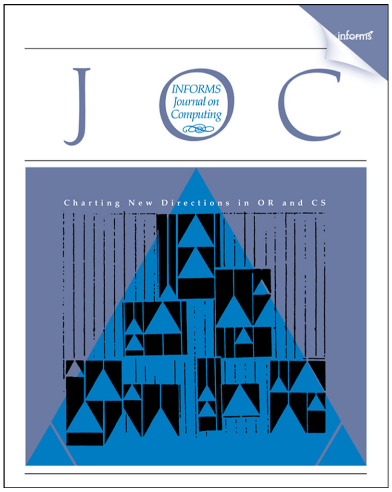

informs
J
INFORMS Journal on Computing
C
Charting New Directions in OR and CS

INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Grid-Level Energy Storage

Daniel F. Salas, Warren B. Powell

To cite this article:
Daniel F. Salas, Warren B. Powell (2018) Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Grid-Level Energy Storage. INFORMS Journal on Computing 30(1):106-123. https://doi.org/10.1287/ijoc.2017.0768

http://pubsonline.informs.org/journal/ijoc/

INFORMS JOURNAL ON COMPUTING
Vol. 30, No. 1, Winter 2018, pp. 106–123
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Grid-Level Energy Storage

Daniel F. Salas,[a] **Warren B. Powell**[b]

[a] Department of Chemical and Biological Engineering, Princeton University, Princeton, New Jersey 08540; [b] Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey 08540

Contact: dsalas@alumni.princeton.edu (DFS); powell@princeton.edu (WBP)

Received: March 2, 2015
Revised: June 29, 2015; December 31, 2015
Accepted: March 17, 2016
Published Online: November 17, 2017

https://doi.org/10.1287/ijoc.2017.0768

Copyright: © 2017 INFORMS

Abstract. We present and benchmark an approximate dynamic programming algorithm that is capable of designing near-optimal control policies for a portfolio of heterogenous storage devices in a time-dependent environment, where wind supply, demand, and electricity prices may evolve stochastically. We found that the algorithm was able to design storage policies that are within 0.08% of optimal on deterministic models, and within 0.86% on stochastic models. We use the algorithm to analyze a dual-storage system with different capacities and losses, and show that the policy properly uses the low-loss device (which is typically much more expensive) for high-frequency variations. We close by demonstrating the algorithm on a five-device system. The algorithm easily scales to handle heterogeneous portfolios of storage devices distributed over the grid and more complex storage networks.

Funding: This research was supported, in part, by National Science Foundation [Contract ECCS-1127975], and the SAP Initiative for Energy Systems Research.

Supplemental Material: The online supplement is available at https://doi.org/10.1287/ijoc.2017.0768.

Keywords: dynamic programming-optimal control • industries: electric • programming: stochastic

1. Introduction

Increasing interest in renewables over the past two decades has led to active research and development of storage technologies. Energy storage has received significant attention as a way to increase the efficiency of the electrical grid by smoothing the flow of power from renewables such as wind and solar. In particular, storage has been proposed as a strategy for reducing curtailment, time shifting, spinning reserve, peak shaving, and electricity price arbitrage, as highlighted in Dell and Rand (2001), Barton and Infield (2004), Eyer et al. (2004), Black and Strbac (2006), Beaudin et al. (2010), Zhou (2013), Xi et al. (2013). A real challenge, however, is managing a portfolio of storage devices distributed around a congested grid, in a time-dependent environment, under significant uncertainty from the variability of intermittent sources.

Applications of control theory and stochastic optimization to energy storage problems are relatively recent but they have increased in number over the last decade. In Castronuovo (2004), the variability of wind is exploited to improve operational conditions of a combined wind-hydro facility with generation and pumping capabilities. Costa et al. (2008) present a dynamic programming algorithm to operate an energy storage facility combined with a wind farm. In Teleke et al. (2010), the finite-horizon energy storage problem is studied and a rule-based dispatch solution is obtained; however, the authors do not take into account the effect of electricity prices or the uncertainty in wind energy (only variability). Dicorato et al. (2012) studies the planning and operation of a wind energy storage system in an electricity market using forecasts of the prices. In Ru et al. (2013), the storage sizing problem for a grid-level device is treated in the presence of device aging using deterministic prices and taking into account the variability in the wind energy supply. In Mokrian and Stephen (2006), a stochastic programming algorithm is used to operate a storage device specifically for energy arbitrage. Fleten and Kristoffersen (2007) account for uncertainty in spot market prices in a stochastic programming-based bidding model but it ignores the stochasticity of the energy supply. In Lohndorf and Minner (2010), an approximate dynamic programming (ADP) least-squares policy evaluation approach based on temporal differences is used to find the optimal infinite-horizon storage and bidding strategy for a system of renewable power generation and energy storage in the day-ahead market. Zhou (2013) finds a three-tiered structure for the storage policy for a single storage device with a three-dimensional state variable. The value of a storage device used for energy arbitrage in the PJM market is studied in Sioshansi and Hurlbut (2010), and Zhou et al. (2013) study the value of an energy disposal strategy in the presence of negative electricity prices for a standalone storage facility located at a market.

Lohndorf et al. (2013) present an approach named approximate stochastic dual dynamic programming (ADDP), which integrates ideas from ADP with stochastic dual dynamic programming (SDDP), which approximates the value of energy storage using multidimensional Bender's cuts. While this problem and ours are stochastic, convex optimization problems, ADDP is restricted to solving a sampled problem, which means that ADDP is a lookahead policy that has to be reoptimized as we step forward in time (since it is highly unlikely that we land in a state that was captured in the original sample when ADDP was solved). This is possible in a hydro setting, where a time step might be an entire week, but is completely impractical for our battery storage problem where time steps might be measured in minutes or less (frequency regulation decisions for battery storage are made every two seconds). Also, while ADDP enjoys bounds on the quality of the solution, bounds on the solution of a lookahead model are not bounds on the performance of the policy in the context of the full problem, which we term the *base model*.

While the management of water reservoirs has been a popular area of research (Pritchard et al. 2005, Philpott and de Matos 2012, Shapiro et al. 2013, Lohndorf et al. 2013), these problems tend to be modeled at a coarser level of aggregation, spatially and temporally, in the stochastic programming literature. These models are, of course, streamlined versions of actual problems, which can exhibit hundreds of reservoirs and complex dynamics, but standard modeling practice is to aggregate reservoirs to less than 10. Granville et al. (2003), Shapiro et al. (2013), for example, describe models of the massive Brazilian hydroelectric network, which are aggregated into four generating regions (which means four aggregate generators and four aggregate reservoirs) with monthly time steps spanning 10 years (120 stages). The solution to these aggregate models are then used to guide more detailed models. A notable exception is described in Gjerden et al. (2015), which models the large Norwegian network, consisting of 279 reservoirs, which are modeled in weekly increments over a multiyear horizon. However, the paper is not clear whether all 279 reservoirs are actually represented in the model, and we could not find an explicit statement of the planning horizon in the model.

There is a range of different grid storage technologies (pumped hydro, batteries, ultracapacitors, compressed air, flywheels), each of which exhibits different capacities, conversion efficiencies and power. Each technology is suited to different storage patterns: smaller, low-loss devices are better at high-frequency variations while larger devices, which exhibit higher losses are better for long-term storage.

In this paper, we present an ADP algorithm that is capable of designing near-optimal time-dependent control policies for finite-horizon energy storage problems, where the battery operator is a price-taker seeking to maximize revenue in the presence of stochastically evolving renewable energy supply, demand, and electricity prices. Our model works on a time scale of five minutes, which means 288 time periods over a daily cycle. The time scale is fixed by the grid operator, PJM, which updates electricity prices every five minutes. This policy fixes the time points at which a battery is operator is allowed to update the *economic base point*, which represent the rate at which a battery will charge or discharge for a five-minute period. We do not address frequency regulation, which involves instructions that are made every two seconds, whose purpose is to stabilize fine-grained power variations. At the five-minute time scale, we can handle energy arbitrage (purchasing power at lower prices, selling at higher prices), energy shifting, and peak shifting.

We emphasize that our policies apply to the base model; they are not lookahead policies that have to be re-optimized as we step forward in time. We then benchmark the algorithm against optimal policies when applied to a single storage device (we again emphasize that these are not bounds on an approximate lookahead model—we benchmark against optimal policies for the full problem). This algorithm scales to systems with many, potentially heterogeneous storage devices, which has been recommended for grid storage (see Kraining et al. 2011, Vazquez et al. 2010, Kuperman and Aharon 2011).

The algorithm is based on literature that demonstrates the effectiveness of using separable, piecewise linear, concave value function approximations (VFAs) for high-dimensional stochastic resource allocation problems such as rail fleet management (Topaloglu and Powell 2005). This work has been used in the context of a multidecade planning problem in energy (Powell et al. 2012) with a single pumped hydro reservoir, for which a convergence proof is available (Nascimento and Powell 2013). Again, the proof applies to the full-based model, not a sampled approximation. We use the principle of backpropagation (BP) (Werbos 1974, 1989, 1992). In this paper, we use separable, piecewise linear approximations, which easily scales to very large numbers of storage devices.

This paper makes the following contributions: (1) We describe in detail a scalable ADP algorithm based on piecewise linear separable VFAs for obtaining near-optimal control policies for stochastic energy storage problems. Unlike our prior work with this class of approximation (see, e.g., Topaloglu and Powell 2006; Nascimento and Powell 2013; Powell 2011a, Chap. 14), this paper uses a backward pass, which is needed to accelerate the feedback learning, given the large number of time periods. This is our first empirical work with a multidimensional "state of the world" variable, which produces a dramatic growth in the number of value functions. This paper also represents our first use of the bias-adjusted Kalman filter (BAKF) stepsize rule (George and Powell 2006a) in the setting of piecewise linear value functions, which was shown to outperform other stepsize rules and avoiding the need to tune parameters. (2) We benchmark against optimal policies (not bounds on lookahead models) on deterministic and stochastic time-dependent problems for a one-device system, which include the presence of exogenous information such as wind, prices, and demand. This is the first time we have provided optimal policies as benchmarks, which required creating discretized version of the problem (as opposed to performing discretization within the algorithm) so that classical backward dynamic programming produces an optimal policy. We note that providing optimal policies as benchmarks is surprisingly rare in the ADP/reinforcement learning (RL) communities. (3) We set forth this set of problems as a library that may be easily used to test the performance of other algorithms (this library is available at http://www.castlelab.princeton.edu/datasets.htm). (4) We demonstrate that the policies adapt to different types of storage devices (e.g., different loss rates and capacities) with different behaviors.

2. The Mathematical Model

We consider the problem of allocating energy to M grid-level storage devices over a finite-time horizon $t = 0, \Delta t, 2\Delta t, \ldots, T$, where Δt is the time step, while maximizing the undiscounted total revenue. We let $\mathcal{T} = \{0, \Delta t, 2\Delta t, \ldots, T\}$. Our model applies to storage systems that respond linearly to charge/discharge controls, an approximation that can be justified within certain storage ranges, and at the relatively large time steps that we use in this paper. Battery storage is widely used for frequency regulation where controls are updated every two seconds, a time scale where nonlinearities may become more pronounced (but this is an area of active research).

The storage devices may be colocated with a source of renewable energy like a solar or wind farm, but our model and algorithm can be applied to general grid-level storage devices. We can import from/export to the grid, in addition to satisfying a specific set of demands. Electricity may flow directly from the renewable source to each of the storage devices or it may be used to satisfy the demand. Energy from storage may be sold to the grid at any given time, and electricity from the grid may be bought to replenish the energy in storage or to satisfy the demand.

We now describe the different elements of the model:

The State of the System: The variable $S_t = (R_t, E_t, D_t, P_t)$ describes the state of the system at time t and includes all information that is necessary and sufficient to make decisions, calculate costs, and simulate the process over time:

—R_t: The resource vector containing the amount of energy in each storage device at time t in MWh. The mth component of R_t, R_{tm}, corresponds to the amount of energy in device m.

—E_t: The net amount of renewable energy available at time t in MWh.

—P_t: The price of electricity at time t in the spot market, in \$/MWh.

—D_t: The aggregate energy demand at time t, in MWh.

We let the state space $\mathcal{S}$ be the set of all states.

Static Parameters: The following is a list of parameters used throughout to characterize the storage device:

—κ_m: The energy capacity of device m in MWh.

—η_m^c, η_m^d: The charging and discharging efficiency of device m, respectively.

—γ_m^c, γ_m^d: The maximum charging and discharging rates of device m, given as MWh per time period.

The Decisions: At any point in time, the decision is given by the column vector:

$$x_t = (x_t^{ED}, x_t^{GD}, x_t^{R_1 D}, x_t^{ER_1}, x_t^{GR_1}, x_t^{R_1 G}, \ldots, \\ x_t^{R_M D}, x_t^{ER_M}, x_t^{GR_M}, x_t^{R_M G}),$$

where x_t^{IJ} is the amount of energy transferred from I to J at time t. The superscript E stands for renewable energy, D for demand, R_m for storage device m, and G for grid.

The Constraints: In our model, we require $x_t \geq 0$ for all t. At any time t, we require that the total amount of energy stored in a device does not exceed its energy capacity:

$$R_{tm} \leq \kappa_m. \tag{1}$$

We also impose that all demand at time t must be satisfied at time t:

$$x_t^{ED} + \sum_{m=1}^{M} \eta_m^d x_t^{R_m D} + x_t^{GD} = D_t, \tag{2}$$

where η_m^d captures energy conversion losses from discharging device m. The total amount of energy charged to or withdrawn from each device is constrained by its maximum charging and discharging rates:

$$x_t^{ER_m} + x_t^{GR_m} \leq \gamma_m^c, \quad \forall m, \tag{3}$$

$$x_t^{R_m D} + x_t^{R_m G} \leq \gamma_m^d, \quad \forall m. \tag{4}$$

Finally, flow conservation requires that the amount of energy transferred from the renewable source is not greater than the amount of renewable energy available:

$$x_t^{ER} + x_t^{ED} \leq E_t. \tag{5}$$

The feasible action space $\mathscr{X}_t$ is the convex set defined by (1)–(5).

We let $X_t^\pi(S_t)$ be the decision function that returns $x_t \in \mathscr{X}_t$, where $\pi \in \Pi$ represents the type of policy (which we determine later). We emphasize that this policy is time dependent (it is not just a function of a time-dependent state). Problems in this class might have ~100 time periods (15-minute intervals over 24 hours), or over 10,000 time periods (every minute over a week). The algorithm we propose in this paper handles these fine-grained applications. For benchmarking purposes, we use two problem classes, which we can solve optimally: deterministic problems with multiple storage devices, and stochastic problems with a single storage but up to three exogenous state variables. Our deterministic problems have 2,000 time periods, while our stochastic problems have 100 time periods.

We make the following remarks:

—We do not consider the effect of charging and discharging power on the storage efficiency, commonly modeled using Peukert's law (Baert and Vervaet 1999).

—We assume that grid transmission is unconstrained, but our algorithmic strategy does not require this. Sioshansi and Denholm (2013) and Zhou (2013) study energy storage problems in the presence of finite transmission capacity.

The Exogenous Information Process: The variable W_t is the vector that contains the exogenous information processes. In our model, $W_t = (\hat{E}_t, \hat{D}_t, \hat{P}_t)$:

—$\hat{E}_t$: The change in the renewable energy between times $t - \Delta t$ and t.

—$\hat{P}_t$: The change in the price of electricity between times $t - \Delta t$ and t.

—$\hat{D}_t$: The change in the demand between times $t - \Delta t$ and t.

More formally, we define the measurable space $(\Omega, \mathfrak{F})$ and we let $\mathfrak{F}_t = \sigma(\{W_1, \ldots, W_t\}) \subseteq \mathfrak{F}$ be the history up to time t, i.e., we let $\mathfrak{F}_t$ be the σ-algebra on Ω generated by the set $\{W_1, \ldots, W_t\}$. From this, it follows that $\mathscr{F}_t = \{\mathfrak{F}_{t'}\}_{t'=0}^{t}$ is a filtration.

To avoid violating the nonanticipativity condition, we assume that any variable that is indexed by t is $\mathscr{F}_t$-measurable. As a result, W_t is defined to be the information that becomes available between times $t - \Delta t$ and t. The nth sample realization of W_t is denoted $W_t^n = W_t(\omega^n)$ for sample path $\omega^n \in \Omega$. We note that our state depends only on starting conditions and $(W_{t'})_{t'=1}^{t}$, guaranteeing admissability of our policy by construction.

We assume that our exogenous process is nonstationary (to capture diurnal patterns) and are first-order Markov, which means that the exogenous changes between t and $t+1$ depend on the state of each process at time t. This allows us to ensure that the stochastic processes E_t, P_t, and D_t never become negative, and reflect reasonable upper limits. Most important, the diurnal patterns require that our algorithms learn difficult, time-dependent behaviors.

The Transition Function: Also known as the system model, S^M is a mapping from our current state S_t to the next state $S_{t+\Delta t}$, given our decision x_t and new information $W_{t+\Delta t}$ revealed between t and $t + \Delta t$. In our discrete-time model, $S^M(\cdot)$ is given by a set of difference equations.

The transition function for the energy in storage is given by $R_{t+\Delta t} = R_t + \mathbf{\Phi} x_t$, where the matrix $\mathbf{\Phi}$ is given by

$$\mathbf{\Phi} = \begin{pmatrix} 0 & 0 & \phi_1^T & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \phi_M^T \end{pmatrix},$$

where $\phi_m = (-1, \eta_m^c, \eta_m^c, -1)$ models the flow of energy into and out of each device. We use a linear model of the dynamics which we believe is accurate at the time scale of five minutes (which is fairly long for a battery). The physics become more complex for batteries that are used for frequency regulation, where the battery would be called to charge or discharge every two seconds. However, frequency regulation is a purely reactive process, which involves purely following a regulation signal, and does not require any optimization.

Throughout this paper, we use different transition dynamics for the renewable energy, price, and demand processes. They are presented in Sections 8.2, 8.3, and 9.

The Objective Function: The function $C(S_t, x_t)$ represents the contribution (reward) from being in the state S_t and making the decision x_t at time t. We assume that we are price-takers interested in maximizing revenue over time. The contribution function is just the total amount of money paid or collected when we transfer energy to and from the grid at time t: $C(S_t, x_t) = P_t D_t - P_t \sum_{m=1}^{M} (x_t^{GR_m} - \eta^d x_t^{R_m G}) - P_t x_t^{GD} = c_t^T x_t$ with the corresponding cost vector $c_t \equiv c(S_t) \in \mathbb{R}^{\dim(\mathscr{X}_t)}$. The objective function is then given by

$$F^{\pi^*} = \max_{\pi \in \Pi} \mathbf{E}\left[\sum_{t \in \mathscr{T}} c_t^T X_t^\pi(S_t)\right], \tag{6}$$

where $S_{t+\Delta t} = S^M(S_t, X_t^\pi(S_t), W_{t+\Delta t})$.

The expectation in (6) is over the stochastic process $W_1, \ldots, W_T$. The state variable is a function of this stochastic process, therefore the optimal decision decision at time t' is also random at any time $t' > t$. If we fix this process (and the policy), then (6) is deterministic. In that case, we may drop the expectation and solve the control problem using a standard batch linear program (LP):

$$F^* = \max_{x_0, \ldots, x_T} \sum_{t \in \mathscr{T}} c_t^T x_t, \tag{7}$$

such that $x_t \in \mathscr{X}_t$ for each t and subject to transition dynamics expressed as a set of constraints linking all time points. This formulation is most useful when we can make exact predictions about the wind/cloud cover, demand, and price dynamics. However, this is hardly ever the case with processes that are intrinsically stochastic. We use this formulation in Section 8.2 as a way to benchmark our approximate algorithm.

3. An ADP Approach

To solve (6), we can define the policy determined by an optimal value function $V_t^*(S_t)$, which is the maximum revenue we expect to receive from time t onward if we are in state S_t. The optimal policy π^* chooses the action, which maximizes the value of being in a state.

The optimal value function is characterized recursively by Bellman's equation:

$$V_t^*(S_t) = \max_{x_t \in \mathscr{X}_t} \left(c_t^T x_t + \mathbf{E}[V_{t+\Delta t}^*(S_{t+\Delta t}) \mid S_t] \right), \tag{8}$$

the arg max of which defines the optimal policy for making a decision, where $S_{t+\Delta t} = S^M(S_t, x_t, W_{t+\Delta t})$ and the conditional expectation is taken over the random variable $W_{t+\Delta t}$. We assume that $V_{T+\Delta t}^*(\cdot) \equiv 0$.

In our energy storage problems, the information space is continuous and multidimensional, which requires computing the conditional expectation over a set of potentially dependent, continuous random vectors. This calculation is computationally intractable and is known as the curse of dimensionality in the information space (Powell 2011b). To overcome this, we redefine the value function around the *postdecision* state variable S_t^x, which is the state after a decision has been made but before any new random information is revealed (Van Roy et al. 1997, Powell 2011b).

Using the postdecision state variable, we replace the expectation by a postdecision value function and reformulate (8) as a deterministic maximization problem:

$$V_t^*(S_t) = \max_{x_t \in \mathscr{X}_t} \left(c_t^T x_t + V_t^x(S_t^x) \right), \quad \forall\, t \in \mathscr{T}, \tag{9}$$

where $V_t^x(S_t^x) = \mathbf{E}[V_{t+\Delta t}^*(S_{t+\Delta t}) \mid S_t]$. Solving (9) requires the ability to calculate the value function for every possible state S_t^x for every t. This function is unknown to us a priori, and assigning a value to every state becomes computationally intractable when S_t^x is continuous and multidimensional, therefore we would like to estimate it accurately to determine a storage policy, which is as close to optimal as possible.

We note that the postdecision state only includes variables, which are required to compute the transition dynamics. In other words, a variable is part of the postdecision state if and only if its value at time $t + \Delta t$ is dependent on its value at time t. In our energy problems, S_t^x includes $R_t^x = R_t + \mathbf{\Phi} x_t$, and it also includes D_t, E_t, and P_t if the corresponding processes are not independent over time.

It is easy to show that the value function is concave in the resource dimension (Boyd and Vandenberghe 2004). It is concave because the resource vector is a right-hand side constraint, and any maximizing LP is concave in right-hand side constraints. However, the value function is not necessarily separable in the resource dimension. To approximate the true value function, we introduce a separable VFA $\bar{V}_t(S_t^x) = \sum_{m=1}^M \bar{V}_{tm}(R_{tm}^x, E_t, P_t, D_t)$, where each $\bar{V}_{tm}$ is piecewise linear in the postdecision resource dimension R_{tm}^x. For given values of E_t, P_t, and D_t, we write this type of VFA as

$$\bar{V}_t(R_t^x, E_t, P_t, D_t) = \sum_{m=1}^{M} \max_{r_{tm}} \{\bar{v}_{tm}^T(E_t, P_t, D_t) r_{tm}\}, \tag{10}$$

where for every m we have that $\sum_i r_{tmi} = R_{tm}^x$ and that $0 \le r_{tmi} \le \bar{r}_{tmi}$ for every component r_{tmi} of r_{tm}. The variable r_{tmi} is the resource coordinate variable for segment $i \in \{1, \ldots, K_t\}, K_t \in \mathbb{N}$, $\bar{r}_{tmi}$ is the length of segment i, $\bar{v}_{tmi}(E_t, P_t, D_t)$ is its slope.

One of the main advantages of this approximation strategy is that the VFA for each device is completely determined by the vector of slopes, $\bar{v}_{tm}(E_t, P_t, D_t) = (\bar{v}_{tmi}(E_t, P_t, D_t))_{i=1}^{K_t}$, and a corresponding set of breakpoints $\mathscr{B}_t$ such that $|\mathscr{B}_t| = K_t + 1$. This essentially renders this approximation strategy as a structured lookup table, which allows us to easily maintain concavity. With concavity, we have a guarantee that the stochastic gradient always points in the direction of the global maximum. As a result, concavity is a powerful property in this setting because it allows us to use a pure exploitation policy, avoiding the need for exploration policies that are characteristic of all RL policies for problems with discrete actions (Bertsekas and Tsitsiklis 1996, Sutton and Barto 1998, Powell 2011b, Bertsekas 2012). We note that Bender's-based methods (SDDP and ADDP) share these properties as well, but require replacing the full outcome space with a small discrete sample, which has to be enumerated for each time period, at each iteration; the errors from this approximation, for our problem setting, remain an open question.

With this kind of VFA, we still have to deal with the curse of dimensionality in the "state of the world"—the nonresource part of the state—which typically lacks structure. For this purpose, we use a simple aggregation method in the continuous renewable energy and price dimensions of the postdecision state variable (Hastie et al. 2009). We let $\mathscr{G}_E^{g_e}(\cdot)$, $\mathscr{G}_P^{g_p}(\cdot)$, and $\mathscr{G}_D^{g_d}(\cdot)$ be functions, which aggregate the renewable energy, price, and demand dimensions, respectively, where $g_e, g_p, g_d \in \mathbb{N}$ determine the level of aggregation. We let $\mathbf{g} = (g_e, g_p, g_d)$ be the aggregation multi-index and

$S_t^{\mathbf{g},x} = (R_t^x, \mathcal{G}_E^{g_e}(E_t), \mathcal{G}_P^{g_p}(P_t), \mathcal{G}_D^{g_d}(D_t))$ be the aggregated postdecision state variable.

We then construct a VFA that is separable and concave piecewise linear in the resource dimension for every $(\mathcal{G}_E^{g_e}(E_t), \mathcal{G}_P^{g_p}(P_t), \mathcal{G}_D^{g_d}(D_t))$-tuple. For notational convenience, we simply write $\bar{V}_t(S_t^{\mathbf{g},x})$ as $\bar{V}_t(S_t^x)$, with the understanding that the value function is always approximated around an aggregated state. Note that aggregation is only used in constructing the VFA and not in computing state action contributions or simulating the transitions.

We let $\tilde{W}_t = (E_t, P_t, D_t)$ such that $S_t = (R_t, \tilde{W}_t)$. Introducing the VFA into (9) (where the postdecision state eliminates the expectation) and letting $r_t = (r_{t1}, \ldots, r_{tM})$, the ADP formulation of (8) is given by

$$X_t^\pi(S_t) = \arg\max_{\substack{x_t\in\mathcal{X}_t\\ r_t\in\mathcal{R}_t}} \left(c_t^T x_t + \sum_{m=1}^{M} \bar{v}_{tm}(E_t, P_t)^T r_{tm} \right), \qquad \forall t \in \mathcal{T}, \tag{11}$$

$$\text{s.t.} \quad \mathbf{11}^T r_t - \mathbf{\Phi}_t x_t = R_t, \tag{12}$$

$$0 \le r_{tim} \le \bar{r}_{tim}, \quad \forall m \; \forall I, \tag{13}$$

where $\mathcal{R}_t$ is the feasible convex set for r_t defined by (12) and (13). Letting B be the (static) matrix of constraint coefficients in (12) and (13), and d_t be the vector of bounds on the constraints, then $\mathcal{R}_t = \{r \colon Br \le d_t(S_t)\}$. It is important to keep in mind that the coefficients in the objective function in (9) are stochastic since they contain the electricity prices. Furthermore, we index $\mathbf{\Phi}_t$ by time t to highlight the potential for stochastic losses due to congestion.

4. The Algorithm

We would like to solve (11)–(13) iteratively by generating sample observations of the slope of the VFA at one iteration and using them to update the VFA from the previous iteration. Throughout this chapter, we denote any variable at a particular point in time by a subscript t. We also include a superscript n to denote a particular realization of that variable while following sample path $\omega^n \in \Omega$. For example, S_t refers to any possible value of the state variable at time t, while S_t^n refers to the actual state visited at time t for sample path ω^n.

Since at each time t the VFA is entirely determined by the set $\mathcal{B}_t^n$ and the vectors $\bar{v}_{t1}^n, \ldots, \bar{v}_{tM}^n$, these are the only variables we are required to keep track of. To start, we initialize the VFA by letting it be zero everywhere, which is equivalent to setting $\bar{v}_{tm}^0 = \{0\}$ for every t for every m, where the superscript $n = 0$ indicates the initialization. We also let $\mathcal{B}_t^0 = \{0, \delta, 2\delta, \ldots, \kappa - \delta, \kappa\}$ for some scaling factor $\delta > 0$. We assume the set of breakpoints is constant for all t and all n, therefore we have $\mathcal{B}_t^n \equiv \mathcal{B}$ and $K_t^n \equiv K$.

At the beginning of iteration $n \ge 1$, we draw a random sample realization $\{W_1^n, \ldots, W_T^n\}$ from the full outcome space, and then step forward through time. At time t, the action is determined by solving (11) using the VFA from the previous iteration:

$$X_t^\pi(S_t^n) = \arg\max_{\substack{x_t\in\mathcal{X}_t^n\\ r_t\in\mathcal{R}_t^n}} \left((c_t^n)^T x_t + \sum_{m=1}^{M} (\bar{v}_{tm}^{n-1}(\tilde{W}_t^n))^T r_{tm} \right), \tag{14}$$

where $c_t^n \equiv c(S_t^n)$.

Since the VFA for each device is determined by a set of slopes, what we are really after is an observation of the marginal value of energy in each storage device m at time t, $\hat{v}_{tm}^n$. Once we have an observation, we can smooth it into our current estimate of the marginal value of energy that storage device using classical stochastic approximation (Robbins and Monro 1951). Letting $\bar{v}_t(S^{x,n}) = (\bar{v}_{t1}(S^{x,n}), \ldots, \bar{v}_{tM}(S^{x,n})) \in \mathbb{R}^M$ be the vector containing the estimates of the VFA at state $S_t^{x,n}$, and $\hat{v}_t(S_t^n) = (\hat{v}_{t1}(S_t^n), \ldots, \hat{v}_{tM}(S_t^n)) \in \mathbb{R}^M$ be the vector containing the observations of the marginal value of energy in each device, we have that

$$\bar{v}_{t-\Delta t}^n(S_{t-\Delta t}^{x,n}) = (1 - \alpha_t^{n-1}) \bar{v}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n}) + \alpha_t^{n-1} \hat{v}_t^n(S_t^n), \tag{15}$$

which requires a possibly stochastic stepsize sequence, $\{\alpha_t^n\}_{n=0}^\infty$, such that

$$\alpha_t^n \ge 0 \quad \text{a.s.} \; \forall n, \qquad \mathbf{E}\left[\sum_{n=0}^{\infty} (\alpha_t^n)^2 \right] < \infty, \qquad \sum_{n=0}^{\infty} \alpha_t^n = \infty \quad \text{a.s.} \tag{16}$$

We note that α_t^n represents the stepsize α_t at iteration n. In Section 5, we present two ways to estimate the unbiased observations $\hat{v}_{tm}^n(S_t^n)$ of the marginal value of energy.

Once we have computed the smoothed estimate $\bar{v}_{t-\Delta t,m}^n(S_{t-\Delta t}^{x,n})$ for each m, we use the concave adaptive value estimation (CAVE) algorithm to update of the VFA. The CAVE algorithm performs a simple projection operation to enforce concavity of the piecewise linear approximation while updating the slopes (see Godfrey and Powell 2001 for a detailed presentation). It is important to note that we use the observation of the slope at S_t^n to update the VFA at the previous postdecision state $S_{t-\Delta t}^{x,n}$. Recall that $V_{t-\Delta t}^x(S_{t-\Delta t}^x) = \mathbf{E}[V_t^*(S_t) \mid S_{t-\Delta t}]$. Therefore, by the chain rule, we have the following for a fixed sample path n and for each device m:

$$\begin{aligned} \hat{v}_{t-\Delta t,m}^n(S_{t-\Delta t}^{x,n}) &= \frac{\partial}{\partial R_{tm}^n} \left(\max_{x_t \in \mathcal{X}_t} (C(S_t^n, x_t) + V_t^x(S_t^{x,n})) \right) \frac{dR_{tm}^n}{dR_{t-\Delta t,m}^{x,n}} \\ &= \hat{v}_{tm}^n(S_t^n). \end{aligned}$$

For many problems, even stepsizes that satisfy (16) do not produce numerically convergent algorithms if they decline too quickly or too slowly, or if they do not adapt to nonstationary data. For our particular application, we test a harmonic stepsize rule, which is deterministic, and the BAKF stepsize rule, which is time dependent and was designed for stochastic simulations (George and Powell 2006b).

The harmonic stepsize rule can be computed as $\alpha_t^n = a/(a+n)\ \forall\, t$, where a is a tunable parameter. The BAKF stepsize rule is given by $\alpha_t^n = 1-(\bar{\sigma}_t^n)^2/\bar{\nu}_t{}^n$, where $(\bar{\sigma}_t^n)^2$ is an estimate of the variance of the error, $\varepsilon_t^n = \bar{v}_t^{n-1} - \hat{v}_t^n$, and $\bar{\nu}_t^n$ is an estimate of the total variation. We can construct these estimates as

$$\bar{\beta}_t^n = (1-\eta_t^{n-1})\bar{\beta}_t^{n-1} + \eta_t^{n-1}\varepsilon_t^n,$$
$$\bar{\nu}_t^n = (1-\eta_t^{n-1})\bar{\nu}_t^{n-1} + \eta_t^{n-1}(\varepsilon_t^n)^2.$$

An estimate of the variance of the error can be calculated as $(\bar{\sigma}_t^n)^2 = (\bar{\nu}_t^n - (\bar{\beta}_t^n)^2)(1+\lambda_t^{n-1})$. Here, λ_t^n is a coefficient that is calculated recursively: $\lambda_t^n = (1-\alpha_t^{n-1})^2\lambda_t^{n-1} + (\alpha_t^{n-1})^2$, and η^n is a McClain stepsize: $\eta^n = \eta^{n-1}/(1+\eta^{n-1}-\bar{\eta})$. The calculation of these estimates introduces a scale-free tunable parameter, $\bar{\eta}$, into the BAKF stepsize rule.

5. Computing the Marginal Value of Energy

To illustrate the computation of the marginal value of energy in storage, we formulate the problem of approximating the slopes of the value function based on noisy observations using the stochastic approximation method. We first let $h(\bar{v}_t(s)) = \min_{\bar{v}_t(s)} \mathbf{E}[H(\bar{v}_t(s), \hat{v}_t(s))]$, where $H(\bar{v}_t(s), \hat{v}_t(s)) = \frac{1}{2}\|\bar{v}_t(s) - \hat{v}_t(s)\|^2$. That is, we are interested in minimizing the expected squared error between our estimate and the observation of the marginal value of being in some state s. Letting $\{\hat{v}_t^n(s)\}_{n=0}^{\infty}$ be a random sequence of unbiased observations of $\bar{v}_t$ and letting $g^{n+1}(s)$ be a stochastic subgradient of H at $\hat{v}_t^{n+1}(s)$ with respect to $\bar{v}_t^n(s)$, the stochastic approximation algorithm generates a random sequence $\{\bar{v}_t^n(s)\}_{n=0}^{\infty}$:

$$\begin{aligned}\bar{v}_t^{n+1}(s) &= \bar{v}_t^n(s) - \alpha^n g^{n+1}(s)\\ &= (1-\alpha^n)\bar{v}_t^n(s) + \alpha^n \hat{v}_t^{n+1}(s),\end{aligned} \tag{17}$$

which converges $a.s.$ to the minimizer $\bar{v}_t^*(s)$ of $h(\bar{v}_t(s))$ for any stepsize sequence $\{\alpha^n\}$ satisfying (16). We see that this is the same as (15) with $s = S_t^{x,n}$.

We now turn our attention to ways of computing the observations $\hat{v}_t^n(s)$. Suppose that we follow a fixed policy π (at a given iteration n) determined by our VFA starting at time t and ending at the end of our time horizon $\mathcal{T}$. In this case, we obtain the following sequence of events:

$$S_t^n \to x_t^n \to S_t^{x,n} \to W_{t+\Delta t}^n \to S_{t+\Delta t}^n \to \cdots \to S_T^n \to x_T^n \to S_T^{x,n},$$

where $x_t^n = X_t^\pi(S_t^n)$, $S_t^{x,n} = (R_t^n + \mathbf{\Phi} x_t^n, \tilde{W}_t^n)$, and $S_{t+\Delta t} = S^M(S_t, x_t^n, W_{t+\Delta t}^n)$, and an observation of the (total) value of energy at each time period is given by $\hat{C}^n(S_t^n, x_t^n) = (c_t^n)^T x_t^n$ (recall that $c_t^n \equiv c(S_t^n)$). We can compute an observation of the value over the horizon starting at the predecision state at time t as $\hat{V}_t^n(S_t^n) = \sum_{\tau\in\mathcal{T}:\ \tau\ge t} \hat{C}^n(S_\tau^n, x_\tau^n)$, which can be rearranged as

$$\begin{aligned}\hat{V}_t^n(S_t^n) =&\ \bar{V}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n})\\ &+ \sum_{\tau\in\mathcal{T}:\tau\ge t}\big(\hat{C}^n(S_\tau^n, x_\tau^n) + \bar{V}_\tau^{n-1}(S_\tau^{x,n}) - \bar{V}_{\tau-\Delta t}^{n-1}(S_{\tau-\Delta t}^{x,n})\big),\end{aligned}$$

where we added and subtracted the VFA terms to the right-hand side.

We define the *TD* (also known as the Bellman error) δ_t^π as

$$\delta_t^{\pi,n}(S_{t-\Delta t}^{x,n}) \equiv \hat{C}^n(S_t^n, x_t^n) + \bar{V}_t^{n-1}(S_t^{x,n}) - \bar{V}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n}),$$

which is the difference between the current estimate $\bar{V}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n})$ and the observation $\hat{V}_t^n(S_t^n)$. This allows us to express $\hat{V}_t^n(S_t)$ as

$$\hat{V}_t^n(S_t^n) = \bar{V}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n}) + \sum_{\tau\in\mathcal{T}:\tau\ge t}\delta_\tau^{\pi,n}(S_{\tau-\Delta t}^{x,n}),$$

which is known as the TD update rule in the RL community (Sutton and Barto 1998). In our algorithm, we use a generalization of TD known as the TD(λ) rule:

$$\hat{V}_t^n(S_t^n) = \bar{V}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n}) + \sum_{\tau\in\mathcal{T}:\tau\ge t}\lambda^{\tau-t}\delta_\tau^{\pi,n}(S_{\tau-\Delta t}^{x,n}), \tag{18}$$

for some algorithmic discount $\lambda\in[0,1]$, which places a greater weight on observations that occur closer to t than those that occur further down the time horizon.

To obtain an observation of the marginal value of energy in storage, we need to differentiate (18) with respect to $R_{t-\Delta t}^{x,n}$. This results in a variation of classical TD learning for derivatives. The derivative of the first term in the right-hand side of (18) is just our current estimate of the slopes of the VFA, $\bar{v}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n})$, at $S_t^{x,n} = (R_t^{x,n}, \tilde{W}_t^n)$. A TD($\lambda$) observation of the marginal value of energy in storage is then given by

$$\hat{v}_t^n(S_t^n) = \bar{v}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n}) + \nabla_{R_{t-\Delta t}^{x,n}}\left(\sum_{\tau\in\mathcal{T}:\tau\ge t}\lambda^{\tau-t}\delta_\tau^{\pi,n}(S_{\tau-\Delta t}^{x,n})\right). \tag{19}$$

Introducing (19) into (17) gives us

$$\begin{aligned}\bar{v}_{t-\Delta t}^n(S_{t-\Delta t}^{x,n}) =&\ (1-\alpha^{n-1})\bar{v}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n})\\ &+ \alpha^{n-1}\nabla_{R_{t-\Delta t}^{x,n}}\left(\sum_{\tau\in\mathcal{T}:\tau\ge t}\lambda^{\tau-t}\delta_\tau^{\pi,n}(S_{\tau-\Delta t}^{x,n})\right),\end{aligned}$$

which is the stochastic approximation update for our VFA using TD(λ) observations.

In Section 6, we review the main idea behind an algorithm known as approximate value iteration, which uses TD(0)-type observations to update the estimates of the marginal value of energy in storage. Then, in Section 7, we propose a BP algorithm based on TD(1)-type observations, which we have found improves the performance of the storage control policy over approximate value iteration.

6. Approximate Value Iteration

Nascimento and Powell (2013) (henceforth referred to as N&P) presents in detail a form of classical approximate value iteration for a finite-horizon *single storage* control problem based on the SPAR algorithm (Powell et al. 2004). The algorithm, named SPAR Storage, is outlined in Table 1. This method makes use of TD(0) observations of the marginal value of energy in the storage device to update a concave piecewise linear VFA. Introducing $\lambda = 0$ into (19), we get

$$\begin{aligned}\hat{v}_t^n(R_t^n, \tilde{W}_t^n) &= \bar{v}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n}) \\ &\quad + \frac{\partial}{\partial R_{t-\Delta t}^{x,n}}\big(\hat{C}^n(S_t^n, x_t^n) + \bar{V}_t^{n-1}(S_t^{x,n}) - \bar{V}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n})\big).\end{aligned}$$

We note that $(\partial/\partial R_{t-\Delta t}^{x,n})\bar{V}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n}) = \bar{v}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n})$, which gives us

$$\hat{v}_t^n(R_t^n, \tilde{W}_t^n) = \frac{\partial}{\partial R_t^n}\Big(\max_{x_t \in \mathscr{X}_t^n}(C(R_t^n, \tilde{W}_t^n, x_t) + \bar{V}_t^{n-1}(R_t^n + \mathbf{\Phi} x_t, \tilde{W}_t^n))\Big). \tag{20}$$

This quantity may be easily obtained from the dual solution to (20). However, the VFA may be nondifferentiable at any breakpoint, and at such a point, there exist right and left derivatives. The dual solution may be either of these values, or any value between these two if $R_t^n \in \mathscr{B}$. More importantly, the dual values may very inaccurate at the boundaries of the resource domain, i.e., when $R_t^n = 0$ or $R_t^n = \kappa$. To overcome this, we approximate (20) with right and left numerical derivatives, which is quite fast computationally since it only requires resolving the warm-started LP. These are computed using

$$\hat{v}_t^{n+}(S_t^n) = \frac{1}{\delta}\big(F^+(S_t^{n+}, v_t(\cdot, \tilde{W}_t)) - F(S_t^n, v_t(\cdot, \tilde{W}_t))\big) \quad \text{and} \tag{21}$$

$$\hat{v}_t^{n-}(S_t^n) = \frac{1}{\delta}\big(F(S_t^n, v_t(\cdot, \tilde{W}_t)) - F^-(S_t^{n-}, v_t(\cdot, \tilde{W}_t))\big). \tag{22}$$

Table 1. An Outline of the Approximate Value Iteration (AVI) Algorithm in Section 6

```
initialize $\bar{v}_t^0\ \forall\, t \in \mathscr{T}$
for $n = 1, \ldots, N$,
    draw $\omega^n \in \Omega$
    for $t = 0, \ldots, T$:
        solve $x_t^n = \arg\max_{x_t \in \mathscr{X}_t}(C(S_t^n, x_t) + \bar{V}_t^{n-1}(S_t^{x,n}))$
        calculate $\hat{v}_t^{n+}$ and $\hat{v}_t^{n-}$ as in (21) and (22)
        update $\bar{v}_{t-\Delta t}^n(S_{t-\Delta t}^x) \leftarrow \text{SPAR}(\bar{v}_{t-\Delta t}^{n-1}, \hat{v}_t^n)$
        if $t < T$,
            compute $S_{t+\Delta t}^n = S^M(S_t^n, x_t^n, W_t^n + \Delta t)$
    if $n < N$,
        $n \leftarrow n + 1$
    else
        return $(\bar{v}^N)_{t=0}^T$
```

The algorithm in N&P may be easily extended to multidevice problems if we introduce the separable VFA described in this paper. However, in that case, we lose the theoretical convergence guarantees presented in N&P. Appendix B in the online supplement summarizes the technical conditions, which guarantee asymptotic performance of the algorithm.

7. Backpropagation Through Time

An important limitation of approximate value iteration is that the rate of convergence of the algorithm may decrease significantly if there are many time steps between the time when a decision is made and the time when it has an impact on the solution, as is with relevant energy storage problems that we are interested in solving. For example, we may decide to store energy at 10 a.m., but the marginal impact is not felt until 3 p.m., which may be hundreds of time periods in the future. The decrease in the rate of convergence is due to the discounting role played by the stepsize as $\hat{v}_t^n$ is smoothed into $\bar{v}_{t-\Delta t}^{n-1}$.

In this section, we propose computing an observation of the slope using a double pass algorithm known as backpropagation through time (BPTT) in the artificial neural networks community (Werbos 1990), a variation of classical TD(1) learning for estimating the slopes of functions. We claim that this alternative form of obtaining observations of the marginal value of energy in storage results in an algorithm that numerically outperforms approximate value iteration.

Introducing $\lambda = 1$ into (19), a TD(1)-type observation of $\hat{v}_t^n(S_t^n)$ is given by

$$\begin{aligned}\hat{v}_t^n(S_t^n) = \bar{v}_{t-\Delta t}^{n-1}(S_{t-\Delta t}^{x,n}) + \nabla_{R_{t-\Delta t}^{x,n}}\Bigg(\sum_{\tau \in \mathscr{T}:\, \tau \geq t}(\hat{C}^n(S_\tau^n, x_\tau^n) \\ + \bar{V}_\tau^{n-1}(S_\tau^{x,n}) - \bar{V}_{\tau-\Delta t}^{n-1}(S_{\tau-\Delta t}^{x,n}))\Bigg).\end{aligned} \tag{23}$$

We let $\hat{v}_t^n(S_t^n) \in \mathbb{R}^M$ be the vector containing the observation of the marginal value of energy in each device. We then let $\nabla_{R_{t-\Delta t}^{x,n}} \hat{C}^n(S_t^n, x_t^n)$ be the gradient of the cost with respect to $R_{t-\Delta t}^{x,n}$, and

$$J_{R_{t-\Delta t}^x}(R_t^x) = \frac{\partial(R_{t,1}^x, \ldots, R_{tM}^x)}{\partial(R_{t-\Delta t}^x, \ldots, R_{t-\Delta t,M}^x)} = \begin{pmatrix} \dfrac{\partial R_{t1}^x}{\partial R_{t-\Delta t,1}^x} & \cdots & \dfrac{\partial R_{t1}^x}{\partial R_{t-\Delta t,M}^x} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial R_{tM}^x}{\partial R_{t-\Delta t,1}^x} & \cdots & \dfrac{\partial R_{tM}^x}{\partial R_{t-\Delta t,M}^x} \end{pmatrix}$$

be the Jacobian of R_t^x with respect to $R_{t-\Delta t}^x$. Noting that the sum in (23) is telescoping and using the assumption that $\bar{V}_T^{n-1}(S_T^{x,n}) \equiv 0$, we obtain

$$\begin{aligned}\hat{v}_t^n(S_t^n) &= \nabla_{R_{t-\Delta t}^{x,n}} \Bigg(\sum_{\tau \in \mathcal{T}:\tau \geq t} \hat{C}^n(S_\tau^n, x_\tau^n) \Bigg) \\ &= \nabla_{R_{t-\Delta t}^{x,n}} \hat{C}^n(S_t^n, x_t^n) \\ &\quad + \sum_{\tau \in \mathcal{T}:\tau \geq t} \Bigg(\Bigg(\prod_{\tau' \in \mathcal{T}: t \leq \tau' \leq \tau} (J_{R_{\tau'-\Delta t}^{x,n}}(R_{\tau'}^{x,n}))^T \Bigg) \\ &\qquad \cdot \nabla_{R_\tau^{x,n}} \hat{C}^n(S_{\tau+\Delta t}^n, x_{\tau+\Delta t}^n) \Bigg), \qquad (24)\end{aligned}$$

where the last line follows from the chain rule, since the level in storage at any time t is dependent on the storage level in the previous time periods, i.e., $R_t^{x,n}$ is a function of $R_{t-\Delta t}^{x,n}$, which is a function of $R_{t-2\Delta t}^{x,n}$, and so on. It is easy to show that (24) can be conveniently expressed recursively as

$$\hat{v}_t^n(S_t^n) = \begin{cases} \nabla_{R_{t-\Delta t}^{x,n}} \hat{C}^n(S_t^n, x_t^n) + J_{R_t^x}^T \hat{v}_{t+\Delta t}^n(S_{t+\Delta t}^n) & \text{for } t < T, \\ \nabla_{R_{T-\Delta t}^{x,n}} \hat{C}^n(S_T^n, x_T^n) & \text{otherwise.} \end{cases} \qquad (25)$$

The recursive nature of (25) requires us to step forward through the end of the horizon before we can compute the observations of the slope, rendering BPTT as a purely offline learning strategy. In the forward pass, at time t, we compute an observation of the marginal contribution:

$$\hat{c}_t^{\pi,n} = \nabla_{R_{t-\Delta t}^{x,n}} \hat{C}^n(S_t^n, X_t^\pi(S_t^n)). \qquad (26)$$

In BPTT, $\hat{c}_t^{\pi,n}$ cannot be obtained from the dual solution to (24) since the dual solution would represent the marginal change in the entire objective, and we are only interested in the marginal contribution. As a result, we approximate (26) with right and left numerical derivatives, which is quite fast computationally since it only requires resolving the warm-started LP. We let $S_t^{n,m+} = (R_t^n + \mathbf{e}_m\delta, E_t^n, D_t^n, P_t^n)$, where $\mathbf{e}_m$ is the standard M-dimensional unit column vector with a 1 in the mth component. We then let $x_t^{n,m+} = X_t^\pi(S_t^{n,m+})$. Similarly, we define $S_t^{n,m-}$ and $x_t^{n,m-}$. We can approximate the mth component of the gradient, $\partial \hat{C}^n(S_t^n, x_t^n)/\partial R_{t-\Delta t,m}^{x,n}$, with positive and negative numerical gradients given by

$$\hat{c}_{tm}^{n+} = \frac{1}{\delta}\big(\hat{C}(S_t^{n,m+}, x_t^{n,m+}) - \hat{C}(S_t^n, x_t^n)\big), \qquad (27)$$

$$\hat{c}_{tm}^{n-} = \frac{1}{\delta}\big(\hat{C}(S_t^n, x_t^n) - \hat{C}(S_t^{n,m-}, x_t^{n,m-})\big). \qquad (28)$$

respectively. The right numerical gradient is then given by

$$\begin{aligned}\nabla_{R_{t-\Delta t}^{x,n}} \hat{C}^n(S_t^n, x_t^n) &\approx (\hat{c}_{t1}^{n+} \; \cdots \; \hat{c}_{tM}^{n+}) \\ &\equiv \hat{c}_t^{n+}.\end{aligned}$$

The left numerical approximation is defined analogously.

The BPTT algorithm is based on tracking the marginal value of energy in storage over time by connecting sequences of one-period subproblems in which an incremental perturbation results in holding additional energy in storage. This additional energy held in storage is captured by the components of the Jacobian, i.e., the marginal flows, $\partial R_{tm}^{x,n}/\partial R_{t-\Delta t,m'}^{x,n}$, for devices m and m'. We also approximate the marginal flows using numerical derivatives. We note that the mth column of the Jacobian is the vector of marginal flows with respect to device m.

Letting $\mathbf{\Phi}_{t,m}^T$ be the mth row of $\mathbf{\Phi}_t$, we approximate the mth column of the Jacobian with positive and negative numerical gradients given by

$$\Delta_{tm}^{n+} = \frac{1}{\delta}\mathbf{\Phi}_{t,m}^T(x_t^{n,m+} - x_t^n) \qquad (29)$$

$$\Delta_{tm}^{n-} = \frac{1}{\delta}\mathbf{\Phi}_{t,m}^T(x_t^n - x_t^{n,m-}), \qquad (30)$$

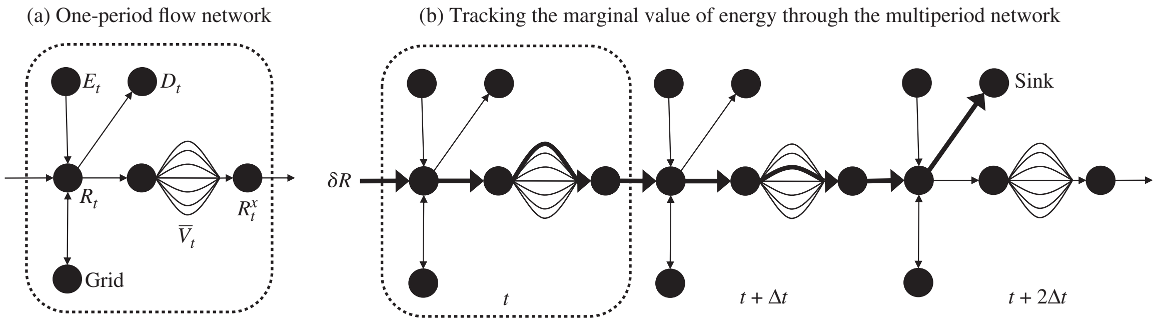
respectively. We may then approximate the Jacobian with the matrix $\mathbf{\Delta}_t^{n+}$:

$$\begin{aligned}J_{R_t^x} &\approx (\Delta_{t1}^{n+} \; \cdots \; \Delta_{tM}^{n+}) \\ &\equiv \mathbf{\Delta}_t^{n+},\end{aligned}$$

and its negative analog.

In Figure 1, we illustrate the computation of the marginal value from a network flow perspective for a problem with a single device. In Figure 1(a), we show

Figure 1. Tracking the Marginal Value of Energy Through the Multiperiod Network

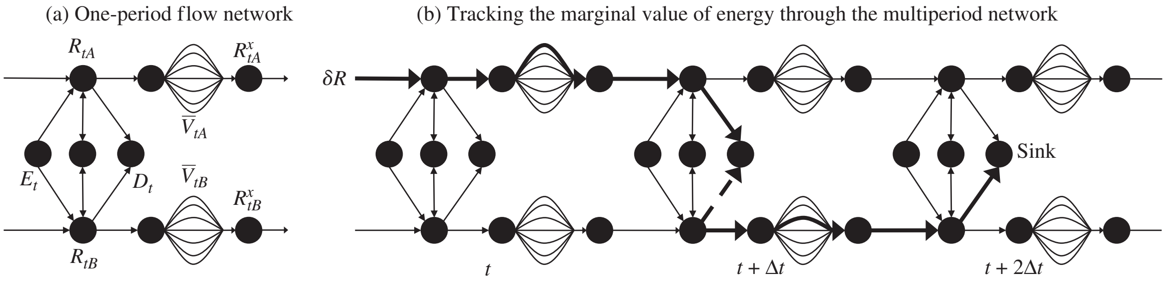


(a) One-period flow network

(b) Tracking the marginal value of energy through the multiperiod network

Notes. In Figure 1(a), an illustration of the flow network for a single device. In Figure 1(b), an illustration of the marginal value calculation for a single device. Bold arrows indicate increased flow due to the perturbation $+\delta R$.

Figure 2. An Illustration of the Marginal Value Calculation for the Two-Dimensional Problem


(a) One-period flow network

(b) Tracking the marginal value of energy through the multiperiod network

Note. Bold arrows indicate increased flow due to the perturbation $+\delta R$, while the dashed arrow indicates a decrease in flow.

the network structure for a single time period. In Figure 1(b), we illustrate a three-time period sequence with the marginal flow through the network shown as a bold line. In other words, the bold line is the flow augmenting path through the network that would result if we started with an infinitesimal extra amount of energy, represented by $+\delta$, at time t. The marginal value over these three-time periods would be the additional contribution received over this path.

In contrast with the one-dimensional case, this perturbation to some device at one-time period may now result in energy held in storage from any other device at a later time. This possibility is illustrated in Figure 2, where we show the extension to the two-dimensional case from the network flow perspective. In this case, an additional amount δ in device 1 at time t is held in storage until $t+\Delta t$; this extra amount is now used to satisfy part of the demand that was originally satisfied by device 2 at time $t+\Delta t$ (this could happen, for example, if device 1 has a lower loss than device 2). As a result, device 2 now has an additional amount δ that can be held on to until $t+2\Delta t$, where it is finally used. The marginal value of energy in device 1 at time t is exactly the additional contribution over this marginal multiperiod path.

Table 2. An Outline of the BPTT Algorithm Presented in Section 7

```
initialize $\bar{v}^0_t\ \forall\, t \in \mathcal{T}$
for $n = 1, \ldots, N$,
    draw $\omega^n \in \Omega$
    for $t = 0, \ldots, T$:
        solve $x^n_t = \arg\max_{x_t \in \mathcal{X}_t} (C(S^n_t, x_t) + \bar{V}^{n-1}_t(S^{x,n}_t))$
        for $m = 1, \ldots, M$:
            calculate $\hat{c}^{n+}_{tm}$ and $\hat{c}^{n-}_{tm}$ as in (27) and (28)
            calculate $\Delta^{n+}_{tm}$ and $\Delta^{n-}_{tm}$ as in (29) and (30)
        if $t < T$,
            compute $S^n_{t+\Delta t} = S^M(S^n_t, x^n_t, W^n_{t+\Delta t})$
    for $t = T, \ldots, \Delta t$:
        calculate $\hat{v}^{n+}_t$ and $\hat{v}^{n-}_t$ as in Equation (31)
        update $\bar{v}^n_{t-\Delta t}(S^x_{t-\Delta t}) \leftarrow \mathrm{CAVE}(\bar{v}^{n-1}_{t-\Delta t}, \hat{v}^n_t)$
    if $n < N$,
        $n \leftarrow n + 1$
    else
        return $(\bar{v}^N_t)^T_{t=0}$
```

From this perspective, it is easy to see how the algorithm may be extended to portfolios of dimension higher than two. We note that with each additional storage device, we simply have to add another piecewise linear function to the objective function. The resulting decision problem is nothing more than a slightly larger LP, which grows linearly with the number of devices. Computationally, we would have no difficulty handling hundreds or even thousands of devices, as might arise with distributed storage and more complex storage networks.

After recording the observation of the marginal contribution and the marginal flow, we transition to the next time period. Since the VFA is concave, we take advantage of a pure exploitation strategy for computing the evolution of the system. For $t < T$, we compute the next state as $S^n_{t+\Delta t} = S^M(S^n_t, x^n_t, W^n_{t+\Delta t})$. Once we have reached the end of the horizon, we then sweep backward in time computing observations of the slopes that we can use to update the VFA. With this in mind, we compute the marginal value of energy in storage as follows:

$$\hat{v}^{n+}_t(S^n_t) = \begin{cases} \hat{c}^{n+}_t + (\boldsymbol{\Delta}^{n+}_t)^T \hat{v}^{n+}_{t+\Delta t}(S^n_{t+\Delta t}) & \text{if } 0 \le t < T, \\ \hat{c}^{n+}_T & \text{if } t = T, \end{cases} \tag{31}$$

with the corresponding analogous equation for $\hat{v}^{n-}_t(S^n_t)$. The BPTT algorithm is outlined in Table 2.

This way of calculating the observation of the slope facilitates the transfer of information backward in time over the iterations but it is more susceptible to underlying noise. Even though there is no optimality guarantee for this BPTT algorithm, it performs significantly better than approximate value iteration on time-dependent problems where energy may be held in storage for many time periods, as discussed in Section 8.4.

8. Algorithmic Performance Analysis

In this section, we assess the optimality of the BPTT algorithm by comparing the performance of the resulting approximate policy to optimal for time-dependent deterministic problems with 2,000 time periods, and optimal policies for a variety of stochastic problems. The deterministic comparison is done against benchmark problems that can be solved exactly using the LP formulation described at the end of Section 2. The stochastic benchmarks consist of discretized problems for which the exact solution can be found by solving (8) exactly. All the benchmark problems are available at http://www.castlelab.princeton.edu/datasets.htm.

Next, we first benchmark the BPTT algorithm on deterministic solutions of time-dependent problems, which can be solved optimally as an LP. We then benchmark on stochastic problems with a single storage device, which can be discretized and solved optimally as a Markov decision process (MDP), and we compare the numerical performance of BPTT against approximate value iteration.

8.1. Algorithm Tuning

Before using the algorithm, we had to tune the process of state aggregation and choice of stepsize algorithm for smoothing new estimates into old. These are described in Appendix C in the online supplement, which reports on experiments testing different levels of state discretation, and comparisons of different stepsize formulas. For stepsizes, we compared simple harmonic stepsize rules with the BAKF rule described in Powell (2011a).

8.2. Deterministic Experiments

For the deterministic benchmarks, we designed the test problems shown in Table 3, where the electricity prices, renewable energy and energy demand evolve deterministically over time. We consider four different dynamics: sinusoidal, constant, step, or fluctuating with no particular pattern. All test problems consist of $T = 2{,}000$ periods, $t = 0, \Delta t, 2\Delta t, \ldots, T$, where $\Delta t = 1$. We consider a generic storage device with a capacity $\kappa = 100$, a round-trip efficiency $\eta^c\eta^d = 0.81$, and maximum rates $\gamma^c = \gamma^d = 0.1$. We also assume $R_0 = 0$. We note that these time-dependent problems are actually quite hard; the policies have to learn to store energy at precisely the right point in time so that it is available when it is needed, which may be hundreds of time periods later.

Table 3. Deterministic Test Problems

Label	Price	Renewables	Demand	F^*	$\mathbf{F}^{1,000}$ (%)
D1	Sinusoidal	Constant	Sinusoidal	2,967.47	99.99
D2	Sinusoidal	Step	Step	1,233.02	99.92
D3	Sinusoidal	Step	Sinusoidal	1,240.78	99.97
D4	Sinusoidal	Sinusoidal	Step	1,419.04	99.98
D5	Constant	Constant	Sinusoidal	2,657.21	99.97
D6	Constant	Step	Step	882.94	99.93
D7	Constant	Step	Sinusoidal	892.30	99.98
D8	Constant	Sinusoidal	Step	1,029.29	99.99
D9	Fluctuating	Fluctuating	Sinusoidal	3,296.57	99.97
D10	Fluctuating	Fluctuating	Constant	8,104.44	99.96

The problem instances are shown in Table 3. We obtain the optimal solution to each one of the test problems by solving one LP given by (7) subject to (1)–(5) for each t and to the transition dynamics expressed as constraints, which link together all the time periods.

To quantify the optimality of the algorithm on this set of problems, we compared the objective value given by BPTT after N iterations, F^N, to the true optimal value given by LP, F^*, by the performance metric

$$\mathbf{F}^N = \frac{F^N}{F^*}. \tag{32}$$

The results are shown in Table 3. In Figure 3(a), we show a plot of the storage profile for test problem D1 obtained by ADP and LP in the presence of constant supply and sinusoidal demand, and they both coincide. It is clear that energy is stored in the device just ahead of each of the humps in demand despite having excess-free supply from renewables at earlier points in time.

We also tested more complex dynamics for the renewable energy supply and the energy spot prices. Figure 3(b) shows the storage level (scaled to fit) obtained BPTT along with the renewable energy and demand profiles corresponding to test problem D9. Figure 3(c) shows the spot price process. The optimal storage profile and the one generated by the approximate policy coincide.

8.3. Discretized Stochastic Experiments

The optimal solution to stochastic problems can only be computed for problems, which have denumerable and relatively small state, decision, and outcome spaces, which also limits us to problems with a single storage device. In these cases, (8) may be rewritten as

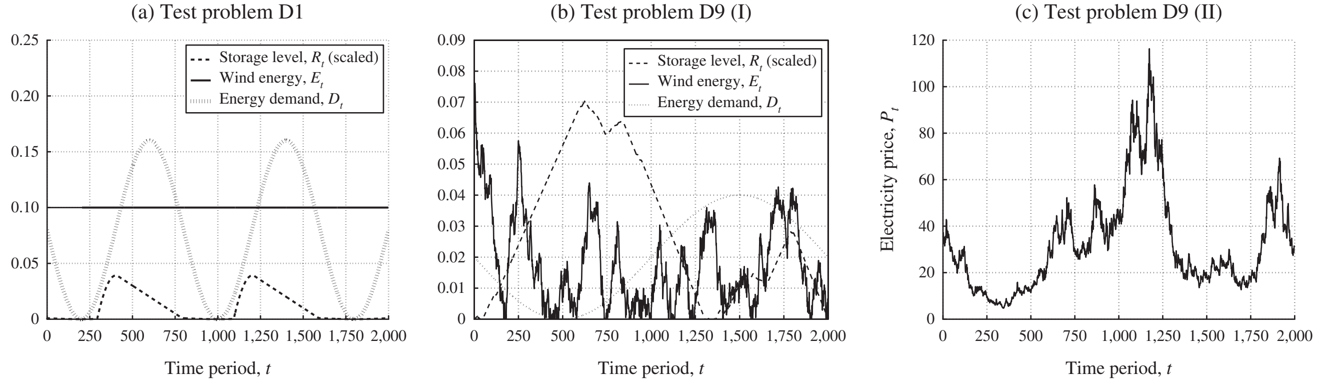
$$V_t^*(S_t) = \max_{x_t \in \mathscr{X}_t}\left(C(S_t, x_t) + \sum_{s' \in \mathscr{S}} \mathbb{P}_t(s' \mid S_t, x_t) V^*_{t+\Delta t}(s')\right), \quad \forall\, t, \in \mathscr{T}, \tag{33}$$

where we replaced the conditional expectation in (8) by the time-dependent conditional transition probability $\mathbb{P}_t(s' \mid S_t, x_t)$ of going from state S_t to state s' given the decision x_t, and where we assume that $V_t^* = 0$. After solving (33) for all $S_t \in \mathscr{S}$, we can simulate the optimal policy π_t^* obtained by finding exact value functions $\{V_t^*\}_{t=0}^T$ from solving (33). Note that these are the value functions around the predecision state. For a given sample path $\omega \in \Omega$, we simulate the MDP using the following policy:

$$\begin{aligned} &X_t^{\pi^*}(S_t(\omega)) \\ &\quad = \arg\max_{x_t \in \mathscr{X}_t}\left(C(S_t(\omega), x_t) + \sum_{s' \in \mathscr{S}} \mathbb{P}_t(s' \mid S_t(\omega), x_t) V^*_{t+\Delta t}(s')\right), \\ &\qquad\qquad \forall\, t \in \mathscr{T}, \end{aligned} \tag{34}$$

with $S_{t+\Delta t}(\omega) = S^M(S_t(\omega), X_t^{\pi^*}(S_t(\omega)), W_{t+\Delta t}(\omega))$.

Figure 3. The Renewable Energy, Demand, and Energy Prices from Two of the Deterministic Test Problems, and the Corresponding Storage Profiles Obtained by ADP



Note. The storage profiles are scaled to fit.

We designed a set of test problems where prices and renewable energy evolve stochastically over time and we obtained the optimal solution by solving (33) and (34). We considered the control of a generic, perfectly efficient storage device with capacity $\kappa = 30$ and $\gamma^c = \gamma^d = 5$ over the finite-time horizon $t = 0, \Delta t, 2\Delta t, \ldots, T$, where $T = 100$ and $\Delta t = 1$. The round-trip efficiency was chosen to be 1 to satisfy the fixed discretization of the state and decision variables.

For the stochastic storage problems, we considered the state variable $S_t = (R_t, E_t, D_t, P_t)$ is four dimensional. If each dimension were discretized uniformly into d states, we would have d^4 possible states at each t, which would become intractable even for relatively small values of d. The maximum number of feasible points in the decision space at each state and time also increases rapidly as the discretization gets finer. To keep the problem computationally tractable, we discretized all variables fairly coarsely. After discretization, the state space consists of either 5,551 or 8,897 states per time period, depending on the problem. We assume that the demand is time dependent but deterministic, which means that we do not need to include it in the state variable since time is implicitly accounted for.

The set of test problems was designed so that each problem may be solved exactly as an MDP in at most one week on a 2.26 GHz machine with 1TB RAM. For each test problem, the range of each of the variables and the corresponding mesh sizes, ΔR and ΔE, are shown in Table 4. Note that $\Delta P = 1$ for problems S5–S21. The transition dynamics used in these simulations are given below. For each test problem, we simulated $K = 256$ different sample paths, $\{\omega^1, \ldots, \omega^{\tilde{N}}\} = \bar{\Omega} \subset \Omega$, and then calculated a statistical estimate of the value of the optimal policy:

$$\bar{F}^* = \frac{1}{K} \sum_{k=1}^{K} \sum_{t \in \mathcal{T}} C\big(S_t(\omega^k), X_t^{\pi^*}(S_t(\omega^k))\big).$$

Before presenting the transition dynamics for the set of discretized stochastic experiments, we define two probability distributions.

The Discrete Uniform Distribution: We let $\mathcal{U}(a,b)$ for $a, b \in \mathbb{R}$ be the uniform distribution, which defines the evolution of a discrete random variable X with mesh size ΔX. Then, each element in $\mathcal{X} = \{a, a + \Delta X, a + 2\Delta X, \ldots, b - \Delta X, b\}$ has the same probability of occurring. The probability mass function is given by

$$u_X(x) = \frac{\Delta X}{b - a + \Delta X}, \quad \forall\, x \in \mathcal{X}.$$

The Discrete Pseudonormal Distribution: Let X be a normally distributed random variable and let $f_X(x; \mu_X, \sigma_X^2)$ be the normal probability density function with mean μ_X and variance σ_X^2. We define a discrete pseudonormal probability mass function for a discrete random variable $\bar{X}$ with support $\mathcal{X} = \{a, a + \Delta X, a + 2\Delta X, \ldots, b - \Delta X, b\}$ as follows, where $a, b \in \mathbb{R}$ are given and ΔX is the mesh size. For $x_i \in \mathcal{X}$, we let

$$g_{\bar{X}}(x_i; \mu, \sigma^2) = \frac{f_X(x_i; \mu_X, \sigma_X^2)}{\sum_{x_j=0}^{|\mathcal{X}|} f_X(x_j; \mu_X, \sigma_X^2)}$$

be the probability mass function corresponding to the discrete pseudonormal distribution. For the purpose of benchmarking only, we say that $\bar{X} \sim \mathcal{N}(\mu_X, \sigma_X^2)$ if $\bar{X}$ is distributed according to the discrete pseudonormal distribution. We recognize this is not standard practice but it simplifies the notation in this chapter.

The different transition dynamics in our test problems are described by the following stochastic processes.

The Renewable Energy Supply: The renewable supply process E_t is modeled using a bounded first-order Markov chain:

$$E_t = E_t + \hat{E}_{t+\Delta t}, \quad \forall\, t \in \mathcal{T} \setminus \{T\},$$

Table 4. Stochastic Test Problems

	Resource, R_t		Wind, E_t			Price, P_t		
Label	Levels	ΔR	Levels	ΔE	$\hat{E}_t$	Levels	Process	$\hat{P}_{0,t}$
S1	61	0.50	13	0.50	$\mathcal{U}(-1,1)$	7	Sinusoidal	$\mathcal{N}(0,25^2)$
S2	61	0.50	13	0.50	$\mathcal{N}(0,0.5^2)$	7	Sinusoidal	$\mathcal{N}(0,25^2)$
S3	61	0.50	13	0.50	$\mathcal{N}(0,1.0^2)$	7	Sinusoidal	$\mathcal{N}(0,25^2)$
S4	61	0.50	13	0.50	$\mathcal{N}(0,1.5^2)$	7	Sinusoidal	$\mathcal{N}(0,25^2)$
S5	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	first-order + jump	$\mathcal{N}(0,0.5^2)$
S6	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	first-order + jump	$\mathcal{N}(0,1.0^2)$
S7	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	first-order + jump	$\mathcal{N}(0,2.5^2)$
S8	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	first-order + jump	$\mathcal{N}(0,5.0^2)$
S9	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	first-order + jump	$\mathcal{N}(0,5.0^2)$
S10	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	first-order + jump	$\mathcal{N}(0,5.0^2)$
S11	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	first-order + jump	$\mathcal{N}(0,5.0^2)$
S12	31	1.00	7	1.00	$\mathcal{N}(0,2.0^2)$	41	first-order + jump	$\mathcal{N}(0,5.0^2)$
S13	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	first-order + jump	$\mathcal{N}(0,1.0^2)$
S14	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	first-order + jump	$\mathcal{N}(0,1.0^2)$
S15	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	first-order + jump	$\mathcal{N}(0,1.0^2)$
S16	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	first-order	$\mathcal{N}(0,1.0^2)$
S17	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	first-order	$\mathcal{N}(0,1.0^2)$
S18	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	first-order	$\mathcal{N}(0,1.0^2)$
S19	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	first-order	$\mathcal{N}(0,5.0^2)$
S20	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	first-order	$\mathcal{N}(0,5.0^2)$
S21	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	first-order	$\mathcal{N}(0,5.0^2)$

Note. The stochastic benchmark data sets, available at http://www.castlelab.princeton.edu/datasets.htm.

such that $E^{\min} \le E_t \le E^{\max}$, and where $\hat{E}_t$ is either pseudonormally or uniformly distributed (see Table 4). The bounds $E^{\min} = 1.00$ and $E^{\max} = 7.00$ were fixed for all problem instances.

The Electricity Prices: We test two different stochastic processes for P_t:

Sinusoidal:

$$P_t = \mu_t^P + \hat{P}_{0,t}, \qquad \forall\, t \in \mathcal{T}\setminus\{T\},$$

where $\mu_t^P = \mu_0 - A_P \sin(5\pi t/(2T))$ and $\hat{P}_{0,t} \sim \mathcal{N}(\mu_P, \sigma_P^2)$, where μ_0 and A_P are fixed.

Stationary first-order Markov chain with jumps

$$P_t = P_t + \hat{P}_{0,t} + \mathbf{1}_{\{u_t \le p\}} \hat{P}_{1,t}, \qquad \forall\, t \in \mathcal{T}\setminus\{T\},$$

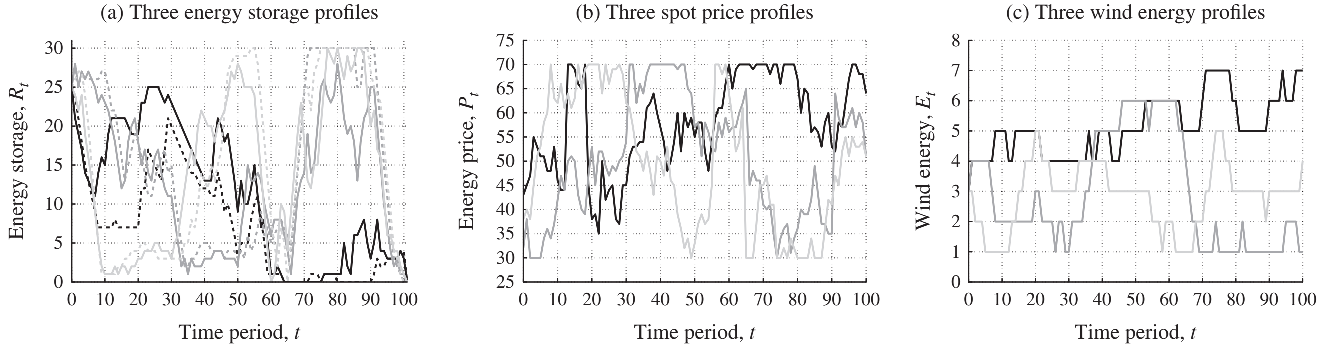
such that $P^{\min} \le P_t \le P^{\max}$, and where $\hat{P}_{0,t}$ is either pseudonormally or uniformly distributed as indicated in Table 4. We let $u_t \sim \mathcal{U}(0,1)$, and $p = 0.03$ for problems where jumps may occur or $p = 0$ otherwise. We let $\hat{P}_{1,t} \sim \mathcal{N}(0, 50^2)$ for all problems. The bounds $P^{\min} = 30.00$ and $P^{\max} = 70.00$ were fixed for all problem instances.

The Demand: The demand is assumed to be deterministic and given by

$$D_t = \left\lfloor \max\left[0, \mu_D - A_D \sin\left(\frac{2\pi t}{T}\right)\right]\right\rfloor,$$

where μ_D and A_D are fixed, and $\lfloor\cdot\rfloor$ is the floor function.

Figure 4. Results and Sample Paths from Test Problem S9



Notes. The storage profiles in Figure 4(a) were generated with the energy prices and renewable energy profiles of the same shade of grey in Figures 4(b) and 4(c).

8.4. Numerical Results

As explained in Section 7, even though the SPAR Storage algorithm is asymptotically optimal, convergence can be extremely slow for the temporally fine-grained problems that arise in battery storage. The BPTT algorithm, which does not enjoy a convergence proof, provides a significant improvement in numerical performance. In this section, we compare the value of the approximate policy generated by these two algorithms over the iterations.

We define the metric $\mathbf{F}^n$ to evaluate the performance of each algorithm

$$\mathbf{F}^n = \frac{\bar{F}^n}{\bar{F}^*},$$

and the corresponding standard error s^n after n iterations. From the results shown in Table 5 for the BPTT algorithm, it is evident that the mean value of the approximate policy is very close to optimal. For all test problems, the BPTT algorithm was capable of learning a policy that is within 0.86% of optimal in the worst case (test problem S5) and within 0.19% in the best case (test problem S17).

As expected, the level of optimality and the rate of convergence of approximate value iteration were lower than those of BPTT for all test problems. We illustrate this in Figure 5, where we show convergence plots comparing the performance of both algorithms on test problems S1, S4, S11, and S20. Approximate value iteration performed well in problems S17–S19 (99.67%, 99.65%, 99.65%, respectively) but failed to be less than 1% from optimal in all other problems. In particular, its worst performance was 1.54% from optimal in problem S10 (compared to 0.66% for BPTT). We claim that the value added by BPTT over approximate value iteration would be even greater for other problems with finer discretizations of time.

9. Application: The Value of a Multidimensional Storage System

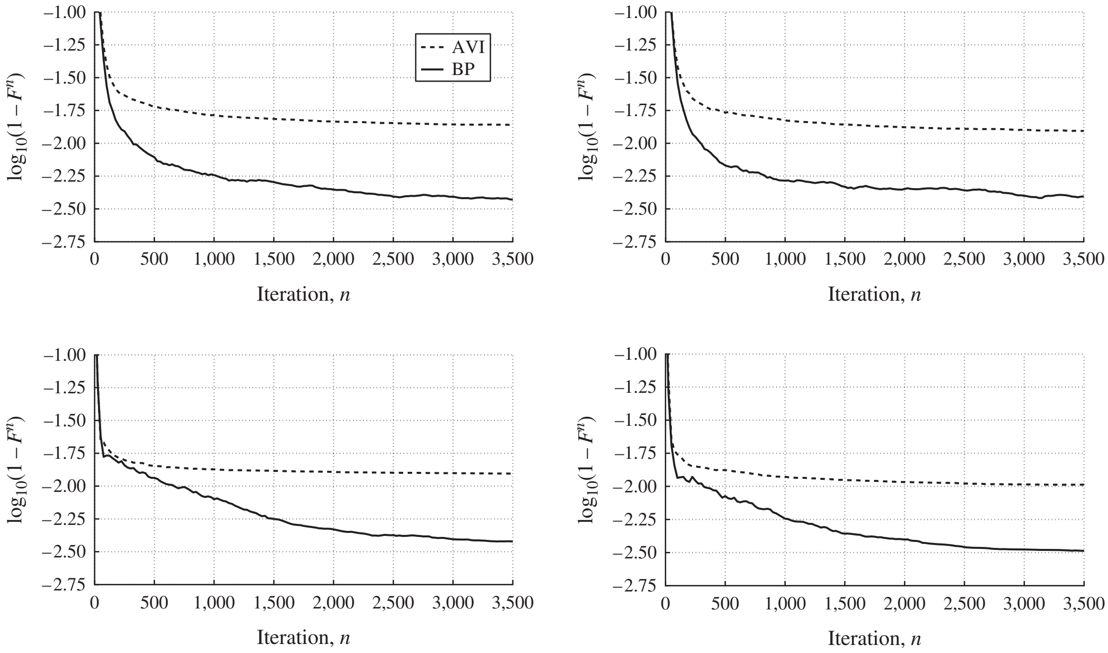
In this section, we use BPTT to solve an illustrative control problem with an M-dimensional portfolio of devices. We first consider a grid-level energy storage system consisting of two storage devices, a less expensive primary device with higher energy capacity but lower efficiency such as a lead-acid or lithium-ion battery, and a more expensive secondary device with lower energy capacity but with higher power capacity and efficiency such as an ultracapacitor. Alternatively, the one with higher energy capacity could be pumped hydro storage, while the smaller one could be any type of battery. We expect the best policy to be to use the larger device for low-frequency variations and to use the smaller low-loss device just for high-frequency variations. This is a fairly sophisticated behavior, and we are interested in seeing if our ADP approximation is able to produce it.

We quantify the value of the low-loss secondary device, which may increase the profit stream due to greater time-shifting capabilities that arise from the flexibility in storage options, or simply due to the increased energy capacity of the whole system. For example, this device may be useful in capturing the energy in high-power, short-lived wind gusts, while the device with the higher losses but lower power capacity can be reserved for use in lower frequency situations.

Table 5. Stochastic Benchmarking Results Using Backpropagation

Label	$\bar{F}^*$	$\mathbf{F}^0$ (%)	$\mathbf{F}^{50}$ (%)	$\mathbf{F}^{250}$ (%)	$\mathbf{F}^{500}$ (%)	$\mathbf{F}^{1,500}$ (%)	$\mathbf{F}^{3,000}$ (%)	$\mathbf{F}^{3,500}$ (%)	$s^{3,500}$ (%)
S1	19,480.11	44.88	91.87	98.77	99.22	99.49	99.61	99.63	±0.25
S2	18,915.05	44.60	90.37	98.63	99.16	99.32	99.47	99.52	±0.22
S3	19,730.28	44.63	90.70	98.83	99.20	99.49	99.59	99.61	±0.25
S4	19,831.66	44.20	90.70	98.88	99.32	99.53	99.60	99.61	±0.65
S5	16,806.49	57.24	97.41	98.92	99.02	98.98	99.14	99.14	±0.32
S6	17,089.29	57.17	97.14	98.78	98.84	99.18	99.55	99.58	±0.33
S7	18,104.07	56.77	97.46	98.69	98.78	99.31	99.57	99.59	±0.34
S8	19,011.24	55.85	97.72	98.65	99.02	99.42	99.58	99.59	±0.34
S9	17,963.79	57.54	96.99	98.74	98.97	99.36	99.43	99.44	±0.35
S10	19,079.16	55.88	97.91	98.85	99.02	99.42	99.55	99.56	±0.34
S11	19,396.01	54.90	97.60	98.56	98.85	99.44	99.61	99.62	±0.33
S12	19,500.51	54.35	97.95	98.90	98.96	99.44	99.60	99.62	±0.33
S13	16,547.09	59.35	97.22	98.53	98.99	99.27	99.35	99.36	±0.35
S14	17,700.78	57.34	97.73	98.42	98.86	99.30	99.50	99.51	±0.35
S15	17,972.71	56.27	96.92	98.63	98.82	99.42	99.58	99.58	±0.34
S16	14,113.12	59.92	98.65	99.44	99.65	99.76	99.78	99.78	±0.30
S17	15,115.53	57.54	98.19	99.26	99.46	99.74	99.80	99.81	±0.07
S18	15,377.29	56.42	98.57	99.37	99.57	99.72	99.79	99.80	±0.08
S19	17,467.27	58.58	96.91	98.65	98.99	98.40	99.50	99.50	±0.34
S20	18,601.82	56.72	97.95	98.89	99.16	99.56	99.67	99.67	±0.12
S21	18,912.27	55.74	97.67	98.78	99.00	99.47	99.58	99.58	±0.11

Figure 5. Comparison of Numerical Performance between AVI and Backpropagation (BP) on Test Problems S1, S4, S11, and S20 (from Top Left to Bottom Right)



Let devices 1 and 2 be two distinct energy storage devices. Any variable indexed by 1 or 2 represents the component of that variable, which corresponds to devices 1 and 2, respectively. We let $\eta_1^c\eta_1^d < \eta_2^c\eta_2^d$, $\gamma_1^c < \gamma_2^c$, and $R_2^c \ll R_1^c$, and we define the state variable as $S_t = (R_{t1}, R_{t2}, E_t, D_t, P_t)$. We consider the transition dynamics given by

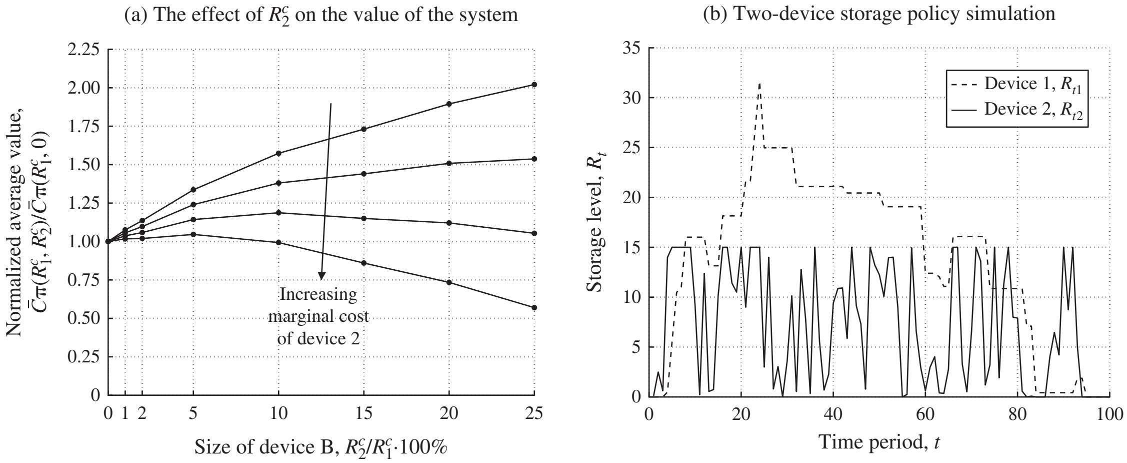
$$R_{t+\Delta t} = R_t + \Phi x_t,$$
$$E_{t+\Delta t} = \theta(z_{t+\Delta t})^6 = \theta(\varphi z_t + \varepsilon_{t+\Delta t}^E + c)^6,$$
$$P_{t+\Delta t} = P_t + J_{t+\Delta t} + \lambda(\mu_P - P_t) + \varepsilon_{t+\Delta t}^P.$$

We use an AR(1) model for the square root of the wind speed at time t, $z_t = \sqrt{\text{wind speed}}$, where φ is the autoregression coefficient and c is a constant. From an energy balance perspective, it is known that the wind energy is proportional to the cube of the wind speed, for some proportionality constant θ. The price is modeled as a mean-reverting first-order Markov chain with mean-reversion rate λ, equilibirum price μ_P, and normally distributed jumps J_t. We assume that the noise terms are normally distributed, $\varepsilon_t^E \sim \mathcal{N}(0, \sigma_E^2)$ and $\varepsilon_t^P \sim \mathcal{N}(0, \sigma_P^2)$. We consider the case of pure arbitrage, i.e., $D_t = 0 \; \forall \, t$.

The value of the energy storage system is given by

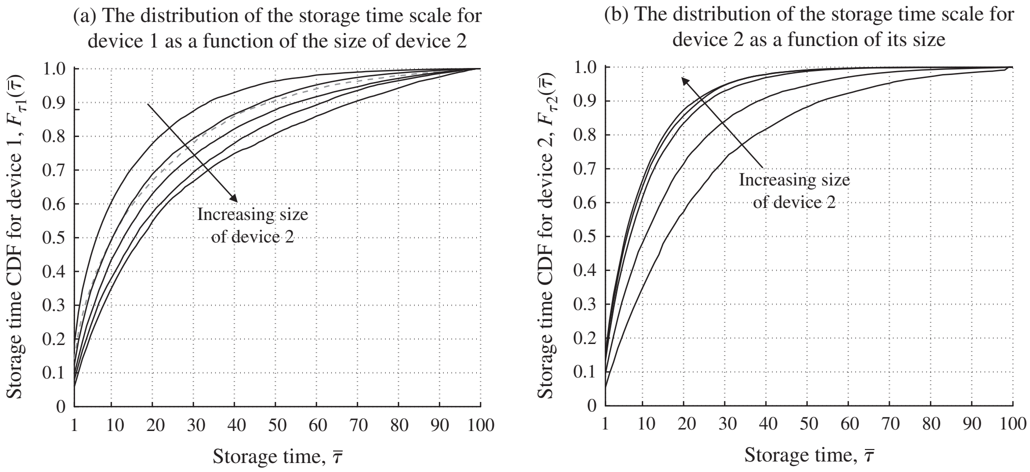
$$C^\pi(R_1^c, R_2^c) = \sum_{t \in \mathcal{T}} C(S_t, X_t^\pi(S_t)) - \mathrm{c}_1(R_1^c) - \mathrm{c}_2(R_2^c),$$

Figure 6. The Effect of Storage Energy Capacity on the Value of a Multidevice System


(a) The effect of R_2^c on the value of the system

(b) Two-device storage policy simulation

Figure 7. The Effect of Storage Energy Capacity on Storage Timescales


(a) The distribution of the storage time scale for device 1 as a function of the size of device 2

(b) The distribution of the storage time scale for device 2 as a function of its size

Notes. In Figure 7(a), the dashed line represents the $F_{\tau_1}(\cdot)$ for the one-device system. In 7(a) and 7(b), the arrow indicates increasing the size of device 2, R_2^c, from 2.5% to 20% of the size of device 1.

where π is the policy given by $\{\bar{V}_t^N(\cdot)\}_{t\in\mathcal{T}}$, and c_1 and c_2 are the marginal capital costs of storage of types 1 and 2, respectively, per time period. An estimate of C^π obtained with K sample paths is denoted $\bar{C}^\pi$.

In Figure 6(a), we show the relationship between the value of the energy storage system as a function of the size of device 2 (its energy capacity, R_2^c) and its marginal cost, normalized by the value of the one-device system composed solely of device 1. This type of analysis can be used for optimally sizing a hybrid energy storage system composed of more than one device. We expect that the increase in the value of the system seen in Figure 6(a) is not only due to increased capacity but also due to the time-shifting capabilities of the storage system. To confirm this, we define τ_i to be the amount of time that a nonzero amount of energy remains in device i (equivalently, the time between periods where the device is empty) and we look at its distribution.

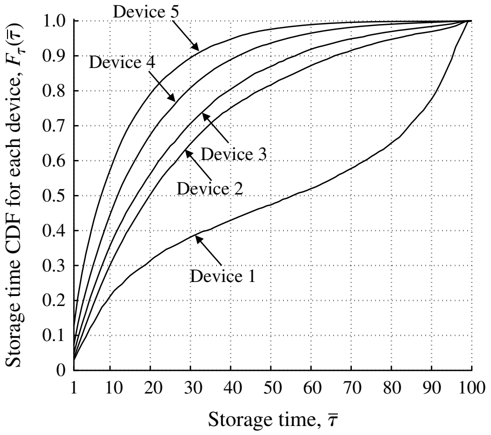
Figure 6(b) shows the storage profiles for devices 1 and 2 generated by simulating π on a single sample path. It is visually evident that devices 1 and 2 operate on two distinct time scales: the frequency of storage in device 2 is much higher than in device 1. In fact, device 2 is completely emptied out several times over the horizon while device 1 is completely emptied out twice. By analyzing the storage profiles over many sample paths, we may obtain an empirical cumulative distribution function for τ_i given by $F_{\tau_i}(\bar{\tau}) = \text{Prob}\{\tau_i \le \bar{\tau}\}$ for $i = 1, 2$.

In Figures 7(a) and 7(b), we show plots of $F_{\tau_1}(\cdot)$ and $F_{\tau_2}(\cdot)$ as a function of the size of device 2, while keeping the size of device 1 constant. It is clear that as the size of device 2 increases, the use of device 1 for higher frequency storage substantially decreases, and that device 2 is only used for high-frequency storage, regardless of its size. This indicates that the value of the storage system is affected significantly by shifting of the storage frequencies and not simply by added capacity.

Finally, we tested the algorithm on a portfolio consisting of five devices such that a device with higher energy capacity had lower power capacity and lower efficiency. Figure 8 shows $F_{\tau_i}(\cdot)$ for $i = 1, 2, 3, 4, 5$. With a storage frequency analysis like the one done for the two-dimensional portfolio, we can determine that the algorithm learns to shift the storage time scales to optimize profit, thanks to increased flexibility in storage options.

The methodology is not limited to 5 or 10 devices. With each additional storage device, we simply have to add another piecewise linear function. The resulting decision problem is nothing more than a slightly larger LP, which grows linearly with the number of

Figure 8. The Distribution of the Storage Timescale for Each Device in the Five-Device Energy System



devices. Computationally, we would have no difficulty handling hundreds or even thousands of devices, as might arise with distributed storage and more complex storage networks.

10. Conclusion

In this paper, we presented in detail an algorithm based on the method of BP that can be used to analyze complex multidimensional time-dependent energy storage problems. We benchmarked the algorithm against the optimal solution on a library of deterministic and stochastic problems with one storage device. We found that the algorithm was able to design time-dependent control policies that are within 0.08% of optimal in deterministic problems and within 1.34% in the stochastic ones. For the large majority of test problems, the policies generated by the algorithm had higher values than those obtained using model predictive control. We then used the algorithm to analyze a dual-storage system consisting of storage devices with different energy capacity, power capacity, and efficiency, and showed that the ADP policy properly uses the low-loss device (which is typically much more expensive) for high-frequency variations. We close by demonstrating the ability of algorithm to learn similar complex behavior for a five-device system. We emphasize that the algorithm easily scales to handling hundreds of devices since the size of the decision problem grows linearly with the number of devices, making it useful for the analysis of distributed storage systems and for more complex storage networks.

Acknowledgments

The authors thank Belgacem Bouzaïene-Ayari for his technical help. The paper also benefited from the careful and constructive comments of two reviewers.

References

Baert D, Vervaet A (1999) Lead-acid battery model for the derivation of Peukert's law. *Electrochimica Acta* 44(20):14.

Barton JP, Infield DG (2004) Intermittent renewable energy. *IEEE Trans. Energy Conversion* 19:441–448.

Beaudin M, Zareipour H, Schellenberglabe A, Rosehart W (2010) Energy storage for mitigating the variability of renewable electricity sources: An updated review. *Energy Sustainable Development* 14:302–314.

Bertsekas DP (2012) *Dynamic Programming and Optimal Control*, Approximate Dynamic Programming, 4th ed., Vol. II (Athena Scientific, Belmont, MA).

Bertsekas DP, Tsitsiklis JN (1996) Neuro-dynamic programming. *Neuro-Dynamic Programming*, Chap. 5 (Athena Scientific, Belmont, CA), 179–253.

Black M, Strbac G (2006) Value of storage in providing balancing services for electricity generation systems with high wind penetration. *J. Power Sources* 162:949–953.

Boyd SP, Vandenberghe L (2004) *Convex Optimization* (Cambridge University Press, Cambridge, UK).

Castronuovo ED (2004) On the optimization of the daily operation of a wind-hydro power plant. *IEEE Trans. Power Systems* 19(3):1–8.

Costa LM, Bourry F, Juban J, Kariniotakis G (2008) Management of energy storage coordinated with wind power under electricity market conditions. *Proc. 10th Internat. Conf. Probabilistic Methods Appl. Power Systems*, 1–8.

Dell R, Rand DAJ (2001) *Understanding Batteries*, 1st ed. (Royal Society of Chemistry, Cambridge, UK).

Dicorato M, Forte G, Pisani M, Trovato M (2012) Planning and operating combined wind-storage system in electricity market. *IEEE Trans. Sustainable Energy* 3:209–217.

Eyer JM, Iannucci JJ, Corey GP (2004) Energy Storage Benefits and Market Analysis Handbook, A Study for the DOE Energy Storage Systems Program. Technical report SAND2004-6177, Sandia National Laboratories, Washington, DC.

Fleten S-E, Kristoffersen TK (2007) Stochastic programming for optimizing bidding strategies of a Nordic hydropower producer. *Eur. J. Oper. Res.* 181:916–928.

George AP, Powell WB (2006a) Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *J. Machine Learn.* 65:167–198.

George AP, Powell WB (2006b) Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine Learn.* 65:167–198.

Gjerden KS, Helseth A, Mo B, Warland G (2015) Hydrothermal scheduling in Norway using stochastic dual dynamic programming; a large-scale case study. *PowerTech.* (IEEE, Eindhoven, Netherlands), 1–6.

Godfrey GA, Powell WB (2001) An adaptive, distribution-free algorithm for the newsvendor problem with censored demands, with applications to inventory and distribution. *Management Sci.* 47(8):1101–1112.

Granville S, Oliveira GC, Thome LM, Campodonico N, Latorre ML, Pereira MVF, Barroso LA (2003) Stochastic optimization of transmission constrained and large scale hydrothermal systems in a competitive framework. 2003 *IEEE Power Engrg. Soc. General Meeting IEEE Cat No*03*CH*37491, Vol. 2 (IEEE, Piscataway, NJ), 1101–1106.

Hastie T, Tibshirani R, Friedman J (2009) *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Corrected ed. (Springer, New York).

Kraining M, Wang Y, Akuiyibo E, Boyd S (2011) Operation and configuration of a storage portfolio via convex optimization. *Proc. IFAC World Congress*, 10487–10492.

Kuperman A, Aharon I (2011) Battery ultracapacitor hybrids for pulsed current loads: A review. *Renewable Sustainable Energy Rev.* 15:981–992.

Lohndorf N, Minner S (2010) Optimal day-ahead trading and storage of renewable energies an approximate dynamic programming approach. *Energy Systems* 1:1–17.

Lohndorf N, Wozabal D, Minner S (2013) Optimizing trading decisions for hydro storage systems using approximate dual dynamic programming optimizing trading decisions for hydro storage systems using approximate dual dynamic programming. *Oper. Res.* 61(4):810–823.

Mokrian P, Stephen M (2006) A stochastic programming framework for the valuation of electricity storage. 26*th USAEE/IAEE North Amer. Conf.*, 24–27.

Nascimento JM, Powell WB (2013) An optimal approximate dynamic programming algorithm for concave, scalar storage problems with vector-valued controls. *IEEE Trans. Automatic Control* 58:2995–3010.

Philpott AB, de Matos VL (2012) Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *Eur. J. Oper. Res.* 218:470–483.

Powell WB (2011a) *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd ed. (John Wiley & Sons, Hoboken, NJ).

Powell WB (2011b) *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd ed. (John Wiley & Sons, Hoboken, NJ).

Powell W, Ruszczyński A, Topaloglu H (2004) Learning algorithms for separable approximations of discrete stochastic optimization problems. *Math. Oper. Res.* 29(4):814–836.

Powell WB, George A, Simão H, Scott W, Lamont A, Stewart J (2012) SMART: A stochastic multiscale model for the analysis of energy resources, technology, and policy. *INFORMS J. Comput.* 24(4):665–682.

Pritchard G, Philpott AB, Neame PJ (2005) Hydroelectric reservoir optimization in a pool market. *Math. Programming* 103:445–461.

Robbins H, Monro S (1951) A stochastic approximation method. *Ann. Math. Statist.* 22:400–407.

Ru Y, Kleissl J, Martínez S (2013) Storage size determination for grid-connected photovoltaic systems. *IEEE Trans. Sustainable Energy* 4:68–81.

Shapiro A, Tekaya W, Paulo J, Pereira MVF (2013) Risk neutral and risk averse stochastic dual dynamic programming method. *Eur. J. Oper. Res.* 224:375–391.

Sioshansi R, Denholm P (2013) Benefits of colocating concentrating solar power and wind. *IEEE Trans. Sustainable Energy* 4(4): 877–885.

Sioshansi R, Hurlbut D (2010) Market protocols in ERCOT and their effect on wind generation. *Energy Policy* 38(7):3192–3197.

Sutton RS, Barto AG (1998) *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA).

Teleke S, Baran ME, Bhattacharya S, Huang AQ (2010) Rule-based control of battery energy storage for dispatching intermittent renewable sources. *IEEE Trans. Sustainable Energy* 1:117–124.

Topaloglu H, Powell WB (2005) A distributed decision-making structure for dynamic resource allocation using nonlinear functional approximations. *Oper. Res.* 53(2):281–297.

Topaloglu H, Powell WB (2006) Dynamic-programming approximations for stochastic time-staged integer multicommodity-flow problems. *INFORMS J. Comput.* 18(1):31–42.

Van Roy B, Bertsekas DP, Lee Y, Tsitsiklis JN (1997) A neuro-dynamic programming approach to retailer inventory management. *Proc. 36th IEEE Conf. Decision and Control*, Vol. 4 (IEEE, Piscataway, NJ), 4052–4057.

Vazquez S, Lukic SM, Galván E, Franquelo LG, Carrasco JM (2010) Energy storage systems for transport and grid applications. *IEEE Trans. Indust. Electronics* 57:3881–3895.

Werbos PJ (1974) Beyond regression: New tools for prediction and analysis in the behavioral sciences. Unpublished doctoral dissertation, Harvard University, Cambridge, MA.

Werbos PJ (1989) Backpropagation and neurocontrol: A review and prospectus. *Internat. Joint Conf. Neural Networks* (IEEE, Piscataway, NJ), 209–216.

Werbos PJ (1990) A menu of designs for reinforcement learning over time. Miller WT, Sutton RS, Werbos PJ, eds. *Neural Networks* (MIT Press, Cambridge, MA), 67–95.

Werbos PJ (1992) Approximate dynamic programming for real-time control and neural modelling. White DA, Sofge DA, eds. *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches* (Van Norstrand Reinhold, New York), 493–525.

Xi X, Sioshansi R, Marano V (2013) A stochastic dynamic programming model for co-optimization of distributed energy storage. *Energy Systems* 5:475–505.

Zhou H (2013) New energy storage devices for post lithium-ion batteries. *Energy Environ. Sci.* 6:2256.

Zhou Y, Scheller-Wolf A, Secomandi N, Smith S (2013) Is it more valuable to store or destroy electricity surpluses? Available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2126302.