \omega_2, \ldots, \omega_t\}$, where $H_t \in \mathcal{H}_t = \{S_0\} \times \Omega_1 \times \Omega_2 \times, \cdots \times \Omega_t$. Further, we define the post–decision information history at time t to be $H_t^x = \{S_0, x_0, \omega_1, x_1, \omega_2, x_2, \ldots, \omega_t, x_t\}$.*

Employing a dynamic programming framework, we distinguish between two types of states of the system, the pre–decision states S_t and the post–decision states S_t^x.

DEFINITION 2. *The (pre–decision) state S_t of the system at time $t \geq 1$ is all the information in $H_{t-1}^x \cup \omega_t$ that is necessary and sufficient to make a decision at time t, and model the impact of $H_{t-1}^x \cup \omega_t$ on the computation of costs, constraints and transitions from time t onward.*

Furthermore, the pre–decision state of the system S_t can be represented as $S_t = (R_t, I_t)$, where the pre–decision resource state R_t is the amount of resources available at the beginning of time period t, and I_t is the pre–decision information state. Please note that R_t depends on both the decision x_{t-1} and the random vector b_t,

$$R_t = B_{t-1}x_{t-1} - b_t.$$

The information state I_t contains all the remaining information of S_t that is necessary and sufficient to model the system but is not in R_t. Formally, we consider the following model for the evolution of the system over time:

$$S_0 \xrightarrow{x_0} S_0^x \xrightarrow{\omega_1} S_1 \xrightarrow{x_1} S_1^x \xrightarrow{\omega_2} \ldots \xrightarrow{\omega_T} S_T \xrightarrow{x_T} S_T^x.$$

DEFINITION 3. *The post–decision state $S_t^x, t \geq 0$ of the system at time t is all the information in the post–decision history H_t^x that is necessary and sufficient to model the impact of H_t^x on the computation of costs, constraints and transitions from time t onward, after a decision has been made.*

We also represent the post–decision state of the system as $S_t^x = (R_t^x, I_t^x)$. The *post–decision resource state* R_t^x is given by

$$R_t^x = B_t x_t,$$

and the *post–decision information state* I_t^x represents all the information in S_t^x that is not in R_t^x. Moreover, we refer to the rank of the matrix B_t as the *dimension of the post–decision resource state*. If we define

$$C(S_t, x_t) := \langle c_t, x_t \rangle$$

and the set $\mathcal{X}_t(S_t)$ is such that the following conditions are satisfied,

$$\mathcal{X}_t(S_t) := \begin{cases} x_t \in \mathbb{R}^{n_t} \ : \ A_t x_t = b_t, & \text{if } t = 0 \\ x_t \in \mathbb{R}^{n_t} \ : \ B_{t-1}x_{t-1} + A_t x_t = b_t, & \text{if } t > 0 \end{cases}$$

then we can rewrite problem (1) using dynamic programming notation as follows,

$$\min_{x_0 \in \mathcal{X}_0(S_0)} C(S_0, x_0) + \mathbb{E}_1 \left[\min_{x_1 \in \mathcal{X}_1(S_1)} C(S_1, x_1) + \mathbb{E}_2 \left[\cdots + \mathbb{E}_T \left[\min_{x_T \in \mathcal{X}_T(S_T)} C(S_T, x_T) \right] \ldots \right] \right]. \tag{2}$$

Since the problem is stochastic, its optimal solution is not a vector but rather a policy π, which is a function that maps states S_t to decisions $x_t \in \mathcal{X}_t(S_t)$. Thus, we can consider the optimization problem (2) to be a search for an optimal policy π^* over the set Π consisting of all feasible and implementable policies

$$\min_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{T} C(S_t, X_t^{\pi}(S_t))\right]. \tag{3}$$

We refer to equation (3) as the *base model*, and we can solve it by constructing an optimal lookahead policy. In that case, the optimal decisions $X_t^*(S_t)$ corresponding to π^* satisfy the following optimality equation:

$$X_t^*(S_t) \in \underset{x_t \in \mathcal{X}_t(S_t)}{\arg\min} \left(C(S_t, x_t) + \min_{\pi \in \Pi} \mathbb{E}\left\{ \sum_{t'=t+1}^{T} C(S_{t'}, X_{t'}^{\pi}(S_{t'})) \middle| S_t^x \right\} \right). \tag{4}$$

Therefore, we can also specify an optimal lookahead policy π^* by employing its corresponding post–decision value functions $V_t^*(S_t^x)$,

$$V_t^*(S_t^x) = \min_{\pi \in \Pi} \mathbb{E}\left\{ \sum_{t'=t+1}^{T} C(S_{t'}, X_{t'}^{\pi}(S_{t'})) \middle| S_t^x \right\} \tag{5}$$

REMARK 1. *It is common for practitioners to employ a scenario tree in order to construct an approximate lookahead policy for problem (2) as follows:*

$$X_t^*(S_t) \in \underset{x_t \in \mathcal{X}_t(S_t)}{\arg\min} \left(C(S_t, x_t) + \min_{\pi \in \Pi} \mathbb{E}\left\{ \sum_{t'=t+1}^{t''} C(S_{t'}, X_{t'}^{\pi}(S_{t'})) \middle| S_t^x \right\} \right). \tag{6}$$

When $t'' < T$ the optimality of the approximate lookahead policy given by (6) cannot be established. In addition, lower and upper bounds to the optimal value of problem (2) might not be readily available (due to approximation errors stemming from the stage reduction). Nonetheless, lookahead models can produce high-quality solutions in selected problems (see [9]).

Thus, at any time period $t = 0, \ldots, T$, the optimal decision $X_t^*(S_t)$ for problem (2) can be computed as

$$X_t^*(S_t) \in \underset{x_t \in \mathcal{X}_t(S_t)}{\arg\min} \{C(S_t, x_t) + V_t^*(S_t^x)\}.$$

Hence, the search for an optimal policy π^* is equivalent to the computation of optimal post–decision value functions $V_t^*(S_t^x), t = 0, \ldots, T$. One of the well–known methods that allows us to construct such value functions is Stochastic Dual Dynamic Programming.

4. Stochastic Dual Dynamic Programming. Typically, stochastic programming techniques model the flow of information by utilizing a scenario tree that involves the entire set $\mathcal{H}_t = \{S_0\} \times \Omega_1 \times \cdots \times \Omega_t$. While such an approach is useful for analytical purposes, its practical applicability is limited by the exponential growth of the number of nodes in the scenario tree when the length of the time horizon T increases. To overcome that challenge, Pereira and Pinto [19] introduced the Stochastic Dual Dynamic Programming (SDDP) method for the solution of multistage stochastic linear optimization problems over long time horizons. SDDP overcomes the combinatorial explosion of the information state by exploiting (a key

and limiting assumption of) stagewise independence as $\mathbb{P}(\omega_{t+1}|H_t) = \mathbb{P}(\omega_{t+1})$, and therefore all post–decision states S_t^x share a single information state I_t^x. Hence, $S_t^x = R_t^x$ and the optimal value functions $V_t^*(S_t^x)$ only depend on the post–decision resource states R_t^x,

$$V_t^*(S_t^x) = V_t^*(R_t^x),\ t = 0, \dots, T.$$

The convexity of the optimal value functions $V_t^*(R_t^x)$ is the key property that allows one to partially escape the curse of dimensionality arising from partitioning the resource space. Instead, $V_t^*(R_t^x)$ can be approximated with lower–bounding convex outer approximations $\overline{V}_t^k(R_t^x)$ whose functional form is the maximum over a collection of affine functions. Those are commonly known as cutting hyperplanes or *Benders cuts*, and are constructed at the resource points $R_t^{x,j}$ that are visited during the j–th forward pass,

$$\overline{V}_t^k(R_t^x) := \max_{j \le k} \left\{ \alpha_t^j + \langle \beta_t^j, R_t^x - R_t^{x,j} \rangle \right\}. \tag{7}$$

For example, at iteration k we would obtain $R_t^{x,k}$ by solving the following linear program

$$x_t^k \in \underset{x_t \in \mathcal{X}_t(S_t)}{\arg\min} \left\{ C(S_t, x_t) + \overline{V}_t^{k-1}(R_t^x) \right\}, \text{ and setting } R_t^{x,k} \leftarrow B_t^k x_t^k \tag{8}$$

where $\overline{V}_t^0(R_t^x) = 0$.

The approximations $\overline{V}_t^{k-1}(R_t^x)$ are updated in the backward pass of iteration k by constructing a cutting hyperplane $h_t^k(R_t^x)$ to the optimal value function $V_t^*(R_t^x)$. To accomplish this, we use a lower bound $\underline{V}_{t+1}^k(R_t^{x,k})$ (derived from solutions to subproblems for time $t+1$) to $V_t^*(R_t^{x,k})$,

$$h_t^k(R_t^x) := \underline{V}_{t+1}^k(R_t^{x,k}) + \langle \beta_t^k, R_t^x - R_t^{x,k} \rangle. \tag{9}$$

Please note that the hyperplane $h_t^k(R_t^x)$ is not necessarily tangent to $V_t^*(R_t^x)$ since $\underline{V}_{t+1}^k(R_t^{x,k})$ might be strictly smaller than $V_t^*(R_t^{x,k})$.

REMARK 2. *When we need to emphasize the dependence of the feasible set $\mathcal{X}_{t+1}(S_{t+1})$ on the previous post–decision state R_t^x, we use the notation $\mathcal{X}_{t+1}(R_t^x, I_{t+1})$, where the exogenous information in R_{t+1} that is not contained in R_t^x is assumed to be contained in I_{t+1}.*

In order to construct $\underline{\mathbf{V}}_{t+1}^k$, we consider every element of the sample set $\omega_{t+1} \in \Omega_{t+1}$ and denote with $\underline{V}_{t+1}^k(R_t^x, \omega_{t+1})$ the optimal value of the following optimization problem,

$$\underline{V}_{t+1}^k(R_t^x, \omega_{t+1}) := \min_{x_{t+1} \in \mathcal{X}_{t+1}(R_t^x, I_{t+1}(\omega_{t+1}))} \left\{ C(S_{t+1}(\omega_{t+1}), x_{t+1}) + \overline{V}_{t+1}^k(R_{t+1}^x) \right\}.$$

Finally, we set

$$\underline{V}_{t+1}^k(R_t^x) := \sum_{\omega_{t+1} \in \Omega_{t+1}} \mathbb{P}(\omega_{t+1}) \underline{V}_{t+1}^k(R_t^x, \omega_{t+1}).$$

Hence, if we choose

$$\beta_t^k \in \partial_R \underline{V}_{t+1}^k(R_t^{x,k}),$$

then we can construct a new aggregated cut $h_t^k(R_t^x)$ as described in equation (9). Thus, in the backward pass of iteration k, we can update the approximate value function $\overline{V}_t^k(R_t^x)$ as follows,

$$\overline{V}_t^k(R_t^x) := \max \left\{ \overline{V}_t^{k-1}(R_t^x),\ h_t^k(R_t^x) \right\}.$$

If none of the constructed cuts are removed, then the growing collections of affine functions generate sequences of *monotonically increasing lower bounding approximations* $\overline{V}_t^k(R_t^x)$ to the optimal post–decision value functions $V_t^*(R_t^x)$ for any $t = 0, \ldots, T-1$.

$$\overline{V}_t^{k-1}(R_t^x) \leq \overline{V}_t^k(R_t^x) \leq V_t^*(R_t^x), \forall k \in \mathbb{N}, t = 0, \ldots, T-1.$$

Furthermore, in this work we assume relatively complete recourse, i.e. for any feasible solutions to the optimization problems at time periods $t = 0, \ldots, T-1$, there exists a feasible solution to any realized stage $t+1$ subproblem with probability one. This assumption alleviates the need for feasibility cuts and allows us to improve the clarity of the presentation.

5. Quadratic Regularization. Existing regularization approaches [29, 25, 10, 31, 17] utilize a scenario tree and consider separate incumbent solutions $\bar{x}_t(H_t)$ for every possible history $H_t \in \mathcal{H}_t, t = 0, \ldots, T-1$. The underlying idea in such methods has been to augment optimization problems of the form (8) with a regularization term as follows,

$$x_t^k \in \underset{x_t \in \mathcal{X}_t(S_t)}{\arg\min} \left\{ C(S_t, x_t) + \overline{V}_t^{k-1}(R_t^x) + \frac{\rho}{2} ||x_t - \bar{x}_t(H_t)||_2^2 \right\}. \tag{10}$$

As the algorithm progresses, each incumbent solution $\bar{x}_t(H_t)$ is updated to a new optimal solution, if certain conditions are satisfied. While such an approach is feasible for problems on scenario trees with small T, it is not practical for non–trivial time horizons. The exponential growth of the scenario tree ensures that only a tiny fraction of all possible realizations $H_t \in \mathcal{H}_t,\ t = 0, \ldots, T-1$ could be examined in the forward pass in a reasonable computational time. Moreover, multiple visits to each $H_t \in \mathcal{H}_t,\ t = 0, \ldots, T-1$ and multiple updates of its incumbent solution are also out of the realm of computational feasibility for most practical instances. One way to remedy this difficulty would be for different histories to share incumbent solutions. For example, a single incumbent solution $\bar{x}_t$ can be shared among all realizations $H_t \in \mathcal{H}_t$ and that would result in the optimization problem

$$x_t^k \in \underset{x_t \in \mathcal{X}_t(S_t)}{\arg\min} \left\{ C(S_t, x_t) + \overline{V}_t^{k-1}(R_t^x) + \frac{\rho}{2} ||x_t - \bar{x}_t||_2^2 \right\}. \tag{11}$$

Equation (11) can be used in place of equation (8), and it would still result in a convergent method for a fixed set of incumbent solutions $\bar{x}_t, t = 0, \ldots, T-1$. However, the optimality of the resulting policy cannot be established. Moreover, since the purpose of the quadratic regularization term is to mitigate the inaccuracy of the value function approximations, we do not need to regularize around the entire vector x_t (which might be very high–dimensional) but only around the parameters R_t^x of the post–decision value function approximations $\overline{V}_t^{k-1}(R_t^x)$. Thus, we can adjust problem (11) to address these concerns by making the following adjustments,

$$x_t^k \in \underset{x_t \in \mathcal{X}_t(S_t)}{\arg\min} \left\{ C(S_t, x_t) + \overline{V}_t^{k-1}(R_t^x) + \frac{\varrho^k}{2} \left\langle R_t^x - \overline{R}_t^{x,k-1}, Q_t (R_t^x - \overline{R}_t^{x,k-1}) \right\rangle \right\} \tag{12}$$

where the sequence of penalty coefficients $\{\varrho^k\}$ is such that $\varrho^k \geq 0,\ \forall k \in \mathbb{N}$ and $\lim_{k \to \infty} \varrho^k = 0$. We also introduce a positive semi–definite matrix $Q_t \succeq 0$, which can be used to address any scaling concerns across different entries of the resource vectors R_t^x. Please note that the meaning of the proposed regularization strategy is quite different from its scenario tree counterparts, as it aims to steer the solution towards a "known" region of the value function domain, rather than to the "correct" solution for the given history H_t of the stochastic

process. Hence, we choose the incumbent solutions to be the previous points encountered in the forward pass since the cuts supported at those points are the ones generated with the most information. Finally, we also point out that unlike the case of scenario trees, in the current method we do not aim for the convergence of the incumbent solutions towards any point. Interested readers are free to choose different incumbent solutions that they might find appropriate, and the convergence results presented below would still hold.

Now, we can substitute equation (12) for equation (8) in the forward pass of SDDP, and the new procedure would still converge to an optimal solution of problem (2) with probability one after a finite number of iterations. That might appear somewhat surprising since gradient methods applied to quadratic optimization problems typically entail asymptotic convergence. However, in this case a finite number of iterations is sufficient since the true problem remains linear, and the quadratic terms are only used to guide the exploration phase of the forward pass. Moreover, the generation of the supporting hyperplanes in the backward pass utilizes linear programming problems which can generate only a finite number of different cuts when basic dual feasible solutions are used. The details of the resulting method are presented in Algorithm 1, and we study its convergence properties below.

Algorithm 1 Quadratic Regularization Method with Stagewise Independence

Choose $Q_t \succeq 0, t = 0, \dots, T$, and define sequence $\{\varrho^k\}$.
Define $\overline{V}_T^k(R_T^x) := V_T^*(R_T^x), k = 0, \dots, K$.
Define $\overline{V}_t^0(R_t^x) := -\infty, t = 0, \dots, T-1$.
$(R_{-1}^{x,k}, I_0) \leftarrow S_0,\ k = 0, \dots, K$
for $k = 0, \dots, K$ **do**
  Forward Pass:
  Sample $\omega \in \Omega$.
  for $t = 0, \dots, T$ **do**
    if $(k = 0)$ **then**
      Select $x_t^k \in \arg\min_{x_t \in \mathcal{X}_t(R_{t-1}^{x,k}, I_t(\omega))} \{C(S_t(\omega), x_t)\}$
    else
      if $t < T$ **then**
        $x_t^k \in \arg\min_{x_t \in \mathcal{X}_t(R_{t-1}^{x,k}, I_t(\omega))} \left\{C(S_t(\omega), x_t) + \overline{V}_t^{k-1}(R_t^x) + \varrho^k \left\langle R_t^x - \overline{R}_t^{x,k-1}, Q_t(R_t^x - \overline{R}_t^{x,k-1})\right\rangle\right\}$
      else
        Select $x_t^k \in \arg\min_{x_t \in \mathcal{X}_t(R_{t-1}^{x,k}, I_t(\omega))} \left\{C(S_t(\omega), x_t) + \overline{V}_t^{k-1}(R_t^x)\right\}$
      end if
    end if
    Set $R_t^{x,k} \leftarrow B_t^k x_t^k$; $S_{t+1}(\omega) \leftarrow (R_t^{x,k} - b_{t+1}(\omega), I_{t+1}(\omega))$
  end for
  Backward Pass:
  for $t = T, \dots, 1$ **do**
    Define $\underline{V}_t^k(R_{t-1}^x, \omega_t) := \min_{x_t \in \mathcal{X}_t(R_{t-1}^x, I_t(\omega_t))} \left\{C(S_t(\omega_t), x_t) + \overline{V}_t^k(R_t^x)\right\}$
    for all $\omega_t \in \Omega_t$ **do**
      Select $\underline{\beta}_t^k(\omega_t) \in \partial_{R_{t-1}^x} \underline{V}_t^k(R_{t-1}^{x,k}, \omega_t)$
    end for
    $\alpha_{t-1}^k \leftarrow \sum_{\omega_t \in \Omega_t} \mathbb{P}(\omega_t) \underline{V}_t^k(R_t^{x,k}, \omega_t)$; $\beta_{t-1}^k \leftarrow \sum_{\omega_t \in \Omega_t} \mathbb{P}(\omega_t) \underline{\beta}_t^k(\omega_t)$
    $h_{t-1}^k(R_{t-1}^x) := \alpha_{t-1}^k + \langle \beta_{t-1}^k, R_{t-1}^x - R_{t-1}^{x,k} \rangle$
    $\overline{V}_{t-1}^k(R_{t-1}^x) := \max\left\{\overline{V}_{t-1}^{k-1}(R_{t-1}^x),\ h_{t-1}^k(R_{t-1}^x)\right\}$
  end for
  $\underline{V}_0^k \leftarrow \left\{\min_{x_0 \in \mathcal{X}_0(S_0)}\ C(S_0, x_0) + \overline{V}_0^k(R_0^x)\right\}$
  $\overline{R}_t^{x,k} \leftarrow R_t^{x,k},\ t = 0, \dots, T-1$
end for

LEMMA 4 ([21, 32]). *Suppose that dual basic solutions are used in the solution of subproblems in the backward pass of Algorithm 1. Then, there exist a finite number of possible value function approximations* $\overline{V}_t(\cdot), t = 0, \dots, T$.

Since the regularization terms are artificial for the original problem, we exclude them from the definition of an optimal policy.

DEFINITION 5. *The value function approximations* $\overline{V}_t^k, t = 0, \dots, T$ *are optimal for problem (2) if for any realization* $\omega \in \Omega$,

$$\min_{x_t \in \mathcal{X}_t(S_t(\omega))} \left\{ C(S_t(\omega), x_t) + \overline{V}_t^k(R_t^x) \right\} = \min_{x_t \in \mathcal{X}_t(S_t(\omega))} \left\{ C(S_t(\omega), x_t) + V_t^*(R_t^x) \right\} \tag{13}$$

for $t = 0, \dots, T$.

THEOREM 6. *Suppose that Algorithm 1 satisfies the following assumptions:*

1. $\overline{V}_T^k(\cdot) \equiv V_T^*(\cdot),\ k \in \mathbb{N}$.
2. *Dual basic optimal solutions are used in the backward pass.*
3. *Every element* $\omega \in \Omega$ *has a strictly positive probability* $\mathbb{P}(\omega) > 0$.
4. $\varrho^k \geq 0$ *and* $\lim_{k \to \infty} \varrho^k = 0$.
5. *The feasible sets* $\mathcal{X}_t(S_t)$ *are bounded for each* $t = 0, \dots, T$.

Then, the regularization method presented in Algorithm 1 converges to an optimal policy of problem (2) after a finite number of iterations with probability one.

Proof. Proof: Let $\overline{\mathcal{V}}_t$ denote the set of all possible value function approximations $\overline{V}_t^k,\ t = 0, \dots, T$ that can be generated by the backward pass of Algorithm 1. Since according to Assumption 2 we use only dual basic optimal solutions in the backward pass, by Lemma 4 we know that the sets $\overline{\mathcal{V}}_t$ have finite cardinality for all $t = 0, \dots, T$. Thus, we know that as the algorithm progresses all the value function approximations $\overline{V}_t^k$ will eventually stabilize. Therefore, there exists an iteration index $k_1 \in \mathbb{N}$ after which no updates can be made to the value functions $\overline{V}_t^k,\ t = 0, \dots, T$ for $k > k_1$. If the value functions $\overline{V}_t^{k_1},\ t = 0, \dots, T$ are optimal for problem (2), then we are done.

Now, suppose that was not the case. Then there exists $t',\ 0 \leq t' < T$, and $\omega' \in \Omega$ such that for any $k > k_1$ we have

$$\min_{x_{t'} \in \mathcal{X}_{t'}(S_{t'}(\omega'))} \left\{ C(S_{t'}(\omega'), x_{t'}) + V_{t'}^*(R_{t'}^x(\omega')) \right\} > \min_{x_{t'} \in \mathcal{X}_{t'}(S_{t'}(\omega'))} \left\{ C(S_{t'}(\omega'), x_{t'}) + \overline{V}_{t'}^{k-1}(R_{t'}^x(\omega')) \right\}$$

Let us consider the set

$$\begin{aligned} \Delta = \Big\{ & \delta \in \mathbb{R}, \\ & \delta = \min_{x_t \in \mathcal{X}_t(S_t(\omega))} \left\{ C(S_t(\omega), x_t) + V_t^*(R_t^x(\omega)) \right\} - \min_{x_t \in \mathcal{X}_t(S_t(\omega))} \left\{ C(S_t(\omega), x_t) + \overline{V}_t(R_t^x(\omega)) \right\} : \\ & \min_{x_t \in \mathcal{X}_t(S_t(\omega))} \left\{ C(S_t(\omega), x_t) + V_t^*(R_t^x(\omega)) \right\} > \min_{x_t \in \mathcal{X}_t(S_t(\omega))} \left\{ C(S_t(\omega), x_t) + \overline{V}_t(R_t^x(\omega)) \right\}, \\ & \text{where } \omega \in \Omega, \text{ and } \overline{V}_t \in \overline{\mathcal{V}}_t, t = 0, \dots, T \Big\} \end{aligned} \tag{14}$$

Since the number of elements $\omega \in \Omega$ is finite, we know that the set Δ also has a finite number of elements. Thus, Δ has a minimum element, and we denote

$$\epsilon = \min \Delta.$$

Hence,

$$\min_{x_{t'}\in\mathcal{X}_{t'}(S_{t'}(\omega'))}\left\{C(S_{t'}(\omega'),x_{t'})+V^*_{t'}(R^x_{t'}(\omega'))\right\}-\min_{x_{t'}\in\mathcal{X}_{t'}(S_{t'}(\omega'))}\left\{C(S_{t'}(\omega'),x_{t'})+\overline{V}^{k-1}_{t'}(R^x_{t'}(\omega'))\right\}\geq\epsilon$$

And if $t'>0$ we know that

$$V^*_{t'-1}(R^x_{t'-1})=\sum_{\omega_{t'}\in\Omega_{t'}}\mathbb{P}(\omega_{t'})\min_{x_{t'}\in\mathcal{X}(R^x_{t'-1},I_t(\omega_{t'}))}\left\{C(S_{t'}(\omega_{t'}),x_{t'})+V^*_{t'}(R^x_{t'})\right\}$$

and using convexity,

$$\overline{V}^{k-1}_{t'-1}(R^x_{t'-1})\leq\sum_{\omega_{t'}\in\Omega_{t'}}\mathbb{P}(\omega_{t'})\min_{x'_t\in\mathcal{X}(R^x_{t'-1},I_t(\omega_{t'}))}\left\{C(S_{t'}(\omega_{t'}),x_{t'})+\overline{V}^{k-1}_{t'}(R^x_{t'})\right\}.$$

Therefore,

$$\begin{aligned}&\min_{x_{t'-1}\in\mathcal{X}_{t'-1}(S_{t'-1}(\omega'))}\left\{C(S_{t'-1}(\omega'),x_{t'-1})+V^*_{t'-1}(R^x_{t'-1}(\omega'))\right\}\\&>\min_{x_{t'-1}\in\mathcal{X}_{t'-1}(S_{t'-1}(\omega'))}\left\{C(S_{t'-1}(\omega'),x_{t'-1})+\overline{V}^{k-1}_{t'-1}(R^x_{t'-1}(\omega'))\right\},\end{aligned}\tag{15}$$

which implies

$$\begin{aligned}&\min_{x_{t'-1}\in\mathcal{X}_{t'-1}(S_{t'-1}(\omega'))}\left\{C(S_{t'-1}(\omega'),x_{t'-1})+V^*_{t'-1}(R^x_{t'-1}(\omega'))\right\}\\&-\min_{x_{t'-1}\in\mathcal{X}_{t'-1}(S_{t'-1}(\omega'))}\left\{C(S_{t'-1}(\omega'),x_{t'-1})+\overline{V}^{k-1}_{t'-1}(R^x_{t'-1}(\omega'))\right\}\geq\epsilon.\end{aligned}\tag{16}$$

Proceeding by backward induction, we know that

$$\min_{x_0\in\mathcal{X}_0(S_0)}\left\{C(S_0,x_0)+V^*_0(R^x_0)\right\}-\min_{x_0\in\mathcal{X}_0(S_0)}\left\{C(S_0,x_0)+\overline{V}^{k-1}_0(R^x_0)\right\}\geq\epsilon.$$

Moreover, using Assumption 5 we know that R^x_t is bounded for each t. Hence, without loss of generality we can assume that k is such that

$$\varrho^k\langle R^x_t-\overline{R}^{x,k-1}_t,Q_t(R^x_t-\overline{R}^{x,k-1}_t)\rangle<\epsilon,\ \text{ for } t=0,\ldots,T-1.$$

Hence, if we denote with $\widetilde{x}^k_0$ the solution to the following regularized problem,

$$\widetilde{x}^k_0=\operatorname*{arg\,min}_{x_0\in\mathcal{X}_0(S_0)}\left\{C(S_0,x_0)+\overline{V}^{k-1}_0(R^x_0(\omega'))+\varrho^k\langle R^x_0(\omega')-\overline{R}^{x,k-1}_0,Q_0(R^x_0(\omega')-\overline{R}^{x,k-1}_0)\rangle\right\}$$

then we know that

$$\min_{x_0\in\mathcal{X}_0(S_0)}\left\{C(S_0,x_0)+V^*_0(R^x_0)\right\}>C(S_0,\widetilde{x}^k_0)+\overline{V}^{k-1}_0(R^{\widetilde{x},k}_0)+\varrho^k\langle R^{\widetilde{x},k}_0-\overline{R}^{x,k-1}_0,Q_0(R^{\widetilde{x},k}_0-\overline{R}^{x,k-1}_0)\rangle$$

And since Q is positive semi–definite, we know that

$$\min_{x_0\in\mathcal{X}_0(S_0)}\left\{C(S_0,x_0)+V^*_0(R^x_0)\right\}>C(S_0,\widetilde{x}^k_0)+\overline{V}^{k-1}_0(R^{\widetilde{x},k}_0)$$

which implies,

$$C(S_0,\widetilde{x}^k_0)+V^*_0(R^{\widetilde{x},k}_0)>C(S_0,\widetilde{x}^k_0)+\overline{V}^{k-1}_0(R^{\widetilde{x},k}_0).$$

and therefore

$$V_0^*(R_0^{\widetilde{x},k}) > \overline{V}_0^{k-1}(R_0^{\widetilde{x},k}).$$

Thus we know that the value function approximation $\overline{V}_0^{k-1}(\cdot)$ is suboptimal at the point $R_0^{\widetilde{x},k}$ corresponding to $\widetilde{x}_0^k$. Hence, if the value function $V_1^{k-1}(\cdot)$ is such that for each $\omega_1 \in \Omega_1$ the following holds,

$$\min_{x_1 \in \mathcal{X}(R_0^{\widetilde{x},k}, I_1(\omega_1))} \left\{ C(S_1(\omega_1), x_1) + \overline{V}_1^{k-1}(R_1^x) \right\} = \min_{x_1 \in \mathcal{X}(R_0^{\widetilde{x},k}, I_1(\omega_1))} \left\{ C(S_1(\omega_1), x_1) + V_1^*(R_1^x) \right\}$$

then the backward pass will result in an updated value function $\overline{V}_0^k(\cdot)$ such that $\overline{V}_0^k(R_0^{\widetilde{x},k}) = V_0^*(R_0^{\widetilde{x},k}) > \overline{V}_0^{k-1}(R_0^{\widetilde{x},k})$ which is a contradiction with the choice of k. Therefore, it must be the case that there exists $\omega_1'' \in \Omega_1$ such that

$$\min_{x_1 \in \mathcal{X}(R_0^{\widetilde{x},k}, I_1(\omega_1''))} \left\{ C(S_1(\omega_1''), x_1) + \overline{V}_1^{k-1}(R_1^x) \right\} < \min_{x_1 \in \mathcal{X}(R_0^{\widetilde{x},k}, I_1(\omega_1''))} \left\{ C(S_1(\omega_1''), x_1) + V_1^*(R_1^x) \right\}.$$

Moreover,

$$\min_{x_T \in \mathcal{X}(S_T(\omega))} \left\{ C(S_T(\omega), x_T) + \overline{V}_T^{k-1}(R_T^x) \right\} = \min_{x_T \in \mathcal{X}(S_T(\omega))} \left\{ C(S_T(\omega), x_T) + V_T^*(R_T^x) \right\}.$$

Therefore, there exists a sample path $\omega'' \in \Omega$ and a time index $t'',\ 0 < t'' < T$ such that the sequence of regularized solutions $\widetilde{x}_t^k(\omega'')$ would result in a suboptimal value function evaluation at t'',

$$\min_{x_{t''} \in \mathcal{X}(R_{t''-1}^{\widetilde{x},k}, I_{t''}(\omega''))} \left\{ C(S_{t''}(\omega''), x_{t''}) + \overline{V}_{t''}^{k-1}(R_{t''}^x) \right\} < \min_{x_{t''} \in \mathcal{X}(R_{t''-1}^{\widetilde{x},k}, I_{t''}(\omega''))} \left\{ C(S_{t''}(\omega), x_{t''}) + V_{t''}^*(R_{t''}^x) \right\},$$

and optimal evaluations at $t'' + 1$ for all possible $\omega_{t''+1} \in \Omega_{t''+1}$,

$$\begin{aligned} &\min_{x_{t''+1} \in \mathcal{X}(R_{t''}^{\widetilde{x},k}, I_{t''+1}(\omega_{t_{t''+1}}))} \left\{ C(S_{t''+1}(\omega_{t''+1}), x_{t''+1}) + \overline{V}_{t''+1}^{k-1}(R_{t''+1}^x) \right\} \\ = &\min_{x_{t''+1} \in \mathcal{X}(R_{t''}^{\widetilde{x},k}, I_{t''+1}(\omega_{t''+1}))} \left\{ C(S_{t''+1}(\omega_{t''+1}), x_{t''+1}) + V_{t''+1}^*(R_{t''+1}^x) \right\}. \end{aligned}$$

□

Hence the backward pass of iteration k will result in an updated value function approximation $\overline{V}_{t''}^k(\cdot)$ such that

$$\overline{V}_{t''}^k(R_{t''}^{\widetilde{x},k}) = V_{t''}^*(R_{t''}^{\widetilde{x},k}) > \overline{V}_{t''}^{k-1}(R_{t''}^{\widetilde{x},k}),$$

which is a contradiction with the choice of k. This completes the proof.

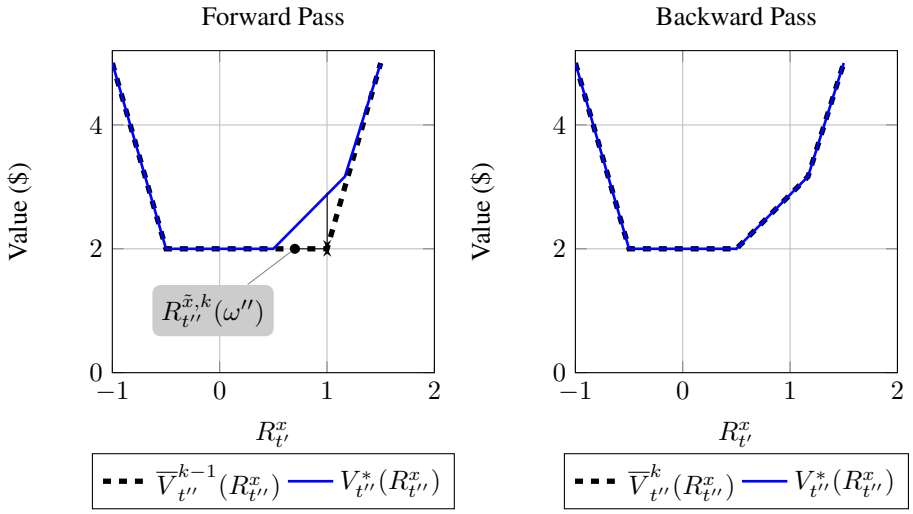

FIG. 1. *Value function update at iteration k.*

6. Beyond Stagewise Independence. Despite its advantages, the SDDP methodology has one crucial drawback. The stagewise independence of $W_t = (A_t, B_t, b_t, c_t)$ will generally not hold in practice since real–world multistage problems often involve stochastic processes that exhibit some degree of temporal dependence. There are different approaches that we can adopt to address this difficulty. First, let us consider the special case when the history dependence occurs only in the right hand side constraint vectors b_t, and it has the following autoregressive structure:

$$b_t = \sum_{t'=1}^{t-1} (\Phi_{t,t'} b_{t'} + \Psi_{t,t'} \eta_{t'}) + \eta_t \tag{17}$$

where the process (A_t, B_t, c_t, η_t) is stagewise independent and the deterministic matrices $\Phi_{t,t'}$ and $\Psi_{t,t'}$ contain the autoregressive information. Then, for each time period $t > 0$ in the SDDP formulation, we can extend the original optimization problem with additional variables to accommodate the realizations of $b_{t'}$ and $\eta^{t'}, t' < t$ that are necessary to model the autoregressive dependence (see [6, 15], and [34]). The advantage of such a solution to the history dependence problem is that stagewise independence is present in the extended formulation. A drawback of the approach is that the dimension of the state space also increases from $|R^x_t|$ (in the stagewise independence case) to possibly as much as $|R^x_t| + \sum_{t'=0}^{t-1}(|b_{t'}| + |\eta_{t'}|)$ (in the history dependent case), which implies a slower convergence rate (note that we can omit terms $|b_{t'}|$ if $\Phi_{t,t'} = \mathbf{0}$, and $|\eta_{t'}|$ if $\Psi_{t,t'} = \mathbf{0}$). This problem can be alleviated with the use of cut sharing strategies as described in [11], and [7].

In the remainder of this section we consider an alternative setup that leads to an increase in the information dimension rather than the resource dimension. We assume that the stochastic process W_t is a discrete state Markov chain. Thus, the probability of occurrence of $\omega_{t+1} \in \Omega_{t+1}$ depends only on the current realization $\omega_t \in \Omega_t$ or the current post–decision information state I^x_t,

$$\mathbb{P}(\omega_{t+1}|H_t) = \begin{cases} \mathbb{P}(\omega_{t+1}|S_t) = \mathbb{P}(\omega_{t+1}|I^x_t), & \text{if } t = 0 \\ \mathbb{P}(\omega_{t+1}|\omega_t) = \mathbb{P}(\omega_{t+1}|I^x_t), & \text{if } t > 0. \end{cases} \tag{18}$$

Such an approach can be suitable for problems where the process (A_t, B_t, c_t) is not stagewise independent, or the autoregressive model (17) does not constitute a good fit to the observed realizations of the random process. For example, historical weather data might indicate the presence of distinct patterns that cannot be explained with a normal error distribution around a given mean (which arise in autoregressive estimation). Alternatively, the relevant information state could be the forecast of the highest temperature tomorrow.

To properly model such weather dynamics one might need to consider different weather regimes that are inherently distinct. Thus, multiple approximations of the value functions need to be employed, which increases the size of the optimization problem. That leads to greater computational requirements for solving the problem as a distinct recourse function approximation needs to be constructed for every $I_t^x \in \mathcal{I}_t^x$, where $\mathcal{I}_t^x$ denotes the set of all possible post–decision information states at time t. Hence, we need to maintain $|\mathcal{I}_t^x(\Omega_t)|$ sets of cuts for each time period $t = 0, \ldots, T$, and therefore the approach is suitable for problems where the cardinality of the possible post–decision information states $|\mathcal{I}_t^x(\Omega_t)|$ is small, or alternatively when the cardinality of the sample sets $|\Omega_t|$ is small. However, unlike the case of an autoregressive fit (17), the dimension of the post–decision resource state is preserved in each set of cuts, and the corresponding exponential increase in the computational time is avoided.

In the forward pass at iteration k, we consider a sample path $\omega = (\omega_1, \ldots, \omega_T)$ that is generated using (18). At each time step $t = 0, \ldots, T-1$ the piecewise–linear value function $V_t^{k-1}(R_t^x, I_t^x(\omega))$ is used to approximate the optimal value function $V_t^*(R_t^x, I_t^x(\omega))$ at the current post–decision information state $I_t^x(\omega)$.

In the backward pass of the algorithm at iteration k, we consider $t = T, \ldots, 1$ and generate the cutting hyperplanes $h_{t-1}^k(R_{t-1}^x, I_{t-1}^x)$ for each $I_{t-1}^x \in \mathcal{I}_{t-1}^x$. Please note that if the random process W_t is a finite state Markov chain, then $|\mathcal{I}_t^x(\Omega_t)| \leq |\Omega_t|, t = 0, \ldots, T$. We employ the conditional probabilities $\mathbb{P}(\omega_t | I_{t-1}^x)$ to construct constant intercepts and slopes,

$$\alpha_{t-1}^k(I_{t-1}^x) \leftarrow \sum_{\omega_t \in \Omega_t} \mathbb{P}(\omega_t | I_{t-1}^x) \underline{V}_t^k(R_t^{x,k}, \omega_t)$$

and,

$$\beta_{t-1}^k(I_{t-1}^x) \leftarrow \sum_{\omega_t \in \Omega_t} \mathbb{P}(\omega_t | I_{t-1}^x) \underline{\beta}_t^k(\omega_t).$$

Thus,

$$h_{t-1}^k(R_{t-1}^x, I_{t-1}^x) := \alpha_{t-1}^k(I_{t-1}^x) + \langle \beta_{t-1}^k(I_{t-1}^x), R_{t-1}^x - R_{t-1}^{x,k} \rangle.$$

Hence, we can construct the new value function approximation $\overline{V}_{t-1}^k(R_{t-1}^x, I_{t-1}^x)$ for the post–decision information state I_{t-1}^x as,

$$\overline{V}_{t-1}^k(R_{t-1}^x, I_{t-1}^x) := \max \left\{ \overline{V}_{t-1}^{k-1}(R_{t-1}^x, I_{t-1}^x),\ h_{t-1}^k(R_{t-1}^x, I_{t-1}^x) \right\}. \tag{19}$$

The description of the method is given in Algorithm 2.

THEOREM 7. *Suppose that $\{W_t, t = 1, \ldots, T\}$ is a discrete Markov process as described by equation (18). If $\overline{V}_T^k(\cdot, I_T^x) \equiv V_T^*(\cdot, I_T^x)$, for $I_T^x \in \mathcal{I}_T^x$, $k = 0, \ldots, K$, and conditions 2, 3, 4, and 5 specified in Theorem 6 are satisfied, then the method presented in Algorithm 2 converges to an optimal policy of problem (2) after a finite number of iterations with probability one.*

Proof. Proof: The proof is analogous to the proof of Theorem 6. The main difference is that for each time period $t = 1, \ldots, T$ we need to consider $|I_t^x|$ different value functions

Algorithm 2 Quadratic Regularization Method for Markov Models

Choose $Q_t \succeq 0, t = 0, \ldots, T$, and define the sequence $\{\varrho^k\}$.
Define $\overline{V}_T^k(R_T^x, I_T^x) := V_T^*(R_T^x, I_T^x),\ k = 0, \ldots, K,\ I_T^x \in \mathcal{I}_T^x$.
Define $\overline{V}_t^0(R_t^x, I_t(\omega_t)) := -\infty,\ \omega_t \in \Omega_t,\ t = 0, \ldots, T-1$.
$(R_{-1}^{x,k}, I_0) \leftarrow S_0,\ k = 0, \ldots, K$
for $k = 0, \ldots, K$ **do**
  Sample $\omega \in \Omega$ using the Markov stochastic process $\{W_t, t = 1, \ldots, T\}$.
  for $t = 0, \ldots, T$ **do**
    if $(k = 0)$ **then**
      Select $x_t^k \in \arg\min_{x_t \in \mathcal{X}_t(R_{t-1}^{x,k}, I_t(\omega))} \{C(S_t(\omega), x_t)\}$
    else
      if $t < T$ **then**
        $x_t^k \in \arg\min_{x_t \in \mathcal{X}_t(R_{t-1}^{x,k}, I_t(\omega))} \left\{C(S_t(\omega), x_t) + \overline{V}_t^{k-1}(R_t^x, I_t^x(\omega)) + \varrho^k \left\langle R_t^x - \overline{R}_t^{x,k-1}, Q_t(R_t^x - \overline{R}_t^{x,k-1})\right\rangle\right\}$
      else
        $x_t^k \in \arg\min_{x_t \in \mathcal{X}_t(R_{t-1}^{x,k}, I_t(\omega))} \left\{C(S_t(\omega), x_t) + \overline{V}_t^{k-1}(R_t^x, I_t^x(\omega))\right\}$
      end if
    end if
    Set $R_t^{x,k} \leftarrow B_t^k x_t^k$; $S_{t+1}(\omega) \leftarrow (R_t^{x,k} - b_{t+1}(\omega), I_{t+1}(\omega))$
  end for
  for $t = T, \ldots, 1$ **do**
    Define $\underline{V}_t^k(R_{t-1}^x, \omega_t) := \min_{x_t \in \mathcal{X}_t(R_{t-1}^x, I_t(\omega_t))} \left\{C(S_t(\omega_t), x_t) + \overline{V}_t^k(R_t^x, I_t^x(\omega_t))\right\}$
    for all $\omega_t \in \Omega_t$ **do**
      Select $\underline{\beta}_t^k(\omega_t) \in \partial_{R_{t-1}^x} \underline{V}_t^k(R_{t-1}^{x,k}, \omega_t)$
    end for
    for all $I_{t-1}^x \in \mathcal{I}_{t-1}^x(\Omega_{t-1})$ **do**
      $\alpha_{t-1}^k(I_{t-1}^x) \leftarrow \sum_{\omega_t \in \Omega_t} \mathbb{P}(\omega_t | I_{t-1}^x) \underline{V}_t^k(R_t^{x,k}, \omega_t)$; $\beta_{t-1}^k(I_{t-1}^x) \leftarrow \sum_{\omega_t \in \Omega_t} \mathbb{P}(\omega_t | I_{t-1}^x) \underline{\beta}_t^k(\omega_t)$
      $h_{t-1}^k(R_{t-1}^x, I_{t-1}^x) := \alpha_{t-1}^k(I_{t-1}^x) + \langle \beta_{t-1}^k(I_{t-1}^x), R_{t-1}^x - R_{t-1}^{x,k} \rangle$
      $\overline{V}_{t-1}^k(R_{t-1}^x, I_{t-1}^x) := \max\left\{\overline{V}_{t-1}^{k-1}(R_{t-1}^x, I_{t-1}^x),\ h_{t-1}^k(R_{t-1}^x, I_{t-1}^x)\right\}$
    end for
  end for
  $\underline{V}_0^k \leftarrow \left\{\min_{x_0 \in \mathcal{X}_0(S_0)}\ C(S_0, x_0) + \overline{V}_0^k(R_0^x, I_0^x)\right\}$
  $\overline{R}_t^{x,k} \leftarrow R_t^{x,k},\ t = 0, \ldots, T-1$
end for

$\overline{V}_t^k(R_t^x, I_t^x)$. Since $|I_t^x|$ is finite, the argument of the proof of Theorem 6 can be extended to show that with probability 1, there exists a large enough $k \in \mathbb{N}$ such that the value functions $\overline{V}_t^k(R_t^{\hat{x},k}(\omega), I_t^x)$ are optimal for all $\omega \in \Omega$, and $I_t^x \in \mathcal{I}_t^x, t = 0, \ldots, T$. □

REMARK 3. *Various optimization methods for Markov models have been studied in the literature for both the risk–neutral (see [23, 22, 33]) and risk–averse cases (see [20, 28, 16] and the references within). An extensive treatment of optimization problems with Markov uncertainty is beyond the scope of this work. The goal of our presentation is the introduction of regularization into the Markovian setting, so that it can be adapted to other problems on a case–by–case basis.*

7. Algorithmic Tuning. In order to turn mathematical arguments into useful numerical results one needs to employ a high quality implementation and suitable parameter tuning. In this section we present some of the potential issues regarding the reliability and computational performance of the methods presented above. We consider the construction of regularization sequences, and discuss numerical concerns regarding the solutions of subproblems.

7.1. Regularization Coefficients. In general, we cannot find a regularization sequence that would lead to the fastest possible convergence. However, if we consider sequences that are defined by a set of parameters, then we can attempt to find suitable parameter values. For example, we can construct regularization sequences $\varrho^k \geq 0$ such that $\lim_{k\to\infty} \varrho^k = 0$ by using the following geometric sequence. Given $\varrho^0 > 0$ and $r \in (0, 1)$, we define

$$\varrho^k = \varrho^0 r^k = r \cdot \varrho^{k-1}, \text{ if } k > 0. \tag{20}$$

In this case, we need to tune the parameters ϱ^0 and r. We can gain insight by solving a small instance of the given problem for different pairs (ϱ^0, r). For example, in section 8 we describe an optimization model to be solved for high–dimensional post–decision resource states $|R_t^x| \geq 50$. As a pre–processing step, we can solve a smaller instance, e.g. $|R_t^x| = 25$, for each $(\varrho^0, r) \in \{1, 10, 100\} \times \{0.9, 0.95, 0.99\}$, and compare the results. Since estimates of the upper bounds and optimality gaps are stochastic, we prefer to compare only the deterministic lower bounds as they are more reliable. The resulting plots can be found in Figure 2. We can see that the sequences of regularization coefficients has an impact on the behavior of the proposed methods. However, various choices of (ϱ^0, r) can be used with similar success. In our experiments in section 8, we use $\varrho^0 = 1,\ r = 0.95$.

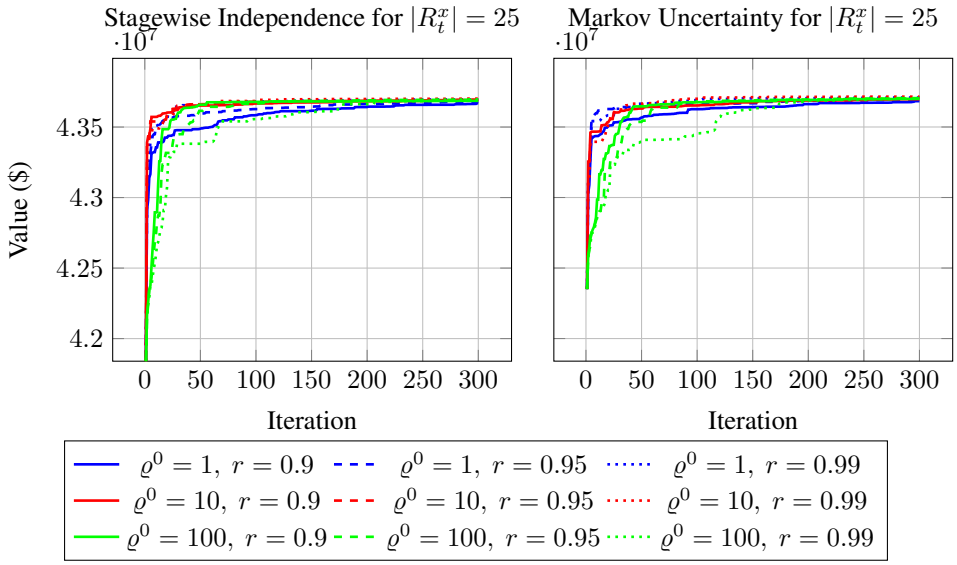

FIG. 2. *Lower bounds to the objective value of a stochastic optimization problem for different regularization sequences.*

7.2. Convergence Tolerance for the Solution of Subproblems. At each step of the forward and backward pass of Algorithm 1 and Algorithm 2, we use the current collection of hyperplanes $\left\{\alpha_t^j + \langle \beta_t^j, R_t^x - R_t^{x,j} \rangle,\ j \leq k\right\}$ and a realization of $W_t = (A_t, B_t, b_t, c_t)$ as a part of the input to a convex optimization problems having the following general form,

$$\begin{aligned} \min\ & \langle c, y \rangle + \frac{1}{2} \langle y, Qy \rangle \\ \text{s.t. } & Ay = b \\ & y \geq 0 \end{aligned} \tag{21}$$

The numerical precision of the solutions to subproblems (21) is essential for the correctness of the resulting policy for problem (1). However, the right–hand side vector b of problem (21) includes the vector b_t and the constant terms $\alpha_t^j - \beta_t^j R_t^{x,j}$ of the value function approximations given in (7) or (19). If problem (1) has a long time horizon, then an aggregation of constant terms with large modulus $|\alpha_t^j - \beta_t^j R_t^{x,j}|$ can occur, and that could lead to numerical solutions of problem (21) which do not satisfy the system of constraints $B_{t-1}x_{t-1} + A_t x_t = b_t$ with a desirable precision. Convex optimization tools, including specialized algorithms for linear and quadratic programming problems, often use convergence tolerance parameters to guide their stopping conditions. For problems with long time horizons, we encounter numerical precision problems that require that we use care in setting tolerance parameters for stopping conditions. In the sections below, we discuss the issues of relative primal feasibility, and the relative complementarity gap.

7.2.1. Relative primal feasibility. Suppose that a given optimization solver has a feasibility condition of the following form,

$$\frac{||Ay - b||}{1 + ||b||} \leq \varepsilon_f. \tag{22}$$

We can consider two right–hand side vectors b^1, b^2 such that $||b^1|| < ||b^2||$ and corresponding candidate solutions y^1, y^2 such that $\dfrac{||Ay^1 - b^1||}{1 + ||b^1||} = \dfrac{||Ay^2 - b^2||}{1 + ||b^2||} = \varepsilon_f$. Then the feasibility errors satisfy $||Ay^1 - b^1|| < ||Ay^2 - b^2||$. Therefore, if we keep the primal feasibility tolerance ε_f fixed while $||b||$ grows, then the feasibility errors $||Ay - b||$ (and therefore $||B_{t-1}x_{t-1} + A_t x_t - b_t||$) could increase as well. Hence, for problems with a long time horizon or a large number of hyperplanes in the value function approximation, one might need to decrease the tolerance ε_f for problem (21) in order to bring the size of the error $||B_{t-1}x_{t-1} + A_t x_t - b_t||$ down to an acceptable level.

7.2.2. Relative complementarity gap. Commercial solvers often include implementations of primal–dual interior point methods (see [36, 4]) that employ a relative complementarity tolerance ε_c in their stopping condition. The presence of large (by modulus) constant terms in the right–hand side vector b can also lead the numerical solver to terminate at an infeasible solution with non–negligible errors $||Ay - b||$ and $||B_{t-1}x_{t-1} + A_t x_t - b_t||$, if ε_c is not chosen appropriately. We present a brief explanation below.
The Lagrangian of problem (21) is given by

$$L(y, \mu, \lambda) = \langle c, y\rangle + \frac{1}{2}\langle y, Qy\rangle + \langle \mu, b - Ay\rangle - \langle \lambda, y\rangle. \tag{23}$$

Hence, the Karush–Kuhn–Tucker conditions for problem (21) are given by the system of constraints,

$$\begin{aligned} Ay &= b \\ A^\top \mu - Qy + \lambda &= c \\ Y\Lambda \mathbf{1} &= 0 \\ y, \lambda &\geq 0 \end{aligned} \tag{24}$$

where $Y = \text{diag}(y)$ and $\Lambda = \text{diag}(\lambda)$.

Interior point methods construct iterative approximations to the solution of (24) using a sequence of scalar barrier parameters $\nu^n > 0$, such that $\nu^n \downarrow 0$. Assuming that the initial

point (y^0, μ^0, λ^0) is infeasible for (24) and $\langle y^0, \lambda^0 \rangle > 0$, we can have a stopping condition for the complementarity gap such as

$$\frac{\langle y^n, \lambda^n \rangle}{\langle y^0, \lambda^0 \rangle} \leq \varepsilon_c \quad \text{or} \quad \nu^n \leq \varepsilon_c \quad \text{or} \quad \frac{\nu^n}{|\langle c, y^n \rangle + \langle y^n, Qy^n \rangle|} \leq \varepsilon_c. \tag{25}$$

At iteration n, the interior point method finds a Newton direction $(\Delta y, \Delta\mu, \Delta\lambda)$ for problem (24) as the solution to the following system :

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^\top & I \\ \Lambda & 0 & Y \end{bmatrix} \cdot \begin{bmatrix} \Delta y \\ \Delta\mu \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} b - Ay^n \\ c - A^\top \mu^n + Qy^n + \lambda^n \\ \nu^n \mathbf{1} - Y^n \Lambda^n \mathbf{1} \end{bmatrix} \tag{26}$$

where $I = \text{diag}(1, 1, \ldots, 1)$ denotes the identity matrix.
Then the current solution (y^n, μ^n, λ^n) can be updated by choosing $\gamma^n \in (0, 1]$ such that

$$(y^n, \lambda^n) + \gamma^n (\Delta y, \Delta\lambda) \geq 0 \tag{27}$$

and setting

$$(y^{n+1}, \mu^{n+1}, \lambda^{n+1}) = (y^n, \mu^n, \lambda^n) + \gamma^n (\Delta y, \Delta\mu, \Delta\lambda). \tag{28}$$

Please note that if $\gamma^n = 1$, then $y^{n+1} \geq 0$ would be feasible for problem (21) since $Ay^{n+1} = b$. However, in practice we usually have $\gamma^n < 1$. Hence, a complementarity tolerance condition (25) can be met even if the system $B_{t-1}x_{t-1} + A_t x_t = b_t$ is not satisfied within the desired precision. In order to address this concern, in our numerical experiments we set the tolerance ε_c to the smallest possible value allowed by the solver (10^{-12}).

8. Numerical Experiments. In this section we study the computational performance of the algorithms proposed above. We focus our analysis on the following questions.

- How is the computational performance of Algorithms 1 and 2 affected by:
  - the dimension of the resource vector R_t?
  - the size of the post–decision information state space $\mathcal{I}_t^x$?
- How does the performance of Algorithms 1 and 2 compare to their non–regularized counterparts?

Our experimental work was conducted using the setting of optimizing grid level storage for a large transmission grid managed by PJM Interconnection. PJM manages grid level storage devices from a single location, making it a natural setting for testing our algorithms. As of this writing, grid level storage is dropping in price, providing a meaningful setting to evaluate the performance of our algorithms for a wide range of storage devices, challenging the ability of the algorithms to handle high dimensional applications. For this reason, we conducted tests on networks with 50 to 500 storage devices. These are much higher dimensional problems than prior research that has focused on the management of water reservoirs.

Another distinguishing feature of our grid storage setting (compared to prior experimental work) is that a natural time step is 5 minutes, which is the frequency with which real–time electricity prices (known as LMPs, for locational marginal prices) are updated on the PJM grid. We anticipate using storage devices to hold energy over horizons of several hours. For this reason, we used a 24 hour model, divided into 5–minute increments, for 288 time periods, which is quite large compared to many applications using this algorithmic technology.

A complete description of the given model is beyond the scope of the current paper and can be found in [1]. Below we briefly describe the construction of the network, and the exogenous stochastic process. Finally we present the results of an extensive set of experiments investigating the effect of regularization, the number of storage devices (which determines the dimensionality of R_t), and the presence of an exogenous post–decision information state, on the rate of convergence and solution quality.

8.1. The network. We performed our experiments using an aggregated version of the PJM grid. Instead of the full network with 9,000 buses and 14,000 transmission lines, we limited our analysis to the higher voltage lines, producing a grid with 1,360 buses and 1,715 transmission lines. The power generators include 396 gas turbines (23,309 MW), 50 combined cycle generators (21,248 MW), 264 steam generators (73,374 MW), 31 nuclear reactors (31, 086 MW), and 84 conventional hydro power generators (2,217 MW). Off–shore wind power was simulated for a set of hypothetical wind turbines with a combined maximum capacity of 16 GW. Moreover, we consider a daily time horizon with 5–minute discretization resulting in a total of 288 time periods.

The data was prepared by first running a unit–commitment simulator called SMART–ISO that determines which generators are on or off at each point in time, given forecasts of wind generated from a planned set of off–shore wind farms. We made the assumption that the use of grid level storage would not change which generators are on or off at any point in time. However, we simultaneously optimize ramping the generators up or down within ranges, while charging and discharging of storage devices around the grid in the presence of stochastic injections from the wind farms.

We placed the distributed storage devices at the points–of–interconnection for wind farms, as well as the buses with the highest demand. Each storage device is characterized by its minimum and maximum energy capacity, its charging and discharging efficiency, and its variable storage cost. The control of multiple storage devices in a distributed energy system is a challenging task that depends on a variety of factors such as the location of each device, and the presence of transmission line congestion. A good storage algorithm needs to respond to daily variations in supply, demand and congestion, taking advantage of opportunities to store energy near generation points (to avoid congestion) or near load points (during off–peak periods). It has to balance when and where to store and discharge in a stochastic, time–dependent setting, providing a challenging test environment for our algorithm.

8.2. The exogenous information. Our only source of uncertainty (the exogenous information) was from the injected wind from the offshore wind farms. In order to calibrate our stochastic wind error model, we employed historical wind data and speed measurements of off–shore wind for the month of January 2010. For each time period, we consider a set of ten vectors of possible wind speed realizations which correspond to ten different weather regimes.

In general, the exogenous information process can be characterized by one of the following: stagewise independence, compact state variables (Markov processes), or scenario–dependence (path dependence). For some instances, the latter case could be reduced to one of the former two by applying an appropriate transformation as described in section 6. In our experiments, we consider instances with stagewise independent transitions between ten equally likely scenarios. When we assumed stagewise independence, we would sample from each of these 10 scenarios with equal probability at each time period. For the problems with Markov uncertainty, we assumed that at every time period t, the probability of continuing with the same weather regime at time $t+1$ is 91 percent. Additionally, each of the remaining nine regimes can be visited at time $t+1$ with a probability of 1 percent.

8.3. Algorithmic comparisons. The proposed algorithms were implemented in Java, and the IBM ILOG CPLEX 12.4 solver was used for the solution of linear and quadratic convex optimization problems. Further, we performed parameter tuning as described in section 7. We set the relative complementarity tolerance of CPLEX to 10^{-12}, and used a geometric regularization sequence with $\varrho^0 = 1$ and $r = 0.95$. Additionally, we run each method for $K = 300$ iterations. Moreover, the scaling matrices $Q_t, t = 0, \ldots, T$ are set to the identity matrix which implies that the amount of energy in each storage device has the same

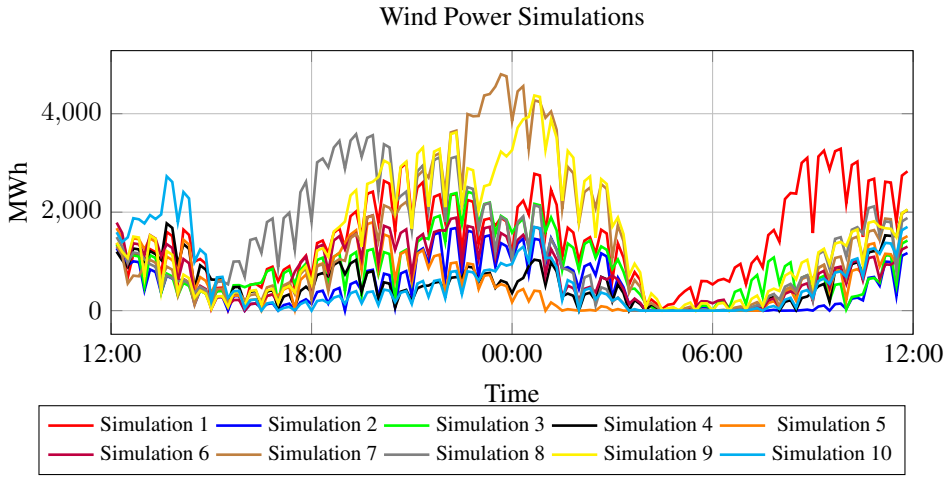

FIG. 3. *Simulated daily realizations of wind power for a given wind farm over 24 hour time horizon.*

weight in the regularization term. In this section we examine the performance of Algorithms 1 and 2 when the number of storage devices (dimension of the resource state variable) is $|R_t^x| = 50, 100, 200, 500$.

Plots of the behavior of the methods can be found in Figures 4, 5, 6, 7 below. Each figure shows the results for stagewise independence on the left, and Markov uncertainty on the right. These graphs show the convergence of the upper and lower bounds, illustrating the dramatic impact of regularization, especially as the number of dimensions grow. The results suggest that we consistently obtain high quality solutions within approximately 50 iterations for all problems.

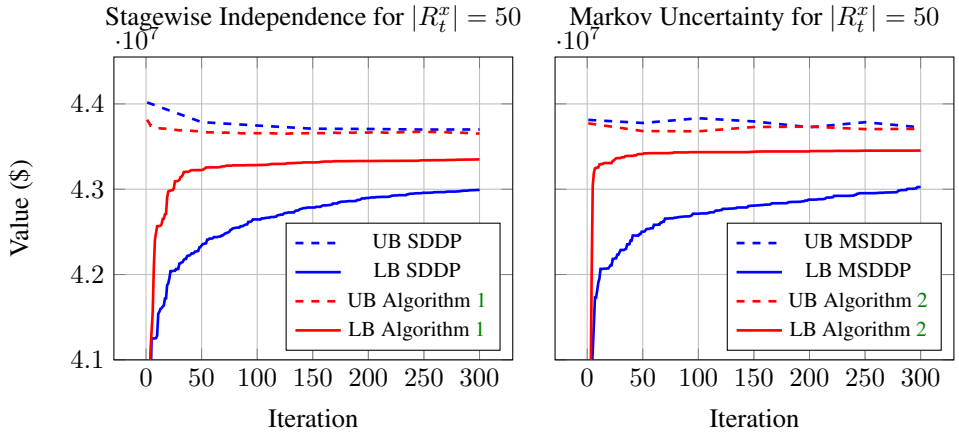

FIG. 4. *Numerical comparison of multistage stochastic optimization methods for* $|R_t^x| = 50$

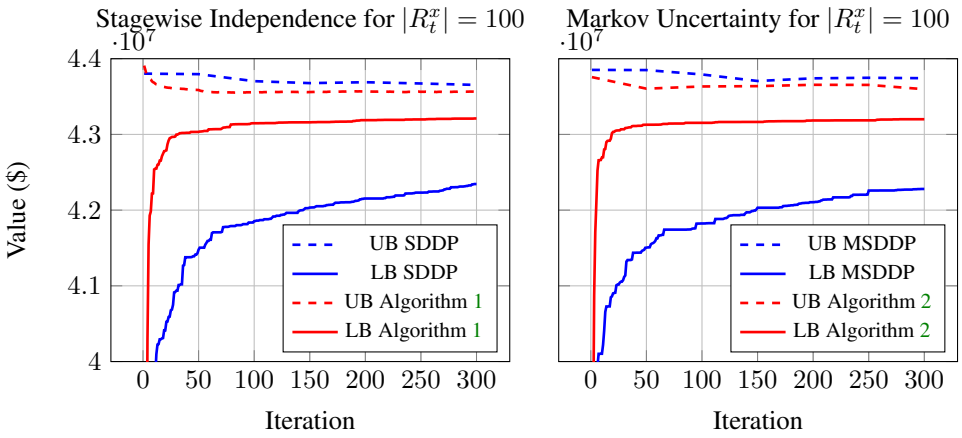

FIG. 5. *Numerical comparison of multistage stochastic optimization methods for* $|R_t^x| = 100$

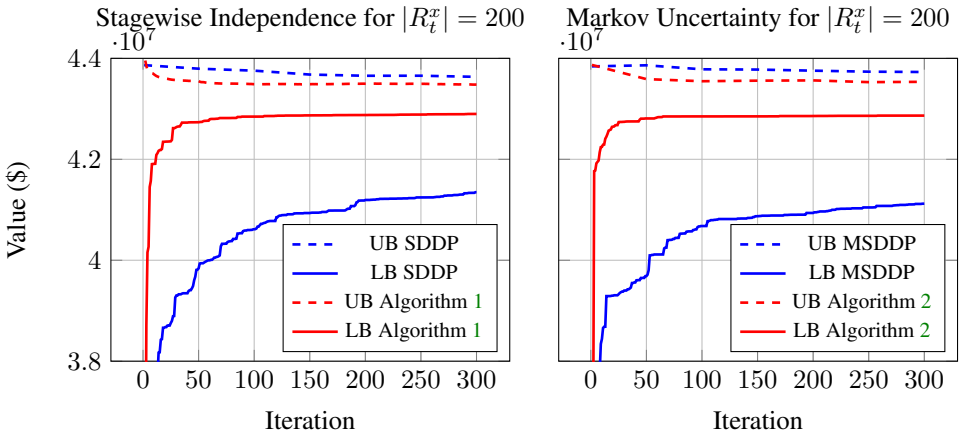

FIG. 6. *Numerical comparison of multistage stochastic optimization methods for* $|R_t^x| = 200$

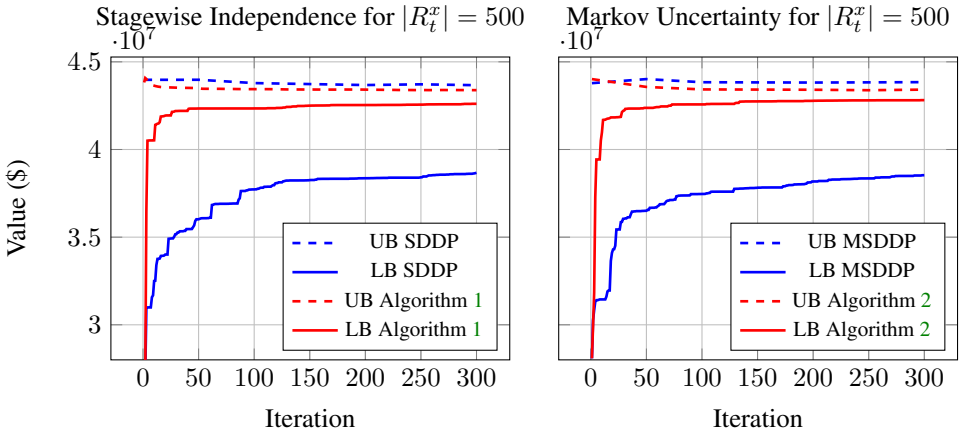

FIG. 7. *Numerical comparison of multistage stochastic optimization methods for* $|R_t^x| = 500$

$\lvert R_t^x \rvert$ \ # Iterations		1	50	100	150	200	250	300
50	Algorithm 1	182.0	217.6	230.8	248.3	266.5	284.3	299.2
	SDDP	181.0	196.2	201.2	208.9	215.4	223.4	230.4
100	Algorithm 1	237.0	306.2	334.3	371.7	412.9	453.9	500.4
	SDDP	246.0	250.0	262.2	275.1	296.0	319.7	341.4
200	Algorithm 1	293.0	358.8	414.3	507.0	587.6	653.5	726.0
	SDDP	265.0	375.3	360.7	394.5	428.8	469.8	525.5
500	Algorithm 1	553.0	664.0	828.4	995.4	1183.3	1673.5	2536.0
	SDDP	332.0	426.5	564.6	651.5	751.6	869.8	1003.2

TABLE 1
Computational time per iteration (in seconds) for risk–neutral stochastic optimization methods.

$\lvert R_t^x \rvert$ \ # Iterations		1	50	100	150	200	250	300
50	Algorithm 2	180.0	225.6	239.1	258.2	277.0	294.2	310.1
	MSDDP	181.0	192.7	198.2	206.3	213.3	221.8	228.9
100	Algorithm 2	256.0	309.5	336.1	371.5	411.2	450.3	495.6
	MSDDP	255.0	255.7	267.2	279.3	300.8	325.1	347.1
200	Algorithm 2	296.0	364.6	422.2	513.0	592.6	657.4	731.1
	MSDDP	339.0	301.6	319.4	364.9	409.3	454.8	515.1
500	Algorithm 2	542.0	650.3	799.5	959.4	1151.7	1637.6	2490.4
	MSDDP	338.0	434.7	586.7	674.3	777.2	886.8	1004.1

TABLE 2
Computational time per iteration (in seconds) for risk–neutral stochastic optimization methods.

Table 1 and Table 2 show the CPU times (in seconds) per iteration for problems with 50 to 500 storage devices, with stagewise independence and Markov uncertainty, for up to 300 iterations. We note that in a practical application, the algorithms would be run offline (for example, the day before, given a particular forecast of wind). The cuts would be stored and then used in real time the next day. This would be easily implementable in a policy updated every 5 minutes.

9. Conclusion. Large scale multistage stochastic optimization problems with long time horizons arise in numerous real–world applications in energy, finance, transportation and other fields. The numerical solution of such models can be computationally demanding, often causing practioners to face a trade–off between solution quality and computational time.

In our work we have developed regularization methods for the SDDP framework and studied their convergence. The algorithms employ regularization terms in the selection of cutting hyperplanes which improve the quality of the resulting value function approximations. The proposed techniques feature straightforward implementation and can be quickly integrated into existing software solutions without the need for major additional efforts in development and testing.

In order to assess the performance of the proposed approach we consider a model for the integration of renewable energy using distributed grid–level storage into the grid of PJM, one of the largest regional transmission operators in the United States. Our numerical experiments indicate that the proposed regularized algorithms exhibits significantly faster convergence than their non–regularized counterparts, with greater gains observed for higher–dimensional

problems.

In the future we can consider several extensions of the current work. One possible direction would involve further investigation into the selection of appropriate regularization terms and coefficients. Another possible path of exploration would be the application of regularization techniques for the solution of risk–averse models involving time–consistent compositions of coherent measures of risk along the lines of [13, 2], and [34]. Additionally, we would also like to extend the proposed approach to the solution of multiobjective stochastic models [37]. Finally, obtaining further empirical results and insights from problems in the field would also be a subject of great interest.

References.

[1] T. ASAMOV AND W. B. POWELL, *Multistage stochastic optimization methods applied to grid energy storage*, tech. report, Department of Operations Research and Financial Engineering, Princeton University, 2015.

[2] T. ASAMOV AND A. RUSZCZYŃSKI, *Time-consistent approximations of risk-averse multistage stochastic optimization problems*, Mathematical Programming, (2014), pp. 1–35,, http://dx.doi.org/10.1007/s10107-014-0813-x, http://dx.doi.org/10.1007/s10107-014-0813-x.

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

[4] H. Y. BENSON, *Interior-point linear programming solvers*, Wiley Encyclopedia of Operations Research and Management Science, (2009).

[5] J. R. BIRGE, *Decomposition and partitioning methods for multistage stochastic linear programs*, Operations Research, 33 (1985), pp. 989–1007.

[6] A. CHIRALAKSANAKUL AND D. P. MORTON, *Assessing policy quality in multi-stage stochastic programming*, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik, (2004).

[7] A. R. DE QUEIROZ AND D. P. MORTON, *Sharing cuts under aggregated forecasts when decomposing multi-stage stochastic programs*, Operations Research Letters, 41 (2013), pp. 311–316.

[8] C. J. DONOHUE AND J. R. BIRGE, *The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse*, Algorithmic Operations Research, 1 (2006).

[9] S. GÜLTEN AND A. RUSZCZYŃSKI, *Two-stage portfolio optimization with higher-order conditional measures of risk*, Annals of Operations Research, 229 (2014), pp. 409–427.

[10] J. L. HIGLE AND S. SEN, *Finite master programs in regularized stochastic decomposition*, Mathematical Programming, 67 (1994), pp. 143–168.

[11] G. INFANGER AND D. P. MORTON, *Cut sharing for multistage stochastic linear programs with interstage dependency*, Mathematical Programming, 75 (1996), pp. 241–256.

[12] J. E. KELLEY, JR, *The cutting-plane method for solving convex programs*, Journal of the Society for Industrial & Applied Mathematics, 8 (1960), pp. 703–712.

[13] V. KOZMIK AND D. MORTON, *Risk-averse stochastic dual dynamic programming*, Optimization Online. http://www. optimization-online.org/DB_HTML/2013/02/3794.html, (2013).

[14] K. LINOWSKY AND A. B. PHILPOTT, *On the convergence of sampling-based decomposition algorithms for multistage stochastic programs*, Journal of Optimization Theory and Applications, 125 (2005), pp. 349–366.

[15] M. MACEIRA AND J. DAMÁZIO, *Use of the par (p) model in the stochastic dual dynamic programming optimization scheme used in the operation planning of the brazilian hydropower system*, Probability in the Engineering and Informational Sciences, 20 (2006), pp. 143–156.

[16] B. MO, A. GJELSVIK, AND A. GRUNDT, *Integrated risk management of hydro power scheduling and contract management*, Power Systems, IEEE Transactions on, 16 (2001), pp. 216–221.

[17] D. P. MORTON, *An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling*, Annals of Operations Research, 64 (1996), pp. 211–235.

[18] Y. NESTEROV AND I. E. NESTEROV, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer, 2004.

[19] M. V. F. PEREIRA AND L. M. V. G. PINTO, *Multi-stage stochastic optimization applied to energy planning*, Mathematicsl Programming, 52 (1991), pp. 359–375.

[20] A. B. PHILPOTT AND V. L. DE MATOS, *Dynamic sampling algorithms for multistage stochastic programs with risk aversion*, tech. report, Electric Power Optimization Centre, University of Auckland, 2011.

[21] A. B. PHILPOTT AND Z. GUAN, *On the convergence of stochastic dual dynamic programming and related methods*, Operations Research Letters, 36 (2008), pp. 450–455.

[22] W. B. POWELL, *Approximate Dynamic Programming: Solving the curses of dimensionality*, John Wiley & Sons, 2011.

[23] M. L. PUTERMAN, *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons, 2014.

[24] R. T. ROCKAFELLAR, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898.

[25] A. RUSZCZYŃSKI, *A regularized decomposition method for minimizing a sum of polyhedral functions*, Mathematical Programming, 35 (1986), pp. 309–333.

[26] A. RUSZCZYŃSKI, *Regularized decomposition of stochastic programs: algorithmic techniques and numerical results*, Citeseer, (1993).

[27] A. RUSZCZYŃSKI, *Decomposition methods in stochastic programming*, Mathematical Programming, 79 (1997), pp. 333–353.

[28] A. RUSZCZYŃSKI, *Risk-averse dynamic programming for markov decision processes*, Mathematical Programming, 125 (2010), pp. 235–261.

[29] A. RUSZCZYŃSKI AND A. ŚWIETANOWSKI, *Accelerating the regularized decomposition method for two stage stochastic linear problems*, European Journal of Operational Research, 101 (1997), pp. 328–342.

[30] A. P. RUSZCZYŃSKI, *Nonlinear optimization*, vol. 13, Princeton university press, 2006.

[31] S. SEN AND Z. ZHOU, *Multistage stochastic decomposition: a bridge between stochastic programming and approximate dynamic programming*, SIAM Journal on Optimization, 24 (2014), pp. 127–153.

[32] A. SHAPIRO, *Analysis of stochastic dual dynamic programming method*, European Journal of Operational Research, 209 (2011), pp. 63–72.

[33] A. SHAPIRO, D. DENTCHEVA, AND A. RUSZCZYŃSKI, *Lectures on stochastic programming: modeling and theory*, vol. 16, SIAM, 2014.

[34] A. SHAPIRO, W. TEKAYA, J. P. DA COSTA, AND M. P. SOARES, *Risk neutral and risk averse stochastic dual dynamic programming method*, European Journal of Operational Research, 224 (2013), pp. 375–391.

[35] R. M. VAN SLYKE AND R. WETS, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM Journal on Applied Mathematics, 17 (1969), pp. 638–663.

[36] S. J. WRIGHT, *Primal-dual interior-point methods*, Siam, 1997.

[37] C. M. YOUNG, M. LI, Y. ZHU, M. XIE, E. A. ELSAYED, AND T. ASAMOV, *Multiobjective optimization of a port-of-entry inspection policy*, Automation Science and Engineering, IEEE Transactions on, 7 (2010), pp. 392–400.