Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Multidimensional Energy Storage Problems

Daniel F. Salas[1], Warren B. Powell[2]

[1]Department of Chemical & Biological Engineering
[2]Department of Operations Research & Financial Engineering
Princeton University, Princeton, NJ 08544

Abstract

We present and benchmark an approximate dynamic programming algorithm that is capable of designing near-optimal control policies for time-dependent, finite-horizon energy storage problems, 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 in deterministic comparisons and within 1.34% in stochastic ones, much lower than those obtained using model predictive control. 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 distriibuted over the grid and more complex storage networks.

1 Introduction

Increasing interest in renewable energy over the past two decades has led to active research and development of technologies for directly harvesting wind energy. Energy storage has received significant attention as a way to increase the efficiency of the electrical grid by smoothing energy fluctuations. In particular, wind energy storage has been proposed as a strategy for reducing curtailment, time-shifting, spinning reserve, peak shaving and electricity price arbitrage, as highlighted in Dell & Rand (2001), Barton & Infield (2004), Eyer et al. (2004), Black & Strbac (2006), Beaudin et al. (2010), Zhou et al. (2013) and Xi et al. (2013).

However, the intermittent and uncertain nature of wind energy, which arrives at rates that vary stochastically over time, presents a fundamental challenge to the integration of storage into the electricity grid as penetration levels rise over the next several years (Bird & Lew 2012). Furthermore, despite aggressive government and private sector efforts to develop more economical electrochemical energy storage technologies, storage solutions for grid-level applications remain too expensive for widespread commercialization. For these reasons, practical and adaptive methods for the optimal control of such power systems are critical for the deployment of reliable and more economical energy storage systems.

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 & Lopes (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) presents 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 & Stephen (2006), a stochastic programming algorithm is used to operate a storage device for energy arbitrage. Fleten & Kristoffersen (2007) accounts for uncertainty in spot market prices in a stochastic programming-based bidding model but it ignores the stochasticity of the energy supply. In Löhndorf & Minner (2010), an approximate dynamic programming (ADP) least-squares policy evaluation approach based on temporal differences (LSTD) 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. An optimal switching approach for the problem of storage control and valuation is presented in Carmona & Ludkovski (2010). In Lai et al. (2010), ADP is used to bound the value of natural gas storage. The value of a storage device used for energy arbitrage in the PJM market is studied in Sioshansi et al. (2009). Zhou et al. (2012) studies the value of an energy disposal strategy in the presence of negative electricity prices for a standalone storage facility located at a market. In Kim & Powell (2011), the relationship between optimal energy commiments and the value of wind energy storage is studied for the infinite-horizon storage problem. Zhou et al. (2013) finds a three-tiered structure for the storage policy for a single storage device with a three-dimensional state variable. The authors demonstrate the near-optimality of a simple policy that requires determining three time-dependent parameters; using 15-minute increments, this means determining $3 \times 4 \times 24 = 288$ parameters to capture a daily cycle.

None of the papers above handle multiple storage devices in a coordinated way. Storage comes in different styles such as pumped-hydro, batteries, ultracapacitors, compressed air energy storage (CAES), flywheels, and even load shifting (which is a form of storage). Each type of storage device exhibits different losses, rated energy capacities and rated power capacities, and as a result is best suited to handle variations in specific frequency ranges. Low-loss devices that are more expensive and have lower energy capacities, such as ultracapacitors and lithium-ion batteries, are best suited for pulse power and high frequency fluctuations of smaller amplitude. Pumped-hydro and CAES, which generally incur higher losses, are better suited to handling power delivery over relatively longer timescales. We also note that our method handles temporally fine-grained, time-dependent policies, and has no difficulty handling problems with thousands of time periods.

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 wind supply, demand and electricity prices may evolve stochastically. We benchmark the algorithm against optimal policies when applied to a single storage device. This algorithm scales to systems composed of multiple devices, *e.g.* a single system with different types of devices or a network of storage devices. Recent research in vehicular electronics and energy systems has pointed out potential benefits of using energy systems composed of more than one type of device for grid-level storage (see Kraining et al. (2011), Vazquez et al. (2010) and Kuperman & Aharon (2011)). Our work is specifically aimed at grid-level storage, where we may have to control many devices, and where we have to deal with the complex dynamics of electricity spot prices.

The algorithm is based on literature that demonstrates the effectiveness of using separable, piecewise linear, convex value function approximations (VFAs) for high-dimensional stochastic resource allocation problems such as rail fleet management (Topaloglu & Powell 2005), replacement of high-voltage transformers (Enders et al. 2010, Enders & Powell 2010), trucking (Simao et al. 2008) and management of high-value spare parts (Simao & Powell 2009). In Powell et al. (2011), the SMART model, based on the framework of ADP, is proposed to solve an energy problem involving dispatch, storage and long-term investment decisions while the uncertainty in energy from wind, demands, prices and rainfall. The results from the algorithm were shown to be in close agreement with the optimal solution from a deterministic version of the problem, but comparisons to stochastic problems were not presented. Furthermore, only storage systems composed of a single device were modeled. Nascimento & Powell (2013) presents a convergence proof for an ADP algorithm based on piecewise linear and convex VFAs for a single energy storage device in the presence of random exogenous information. We use a similar algorithmic strategy which easily scales to multidimensional storage portfolios. However, in this paper we show that the algorithm works not only in theory (for a single storage device) but also in practice, with comparisons against optimal benchmarks (for single storage problems).

This paper makes the following contributions: 1) We lay out a mathematical model for energy storage that handles multidimensinal portfolios, and uncertainty in demand, electricity prices and wind energy supply; 2) 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; 3) we provide benchmarking 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; 4) 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); 5) we show the ability of our algorithm to design time-dependent control policies for complex storage problems for energy systems consisting of multiple devices. In particular, we show the sensitivity of the algorithm to the parameters of the system (like device efficiency and size) in the allocation of energy.

The rest of the paper is organized as follows. In section 2, we lay out the mathematical model of the storage problem. In section 3, we briefly describe the framework of model predictive control. In section 4, we explain the framework of ADP used to design the algorithm. In section 5, we describe in detail the main algorithm. In section 6, we benchmark the algorithm using a set of deterministic and stochastic problems. Finally, in section 7, we extend the application of the algorithm presented in section 5 to a problem involving two storage devices and we perform a simple analysis of the value added by the second device. Similarly, we present brief results for a five-dimensional portfolio of devices. In the process, we make the case that the method can be scaled to problems with multiple storage devices, since the decision problem is a linear program that scales linearly with the number of storage devices. Section 8 concludes the paper.

2 The Mathematical Model

We consider the problem of allocating energy to a single grid-level storage device over a finite-time horizon $t = 0, \Delta t, 2\Delta t, \dots, T$, where Δt is the time step, while maximizing the undiscounted total profit. The storage device may be co-located with a wind farm, but our model and algorithm can be applied to a general grid-level storage device. We can import from/export to the grid, in addition to satisfying a specific set of demands. We let $\mathcal{T} = \{0, \Delta t, 2\Delta t, \dots, T\}$.

Electricity may flow directly from the wind farm to the storage device 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.

2.1 Static Parameters

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

- R^c : The energy capacity of the device in MWh.
- η^c, η^d : The charging and discharging efficiency of the device, respectively.
- γ^c, γ^d : The maximum charging and discharging rates of the device, given as MWh per time period.
- c^h : The holding cost of the device, in \$ per MWh per time step.

2.2 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 amount of energy in the storage device at time t in MWh.
- E_t : The net amount of wind energy available at time t in MWh.
- D_t : The aggregate energy demand at time t, in MWh.
- P_t : The price of electricity at time t in the spot market, in \$/MWh.

2.3 The Decisions

At any point in time, the decision is given by the column vector $x_t = (x_t^{WD}, x_t^{GD}, x_t^{RD}, x_t^{WR}, x_t^{GR}, x_t^{RG})$, where x_t^{IJ} is the amount of energy transferred from I to J at time t. The superscript W stands for wind, D

for demand, R for storage and G for grid. Arguably, the main decision we must make is whether to withdraw energy from storage now or hold it for a later time when we anticipate the wind may die down or the electricity prices may go up.

2.4 The Constraints

In our model, we require that all components of x_t be nonnegative for all t. At any time t, we require that the total amount of energy stored in the device from the wind does not exceed the energy capacity available:

$$x_t^{WR} + x_t^{GR} \leq R^c - R_t. \tag{1}$$

We also make the assumption that all demand at time t must be satisfied at time t:

$$x_t^{WD} + \eta^d x_t^{RD} + x_t^{GD} = D_t. \tag{2}$$

Additionally, the amount withdrawn from the device at time t to satisfy demand plus any amount of energy sold to the grid after satisfying demand must not exceed the amount of energy that is available in the device when we make the decision to store or withdraw:

$$x_t^{RD} + x_t^{RG} \leq R_t. \tag{3}$$

The total amount of energy charged to or withdrawn from the device is also constrained by the maximum charging and discharging rates:

$$x_t^{WR} + x_t^{GR} \leq \gamma^c, \tag{4}$$

$$x_t^{RD} + x_t^{RG} \leq \gamma^d. \tag{5}$$

Finally, flow conservation requires that:

$$x_t^{WR} + x_t^{WD} \leq E_t. \tag{6}$$

The feasible action space, $\mathcal{X}_t$, is the convex set defined by (1)-(6). We let $X_t^\pi(S_t)$ be the decision function that returns $x_t \in \mathcal{X}_t$, where $\pi \in \Pi$ represents the type of policy (which we determine later).

Remarks:

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

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

• We place no constraints on the amount of electricity that can be imported from/exported to the grid. However, amounts typically exchanged are low enough to assume that we are price-takers and our decision has no impact on the electricity prices.

2.5 The Exogenous Information Process

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

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

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

To avoid violating the nonanticipativity condition, we assume that any variable that is indexed by t is $\mathcal{F}_t$-measurable. As a result, W_t is defined to be the information that becomes available between times $t - \Delta t$ and t. A sample realization of W_t is denoted $W_t^n = W_t(\omega^n)$ for sample path $\omega^n \in \Omega$.

2.6 The Transition Function

Also known as the system model, $S_{t+\Delta t} = S^M(S_t, x_t, W_{t+\Delta t})$ is a mapping from a state S_t to the next state $S_{t+\Delta t}$, given our decision x_t and new information $W_{t+\Delta t}$. The transition function for the energy in storage is given by:

$$R_{t+\Delta t} = R_t + \phi^T x_t,$$

where $\phi = (0, 0, -1, \eta^c, \eta^c, -1)$ is an incidence column vector that models the flow of energy into and out of the device. Throughout this paper, we use different transition dynamics for the wind, price and demand processes. They are presented sections 6.1, 6.2 and 7.

2.7 The Objective Function

The function $C(S_t, x_t)$ represents the contribution from being in the state S_t and making the decision x_t at time t. We are interested in maximizing profit 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:

$$C(S_t, x_t) = P_t D_t - P_t(x_t^{GR} - \eta^d x_t^{RG} + x_t^{GD}) - c^h \left(R_t + \phi^T x_t\right).$$

The objective function is then given by:

$$F^{\pi^*} = \max_{\pi \in \Pi} \mathbb{E}\Big[\sum_{t \in \mathcal{T}} C\big(S_t, X_t^{\pi}(S_t)\big)\Big], \tag{7}$$

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

We emphasize that our 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 below handles these fine-grained applications. Our numerical testing below is on problems with 2,000 time periods.

If the state variable evolves deterministically and the dynamics are known *a priori*, we can solve the control problem using a standard batch linear program (LP):

$$F^* = \max_{x_0,\cdots,x_T} \sum_{t\in\mathcal{T}} C(S_t, x_t), \tag{8}$$

such that $x_t \in \mathcal{X}_t$ 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, demand and price trends. However, this is hardly ever the case with physical processes that are intrinsically stochastic. We use this formulation in section 6.1 as a way to benchmark our approximate algorithm.

3 Model Predictive Control

A common advanced method for solving control problems is model predictive control, also known as receding horizon procedure or deterministic lookahead policy (see Camacho & Bordons (2004)). This method solves the control problem over a horizon H at time t using a point forecast of the future, then implements only the decision for time t and repeats the process at $t + \Delta t$, when new information has arrived.

At time t, we solve:

$$\max_{x_t,\cdots,x_{t+H}} \sum_{t'=t}^{\min(T,t+H)} C(S_{t'}, x_{t'}), \tag{9}$$

such that $x_t \in \mathcal{X}_t$ for all t, subject to transition dynamics implemented as constraints and to a forecast of the random variables over the horizon, $f_{t,H} = \{E_{t,t+m}, D_{t,t+m}, P_{t,t+m}\}_{m=1}^{H}$:

$$\begin{aligned} E_{t,t+m\Delta t} &= \mathbb{E}\left[E_{t+m\Delta t} \mid S_t\right], \\ D_{t,t+m\Delta t} &= \mathbb{E}\left[D_{t+m\Delta t} \mid S_t\right], \\ P_{t,t+m\Delta t} &= \mathbb{E}\left[P_{t+m\Delta t} \mid S_t\right]. \end{aligned} \tag{10}$$

We then implement the decision x_t, step forward in time and repeat the procedure at time $t + \Delta t$.

4 An Approximate Dynamic Programming Approach

In order to solve (7), we can define the policy determined by an optimal value function $V_t^*(S_t)$ (also known as the "cost-to-go" function), which is the maximum revenue we expect to receive from time t onwards if we are in state S_t. The optimal policy chooses the action which maximizes the value of being in a state.

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

$$V_t^*(S_t) = \max_{x_t\in\mathcal{X}_t} \left(C(S_t, x_t) + \mathbb{E}\left[V_{t+\Delta t}^*(S_{t+\Delta t}) \mid S_t, x_t\right]\right) \;\forall t \in \mathcal{T}, \tag{11}$$

the arg max of which gives the optimal 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 state, action and information spaces are all continuous and multi-dimensional. To overcome the curse of dimensionality in the outcome space, we redefine the value function around the *post-decision* 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 2011).

Using the post-decision state variable, we replace the expectation by a post-decision value function and reformulate (11) as a deterministic maximization problem:

$$V_t(S_t) = \max_{x_t \in \mathcal{X}_t} \left(C(S_t, x_t) + V_t^x(S_t^x) \right) \; \forall t \in \mathcal{T}, \tag{12}$$

where $V_t^x(S_t^x) = \mathbb{E}[V_{t+\Delta t}(S_{t+\Delta t}) \mid S_t, x_t]$. Solving (12) 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, so we would like to estimate it accurately in order to determine a storage policy which is as close to optimal as possible.

The post-decision state only includes variables which are required to compute the transition dynamics. In other words, a variable is part of the post-decision state if and only if its value at time $t + \Delta t$ is dependent on its value at time t. In our approach, we introduce a value function approximation around the post-decision state, $\overline{V}_t(S_t^x)$. There is a vast literature aimed at developing approximations of the value function (Puterman 1994, Bertsekas & Tsitsiklis 1996, Sutton & Barto 1998, Powell 2011). The three main types are lookup table, parametric and nonparametric approximations (Hastie et al. 2003).

It is easy to show that the value function is concave in the resource dimension (Vanderbei 2007). It is concave because the resource variable is a right-hand side constraint, and any maximizing LP is concave in right-hand side constraints. We introduce an approximation which is concave and piecewise linear in the resource dimension. For given values of E_t and P_t, we write this type of VFA as:

$$\overline{V}_t(R_t^x, E_t, P_t) = \sum_{i=1}^{K_t} \overline{v}_{ti}(E_t, P_t) r_{ti},$$

where $\sum_i r_{ti} = R_t^x$ and $0 \le r_{ti} \le \bar{r}_{ti}$ for every i. The variable r_{ti} is the resource coordinate variable for segment $i \in \{1, \ldots, K_t\}, K_t \in \mathbb{N}$, $\bar{r}_{ti}$ is the capacity of the segment and $\overline{v}_{ti}(E_t, P_t)$ is its slope.

One of the main advantages of this approximation strategy is that each VFA is completely determined by a set of slopes, $\mathfrak{U}_t(E_t, P_t) = \{\overline{v}_{ti}(E_t, P_t)\}_{i=1}^{K_t}$, and a corresponding set of breakpoints, $\mathfrak{B}_t = \{r_{ti}\}_{i=1}^{K_t}$. This essentially renders this approximation strategy as a structured lookup table which allows us to easily maintain concavity. 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 reinforcement learning policies for problems with discrete actions (Bertsekas & Tsitsiklis 1996, Sutton & Barto 1998, Powell 2011, Bertsekas 2012).

We use a simple aggregation method in order to handle the curse of dimensionality in the continuous wind and price dimensions of the post-decision state variable. We let $\mathcal{G}_E^{g_e}(\cdot)$ and $\mathcal{G}_P^{g_p}(\cdot)$ be functions which aggregate the wind and price dimensions, respectively, where $g_e, g_p \in \mathbb{N}$ determine the level of aggregation. We let

$\mathbf{g} = (g_e, g_p)$ be the aggregation multi-index and $S_t^{\mathbf{g},x} = \big(R_t^x, \mathcal{G}_E^{g_e}(E_t), \mathcal{G}_P^{g_p}(P_t)\big)$ be the aggregated post-decision state variable.

We then construct a VFA that is concave piecewise linear in the resource dimension for every $\big(\mathcal{G}_E^{g_e}(E_t), \mathcal{G}_P^{g_p}(P_t)\big)$-pair. For notational convinience, we simply write $\overline{V}_t(S_t^{\mathbf{g},x})$ as $\overline{V}_t(S_t^x)$, 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.

Introducing the VFA into (12) and letting $r_t = (r_{ti})_{i=1}^{K_t}$, we obtain our ADP formulation of (11):

$$\begin{aligned} X_t^\pi(S_t) = &\arg\max_{x_t \in \mathcal{X}_t, r_t} \Big(C(S_t, x_t) + \sum_{i=1}^{K_t} \overline{v}_{ti} r_{ti}\Big) \ \forall t \in \mathcal{T}, \\ \text{s.t.} \ &\sum_{i=1}^{K_t} r_{ti} - \phi^T x_t = R_t, \\ &0 \le r_{ti} \le \bar{r}_{ti} \ \forall i. \end{aligned} \tag{13}$$

We let $\mathcal{R}_t$ be the feasible convex set for r_t.

5 The Algorithm

We would like to solve (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 paper, 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 sets $\mathfrak{B}_t^n$ and $\mathfrak{U}_t^n$, these are the only sets we are required to keep track of. The set of breakpoints determines the coarseness of the domain of the VFA. The domain is the resource level which is a continuous variable, so a very fine discretization of the domain is appropriate.

To start, we initialize the VFA by letting it be zero everywhere, which is equivalent to setting $\mathfrak{U}_t^0 = \{0\}$ for every t, where the superscript $n = 0$ indicates the initialization. We also let $\mathfrak{B}_t^0 = \{0, \delta R, 2\delta R, \cdots, R^c - \delta R, R^c\}$ for some small $\delta R > 0$. We assume the level of discretization is constant for all t and all n, therefore we have $\mathfrak{B}_t^n \equiv \mathfrak{B}$.

At the beginning of iteration $n \ge 1$, we draw a random sample realization $\{W_1^n, ..., W_T^n\}$. Then we step forward through time. At time t, the optimal action is determined by solving (13) using the VFA from the previous iteration:

$$\begin{aligned} X_t^\pi(S_t^n) &= \operatorname*{arg\,max}_{x_t \in \mathcal{X}_t^n} \left(C(S_t^n, x_t) + \overline{V}_t^{n-1}(S_t^{x,n}) \right), \\ &= \operatorname*{arg\,max}_{\substack{x_t \in \mathcal{X}_t^n \\ r_t \in \mathcal{R}_t^n}} \left(C(S_t^n, x_t) + \sum_{i=1}^{K_t^{n-1}} \overline{v}_{ti}^{n-1} r_{ti} \right), \text{ where } \overline{v}_{ti}^{n-1} \in \mathfrak{U}_t^{n-1} \ \forall i. \end{aligned} \tag{14}$$

Since the VFA is determined by a set of slopes, what we are really after is an observation of the marginal value of energy in the storage device. At time t, we may obtain an observation of the marginal value, $\hat{v}_t^n(R_t^n)$, by solving:

$$\hat{v}_t^n(R_t^n) = \frac{\partial}{\partial R_t^n} \max_{x_t \in \mathcal{X}_t^n} \left(C(S_t^n, x_t) + \overline{V}_t^{n-1}(S_t^{x,n}) \right). \tag{15}$$

Using the observation given by (15) to update the VFA is a form of classical value iteration for a finite horizon problem as used by Nascimento & Powell (2013). An important limitation of this approach is that the rate of convergence of the algorithm may decrease significantly, especially 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. For example, we may decide to store energy at 10am but the marginal impact is not felt until 3pm, which may be hundreds of time periods in the future. This is because as $\hat{v}_t^n$ is smoothed into $\overline{v}_t^{n-1}$, the smoothing parameter plays a discounting role. Instead, we compute an observation of the slope using a double pass algorithm which is known in the reinforcement learning community as *temporal difference learning* with discount factor $\lambda = 1$, or TD(1) (Sutton & Barto 1998).

In the forward pass, at time t we compute an observation of the marginal contribution given by:

$$\hat{c}_t^n(S_t^n) = \frac{\partial}{\partial R_t^n} C\left(S_t^n, X_t^\pi(S_t^n)\right). \tag{16}$$

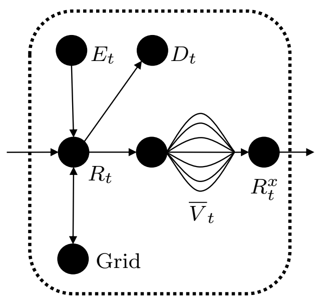
For differentiable value functions, $\hat{c}_t^n$ can be obtained from the dual solution to (14). However, the VFA may be nondifferentiable at any breakpoint and at such a point there exists both a right and a left derivative. The dual solution may be either of these values, or any value between these two, if $R_t^n \in \mathfrak{B}$. To overcome this, we approximate (16) with right and left numerical derivatives, which is quite fast computationally since it only requires resolving the warm-started linear program in (14).

We define $S_t^{n+} = (R_t^n + \delta R, E_t^n, D_t^n, P_t^n)$ and $x_t^{n+} = X_t^\pi(S_t^{n+})$, where δR is the mesh size of the resource domain. We define S_t^{n-} and x_t^{n-} analogously. We obtain observations of the right and left marginal contributions $\hat{c}_t^{n+}$ and $\hat{c}_t^{n-}$, respectively, as follows:

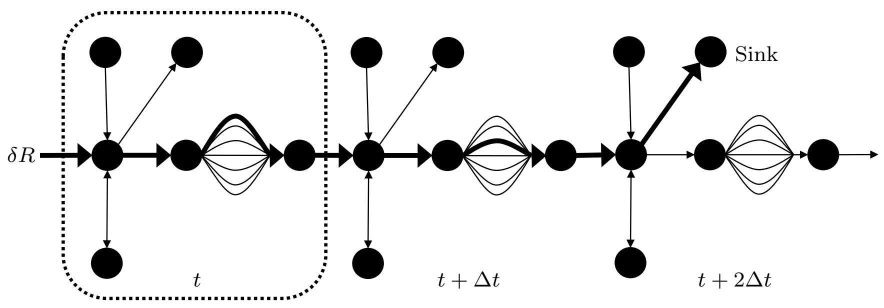
$$\hat{c}_t^{n+}(S_t^n) = \frac{C(S_t^{n+}, x_t^{n+}) - C(S_t^n, x_t^n)}{\delta R}, \tag{17}$$

$$\hat{c}_t^{n-}(S_t^n) = \frac{C(S_t^n, x_t^n) - C(S_t^{n-}, x_t^{n-})}{\delta R}. \tag{18}$$

The 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



(a) One-period flow network.



(b) Tracking the marginal value of energy through the multi-period network.

Figure 1: An illustration of the marginal value calculation for a single device. Bold arrows indicate increased flow due to the perturbation $+\delta R$.

in storage. In figure 1, we illustrate the computation of the marginal value. We let Δ_t^{n+} and Δ_t^{n-} be the variables which indicate whether or not there is a change in the amount of energy held in storage as a result of the perturbation, where $\Delta_t^{n+} = \frac{1}{\delta R}\phi^T(x_t^{n+} - x_t^n)$ and $\Delta_t^{n-} = \frac{1}{\delta R}\phi^T(x_t^{n-} - x_t^n)$. Note that $\Delta_t^{n+} > 0$ if and only if any of the incremental energy δR is held in storage. Similarly for $\Delta_t^{n-} < 0$.

After recording the observation of the marginal contribution, 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 $S_{t+\Delta t}^n = S^M(S_t^n, x_t^n, W_{t+\Delta t}^n)$. Once we have reached the end of the horizon, we then sweep backwards in time computing observations of the slopes that we can use to update the VFA. We compute:

$$\hat{v}_t^{n+}(S_t^n) = \begin{cases} \hat{c}_t^{n+}(S_t^n) + \Delta_t^{n+}\hat{v}_{t+\Delta t}^{n+}(S_{t+\Delta t}^n) & \text{for } 0 \leq t < T, \\ \hat{c}_T^{n+}(S_T^n) & \text{otherwise,} \end{cases} \tag{19}$$

and similarly for $\hat{v}_t^{n-}$.

It is important to note that the observation of the slope made at S_t^n is also an observation of the slope at $S_{t-\Delta t}^{x,n}$, since the transition only involves the sampling of random variables, and for a given sample path we have that:

$$\begin{aligned} \hat{c}_{t-\Delta t}^n(S_{t-\Delta t}^{x,n}) &= \frac{\partial C(S_t^n, X_t^\pi(S_t^n))}{\partial R_{t-\Delta t}^{x,n}} \\ &= \frac{\partial C(S_t^n, X_t^\pi(S_t^n))}{\partial R_t^n}\frac{dR_t^n}{dR_{t-\Delta t}^{x,n}} \\ &= \frac{\partial C(S_t^n, X_t^\pi(S_t^n))}{\partial R_t^n} \cdot 1 \\ &= \hat{c}_t^n(S_t^n). \end{aligned}$$

We can update our estimates of the marginal value of energy in storage as:

$$\overline{v}_{t-\Delta t}^{n+}(S_{t-\Delta t}^{x,n}) = (1-\alpha^{n-1})\overline{v}_{t-\Delta t}^{n-1,+}(S_{t-\Delta t}^{x,n}) + \alpha^{n-1}\hat{v}_t^{n+}(S_t^n), \tag{20}$$

$$\overline{v}_{t-\Delta t}^{n-}(S_{t-\Delta t}^{x,n}) = (1-\alpha^{n-1})\overline{v}_{t-\Delta t}^{n-1,-}(S_{t-\Delta t}^{x,n}) + \alpha^{n-1}\hat{v}_t^{n-}(S_t^n), \tag{21}$$

```
An ADP algorithm
1. Initialize $\overline{v}_t^0$ $\forall t \in \mathcal{T}$.
2. Set: $n = 1$.
3. Draw a sample realization $\omega^n$.
4. For $t = 0, \ldots, T$:
    (a) Solve: $x_t^n = \arg\max_{x_t \in \mathcal{X}_t} \left( C(S_t^n, x_t) + \overline{V}_t^{n-1}(S_t^{x,n}) \right)$.
    (b) Calculate $\hat{c}_t^{n+}$ and $\hat{c}_t^{n-}$ as in (17) and (18).
    (c) If $t < T$, compute: $S_{t+\Delta t}^n = S^M(S_t^n, x_t^n, W_{t+\Delta t}^n)$.
5. For $t = T, \ldots, 0$:
    (a) Calculate $\hat{v}_t^{n+}$ and $\hat{v}_t^{n-}$ as in equation (19).
    (b) Update:
            $\overline{v}_{t-\Delta t}^n(S_{t-\Delta t}^x) \leftarrow \text{CAVE}(\overline{v}_{t-\Delta t}^{n-1}, \hat{v}_t^n)$.
6. If $n < N$, set $n = n + 1$ and go back to step 3. Else:
        Return the VFA $(\overline{v}^N)_{t=0}^T$.
```

Figure 2: An outline of the algorithm presented in section 5.

and then use the Concave Adaptive Value Estimation (CAVE) algorithm to update the VFA (see Godfrey & Powell (2001) for a detailed presentation). The CAVE algorithm performs a simple projection operation to enforce concavity of the piecewise linear approximation. Though it does not enjoy a convergence proof, the CAVE algorithm works remarkably well, as shown in section 6. It does, however, contain tunable parameters of *ad hoc* nature which play a critical scaling role and are crucial for fast numerical convergence. The algorithm is outlined in figure 2.

5.1 Stepsize Rules

We use a classical stochastic approximation method to calculate the smoothed estimates of the slopes at each iteration, as shown in (20) and (21). The smoothing requires a possibly stochastic stepsize, α_n, such that: $\alpha_n \geq 0$ *a.s.* $\forall n$, $\mathbb{E}[\sum_{n=1}^{\infty} \alpha_n^2] < \infty$, and $\sum_{n=1}^{\infty} \alpha_n = \infty$ *a.s.*

For many problems, even stepsizes which satisfy these conditions do not produce convergent algorithms if they decline too quickly or too slowly, or if they do not adapt to non-stationary data. For our particular application, we test a harmonic stepsize rule, which is deterministic, and the bias-adjusted Kalman filter stepsize rule (BAKF), which is time-dependent and was designed for stochastic simulations (George & Powell 2006).

The harmonic stepsize rule can be computed as:

$$\alpha^n = \frac{a}{a+n},$$

where a is a tunable parameter.

The BAKF stepsize rule is given by:

$$\alpha_t^n = 1 - \frac{(\overline{\sigma}_t^n)^2}{\overline{\nu}_t^n},$$

where $\overline{\sigma}_t^2$ is an estimate of the variance of the error, $\varepsilon_t^n = \overline{v}_t^{n-1} - \hat{v}_t^n$, and $\overline{\nu}_t$ is an estimate of the total variation.

We can construct these estimates as:

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

An estimate of the variance of the error can be calculated as:

$$(\overline{\sigma}_t^n)^2 = \frac{\overline{\nu}_t^n - (\overline{\beta}_t^n)^2}{1+\lambda_t^{n-1}}.$$

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 $\eta_{n,t}$ is a McClain stepsize:

$$\eta_t^n = \frac{\eta_t^{n-1}}{1+\eta_t^{n-1}-\overline{\eta}}.$$

The calculation of these estimates introduces a tunable parameter, $\overline{\eta}$, into the BAKF stepsize rule.

6 Algorithmic Performance Analysis

We assess the optimality of the ADP algorithm by comparing the performance of the approximate policy to optimal solutions for both deterministic and 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 (11) exactly. All the benchmark problems are available at http://www.castlelab.princeton.edu/datasets.htm. We also compare the performance of the algorithm against that of a model predictive control policy.

Ultimately, our goal is to solve complex stochastic storage control problems for which exact solutions cannot be computed. The goal of benchmarking is to provide a sense of optimality of the algorithm on problems which, despite being simpler, have overall similar characteristics to the more complex ones in terms of structure, governing physical dynamics, and flow network architecture. For the purpose of testing, the units of storage, wind energy and energy demand are arbitrary and equivalent; therefore, we do not explicitly show units in this section. The price is treated as \$ per unit of energy.

Below, we first benchmark the ADP 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).

6.1 Deterministic Experiments

For the deterministic benchmarks, we designed the test problems shown in Table 1, where the electricity prices, wind 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 $T = 2000$

periods, $t = 0, \Delta t, 2\Delta t, \dots, T$, where $\Delta t = 1$. We consider a generic storage device with a capacity $R^c = 100$, a round trip efficiency $\eta^c \eta^d = 0.81$, and maximum rates $\gamma^c = \gamma^d = 0.1$. We also assume $R_0 = 0$.

The problem instances are shown in Table 1. We obtain the optimal solution to each one of the test problems by solving one LP given by (8) subject to (1)-(6) for each t and to the transition dynamics expressed as constraints which link together all the time periods.

Deterministic problems are useful to test the ability of the algorithm to learn the solution in the presence of holding costs, when energy should be stored in the device as latest as possible in order to avoid incurring extra costs. These test problems also allow us to test the capability of the algorithm to learn to store energy in cases where the impact of storing is not felt until hundreds of time periods into the future.

In figure 3a we show a plot of the storage profile for test problem D1 obtained by ADP in the presence of constant supply and sinusoidal demand. The exact solution is not shown since it is almost identical to that obtained by the ADP approximation. 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 the wind at earlier points in time.

We also tested more complex dynamics for the wind energy supply and the energy spot prices. Figure 3b shows the storage level (scaled to fit) obtained by ADP along with the wind energy and demand profiles corresponding to test problem D9. Figure 3c shows the spot price process. The optimal storage profile is not shown since it was almost identical to the one generated by the approximate policy.

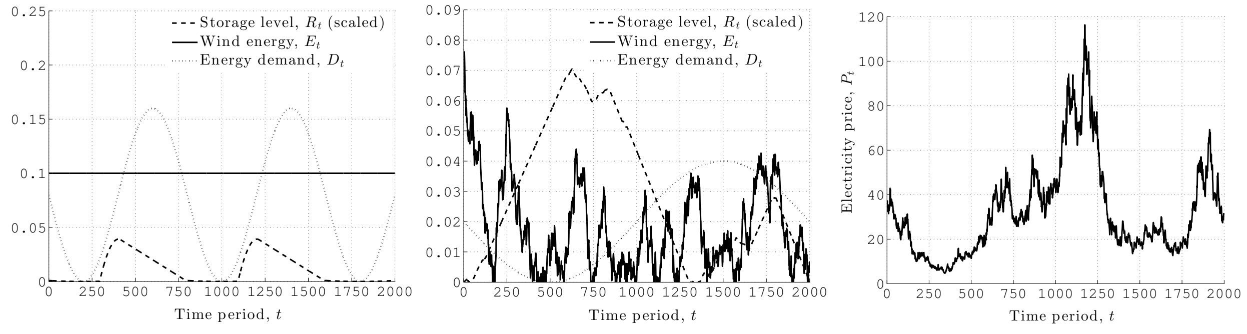
It is evident that the ADP algorithm was capable of learning the right control policy for these complex deterministic problems. To quantify the optimality of our ADP algorithm on this set of problems, we compared the objective value given by ADP after N iterations, F^N, to the true optimal value given by LP, F^*, by the performance metric:

$$\mathbb{F}^N = \frac{F^N}{F^*}.$$

The results are shown in Table 1. In all test problems, we found the ADP solution to be within 0.08% of optimal.

Label	**Price,** P_t	**Wind Energy,** E_t	**Demand,** D_t	F^*	$\mathbb{F}^{1000}$
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%

Table 1: Deterministic test problems.



(a) Test problem D1 (b) Test problem D9 (I) (c) Test problem D9 (II)

Figure 3: The wind energy, demand and energy prices from two of the deterministic test problems, and the corresponding storage profiles obtained by ADP. The storage profiles are scaled to fit.

6.2 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, (11) may be rewritten as:

$$V_t^*(S_t) = \max_{x_t \in \mathcal{X}_t} \left(C(S_t, x_t) + \sum_{s'=1}^{|\mathcal{S}_t|} \mathbb{P}_t(s' \mid S_t, x_t) V_{t+\Delta t}^*(s') \right) \forall t \in \mathcal{T} \tag{22}$$

where we replaced the conditional expectation in (11) by the time-dependent conditional transition probability $\mathbb{P}_t(s'|S_t, x_t)$ of going from state S_t to state s' given the decision x_t, and where we assume that $V_{T+\Delta t}^* = 0$. After solving (22) for all $S_t \in \mathcal{S}_t$, we can simulate the optimal policy π_t^* obtained by finding exact value functions $\{V_t^*\}_{t=0}^T$ from solving (22). Note that these are the value functions around the pre-decision state. For a given sample path $\omega \in \Omega$, we simulate the MDP using the policy:

$$X_t^{\pi^*}(S_t(\omega)) = \arg\max_{x_t \in \mathcal{X}_t} \left(C(S_t(\omega), x_t) + \sum_{s'=1}^{|\mathcal{S}_t(\omega)|} \mathbb{P}_t(s' \mid S_t(\omega), x_t) V_{t+\Delta t}^*(s' \mid S_t(\omega), x_t) \right) \forall t \in \mathcal{T}, \tag{23}$$

where $S_{t+1}(\omega) = S^M(S_t(\omega), X_t^{\pi^*}(S_t(\omega)), W_{t+1}(\omega))$.

We designed a set of test problems where prices and wind energy evolve stochastically over time and we obtained the optimal solution by solving (22) and (23). We considered the control of a generic, perfectly efficient storage device with capacity $R^c = 30$ 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 in order 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. In order to keep the problem computationally tractable, we discretized all variables fairly coarsely. After discretization, the

	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	1st-order + jump	$\mathcal{N}(0,0.5^2)$
S6	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	1st-order + jump	$\mathcal{N}(0,1.0^2)$
S7	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	1st-order + jump	$\mathcal{N}(0,2.5^2)$
S8	31	1.00	7	1.00	$\mathcal{U}(-1,1)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S9	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S10	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S11	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S12	31	1.00	7	1.00	$\mathcal{N}(0,2.0^2)$	41	1st-order + jump	$\mathcal{N}(0,5.0^2)$
S13	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	1st-order + jump	$\mathcal{N}(0,1.0^2)$
S14	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	1st-order + jump	$\mathcal{N}(0,1.0^2)$
S15	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	1st-order + jump	$\mathcal{N}(0,1.0^2)$
S16	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	1st-order	$\mathcal{N}(0,1.0^2)$
S17	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	1st-order	$\mathcal{N}(0,1.0^2)$
S18	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	1st-order	$\mathcal{N}(0,1.0^2)$
S19	31	1.00	7	1.00	$\mathcal{N}(0,0.5^2)$	41	1st-order	$\mathcal{N}(0,5.0^2)$
S20	31	1.00	7	1.00	$\mathcal{N}(0,1.0^2)$	41	1st-order	$\mathcal{N}(0,5.0^2)$
S21	31	1.00	7	1.00	$\mathcal{N}(0,1.5^2)$	41	1st-order	$\mathcal{N}(0,5.0^2)$

Table 2: Stochastic test problems. These datasets, along with sample paths, are available at http://www.castlelab.princeton.edu/datasets.htm.

state space consists of either 5551 or 8897 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 in at most 1 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 2. Note that $\Delta P = 1$ for problems S5-S21. The transition dynamics used in these simulations are given in sections 6.2.1-6.2.3.

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 meshsize 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:

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

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 paper.

6.2.1 The Wind Process

The wind process E_t is modeled using a bounded 1^{st}-order Markov chain:

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

such that $E^{min} \leq E_t \leq E^{max}$, and where $\hat{E}_t$ is either pseudonormally or uniformly distributed (see Table 2). The bounds $E^{min} = 1.00$ and $E^{max} = 7.00$ were fixed for all problem instances.

6.2.2 The Price Process

We test two different stochastic processes for P_t:

Sinusoidal:

$$P_{t+\Delta t} = \mu^P_{t+\Delta t} + \hat{P}_{0,t+\Delta t} \; \forall t \in \mathcal{T} \setminus \{T\},$$

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

1st-order Markov chain:

$$P_{t+\Delta t} = P_t + \hat{P}_{0,t+\Delta t} + \mathbb{1}_{\{u_{t+\Delta t} \leq p\}} \hat{P}_{1,t+\Delta t} \; \forall t \in \mathcal{T} \setminus \{T\},$$

such that $P^{min} \leq P_t \leq P^{max}$, and where $\hat{P}_{0,t}$ is either pseudonormally or uniformly distributed as indicated in Table 2. We let $u_t \sim \mathcal{U}(0,1)$, and we let $p = 0.031$ 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.

6.2.3 The Demand Process

The demand is assumed to be deterministic and given by $D_t = \lfloor \max\left[0, \mu_D - A_D \sin\left(\frac{2\pi t}{T}\right)\right] \rfloor$, where μ_D and A_D are fixed, and $\lfloor \cdot \rfloor$ is the floor function.

For each test problem, we simulated $K = 256$ different sample paths, $\{\omega^1, \ldots, \omega^K\} = \bar{\Omega} \subset \Omega$, and then calculated a statistical estimate of the value of the optimal policy:

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

6.3 Algorithm Tuning

We use our algorithm to solve each test problem and we compute an estimate of the mean value of the VFA policy after N iterations, $\overline{F}^N$. The algorithm requires tuning of the VFA aggregation level and the stepsize tunable parameter. Algorithm tuning is critical to the rate of convergence and the optimality of the algorithm.

6.3.1 State Aggregation

The level of state aggregation is given by the multi-index $\mathbf{g} = (g_e, g_p)$ where $\mathbf{g} = \mathbf{0}$ represents the fully aggregated state. Given g_e, we let $\{E^{min}, E^{min} + \Delta^{g_e} E, E^{min} + 2\Delta^{g_e} E, \ldots, E^{max} - \Delta^{g_e} E, E^{max}\}$ be the aggregated grid for the wind energy dimension of the post-decision state variable, where $\Delta^{g_e} E$ is the mesh size corresponding to level g_e. Similarly, we define the aggregated grid for the price dimension.

In problems S1-S4, the post-decision state variable does not include the electricity price since $P_{t+\Delta t}$ is independent of P_t. For these problems, we tested three aggregation levels in the wind dimension:

$$\begin{aligned}
\Delta^0 E &= E^{max} - E^{min}, \\
\Delta^1 E &= \frac{E^{max} - E^{min}}{7}, \\
\Delta^2 E &= \frac{E^{max} - E^{min}}{13}.
\end{aligned}$$

In problems S5-S21, we tested two aggregation levels in the wind dimension,

$$\begin{aligned}
\Delta^0 E &= E^{max} - E^{min}, \\
\Delta^1 E &= \frac{E^{max} - E^{min}}{7},
\end{aligned}$$

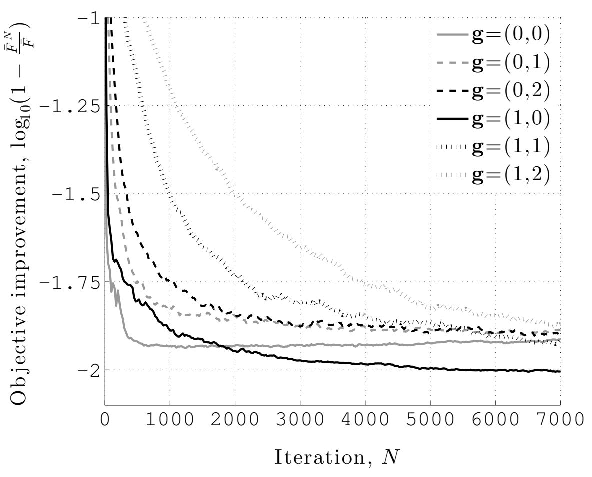
and three in the price dimension,

$$\begin{aligned}
\Delta^0 P &= P^{max} - P^{min}, \\
\Delta^1 P &= \frac{P^{max} - P^{min}}{21}, \\
\Delta^2 P &= \frac{P^{max} - P^{min}}{41}.
\end{aligned}$$

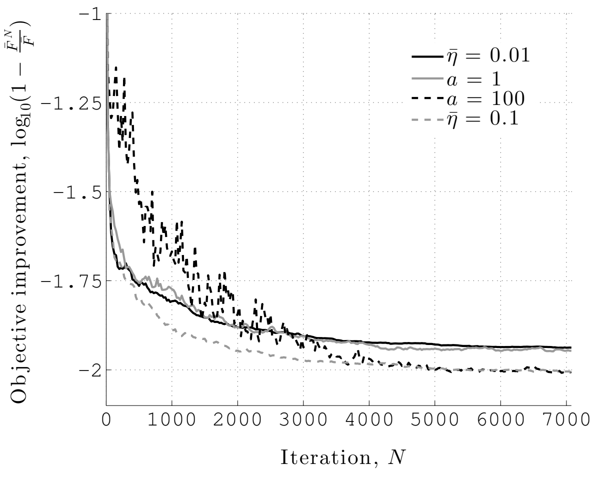
In figure 4a, we show the convergence plot for test problem S19 for each aggregation level. It is clear that the aggregation level $\mathbf{g} = (1, 0)$ resulted in better performance of the algorithm. We found this aggregation level to work best for all test problems.

6.3.2 Stepsize Tuning

We tested the harmonic and BAKF stepsize rules over a range of values for the tunable parameters. The convergence plot for test problem S19 is shown for different values of $\overline{\eta}$ and a on figure 4b. Though most values resulted in similar performance for the algorithm, we found BAKF to be more stable. We found that the harmonic stepsize tended to work well but the best value of a varied from one problem instance to another. We found that BAKF with value of $\overline{\eta} = 0.1$ to consistently outperform other stepsize configurations.

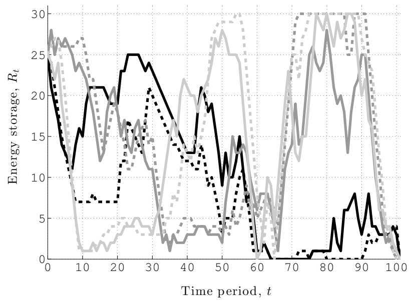


(a) Convergence plot for BAKF with $\overline{\eta} = 0.1$ using different aggregation levels.

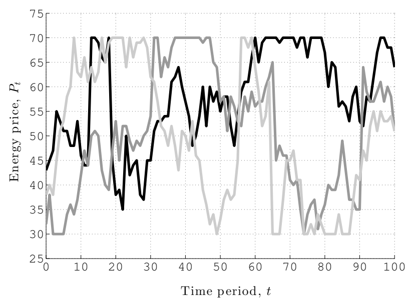


(b) Convergence plot for aggregation level $\mathbf{g} = (1, 0)$ using different stepsize configurations.

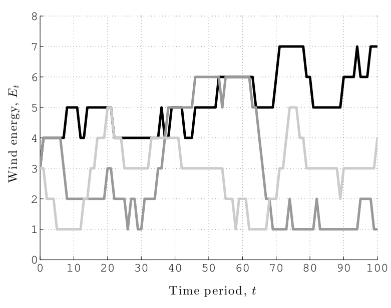
Figure 4: The effect of aggregation and stepsize configuration on the algorithm for test problem S19.



(a) Three energy storage profiles.



(b) Three spot price profiles.



(c) Three wind energy profiles.

Figure 5: Results and sample paths from test problem S9. The storage profiles in figure 5a were generated with the energy prices and wind energy profiles of the same color in figures 5b and 5c.

6.4 Numerical Results

In figure 5a we show the storage profiles for three sample paths from test problem S9. The optimal solution is shown as a dashed line and the approximate solution as a solid one of the same color. In 5b and 5c we show the price and wind energy profiles, respectively, corresponding to the storage profiles of the same color in figure 5a. It is important to note that while the approximate storage policy follows the same overall pattern as the optimal one, it is not exactly the same. However, in stochastic problems we are more concerned with designing policies that are near-optimal in value even if the VFAs, and hence the approximate policy itself, are different from the optimal value functions.

We define the metric $\mathbb{F}^N$ to evaluate the performance of the algorithm:

$$\mathbb{F}^N = \frac{\overline{F}^N}{\overline{F}}.$$

Additionally, we calculate an estimate of the mean value of a model predictive control policy, F^{MPC}, which is

Label	$\overline{F}$	$\mathbb{F}^0$	$\mathbb{F}^{50}$	$\mathbb{F}^{150}$	$\mathbb{F}^{500}$	$\mathbb{F}^{1250}$	$\mathbb{F}^{7000}$	s^{7000}	$\mathbb{F}^{MPC}$	s^{MPC}
S1	19,480.11	44.69%	96.16%	98.68%	99.25%	99.37%	99.45%	±0.25%	97.21%	±0.19%
S2	18,915.05	44.14%	96.21%	98.68%	99.04%	99.11%	99.29%	±0.22%	97.25%	±0.21%
S3	19,730.28	44.40%	95.87%	98.88%	99.21%	99.32%	99.45%	±0.25%	97.12%	±0.20%
S4	19,831.66	44.14%	95.36%	98.04%	98.51%	98.65%	98.80%	±0.65%	96.83%	±0.19%
S5	16,806.49	57.36%	96.68%	98.38%	98.51%	98.89%	99.18%	±0.32%	98.39%	±0.31%
S6	17,089.29	57.30%	97.51%	98.12%	98.68%	98.97%	99.12%	±0.33%	98.09%	±0.30%
S7	18,104.07	56.92%	97.56%	98.26%	98.62%	99.04%	99.21%	±0.34%	97.14%	±0.28%
S8	19,011.24	56.00%	97.53%	98.42%	98.70%	98.96%	99.19%	±0.34%	95.98%	±0.31%
S9	17,963.79	57.09%	96.54%	98.21%	98.48%	98.74%	98.99%	±0.35%	95.17%	±0.46%
S10	19,079.16	55.79%	96.68%	97.84%	98.06%	98.48%	98.78%	±0.34%	96.24%	±0.28%
S11	19,396.01	55.02%	96.89%	97.81%	98.22%	98.48%	98.72%	±0.33%	96.03%	±0.32%
S12	19,500.51	54.51%	96.93%	97.97%	98.17%	98.55%	98.81%	±0.33%	96.38%	±0.26%
S13	16,547.09	58.88%	96.85%	98.02%	98.31%	98.56%	98.80%	±0.35%	97.59%	±0.40%
S14	17,700.78	57.29%	96.97%	98.02%	98.18%	98.47%	98.66%	±0.35%	97.96%	±0.31%
S15	17,972.71	56.43%	97.07%	98.01%	98.20%	98.55%	98.70%	±0.34%	98.02%	±0.28%
S16	14,113.12	59.45%	97.45%	99.02%	99.12%	99.20%	99.24%	±0.30%	99.58%	±0.07%
S17	15,115.53	57.54%	98.74%	99.33%	99.61%	99.67%	99.80%	±0.07%	99.61%	±0.07%
S18	15,377.29	56.42%	98.82%	99.30%	99.47%	99.62%	99.75%	±0.08%	99.60%	±0.06%
S19	17,467.27	58.09%	97.20%	97.94%	98.47%	98.75%	98.90%	±0.34%	96.33%	±0.30%
S20	18,601.82	56.72%	97.39%	98.90%	99.03%	99.27%	99.55%	±0.12%	96.98%	±0.21%
S21	18,912.27	55.86%	97.34%	98.88%	99.02%	99.28%	99.56%	±0.11%	96.73%	±0.20%

Table 3: Stochastic benchmarking results.

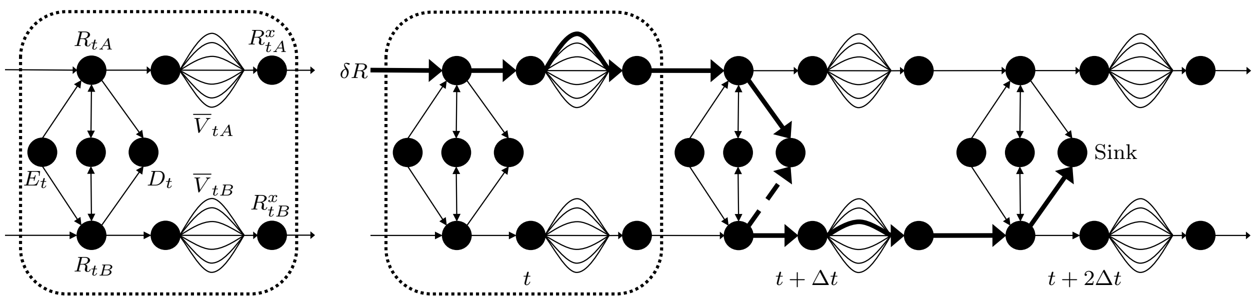
found by solving (9) and (10) for all $\omega \in \bar{\Omega}$. We tested different lookahead horizons $H = 1, 10, 50$ and 100. We compare the performance of our algorithm against the model predictive control policy with $H = 100$, since we found this value of H to yield the best solution. For this purpose, we define the metric:

$$\mathbb{F}^{MPC} = \frac{\overline{F}^{MPC}}{\overline{F}}.$$

We calculate the standard errors for $\mathbb{F}^N$ and $\mathbb{F}^{MPC}$, s^N and s^{MPC}, respectively. The results are shown in Table 3. For all test problems, the ADP algorithm was capable of learning a policy that is within 1.34% of optimal in the worst case (test problem S14) and within 0.20% in the best case (test problem S17). The model predictive control policy performed well on problems S16-S18, where price evolves according to a 1st-order model with very low noise and no jumps. However, in all but one test problem (S16), the ADP policy resulted in a better policy. In particular, ADP performed remarkably well on problems S8-S12, where prices are very noisy and may jump, relative to the much weaker performance of model predictive control. On highly noisy, but time-dependent problems (S1-S4), model predictive control did not perform as well as expected.

7 Application: The Value of a Multidimensional Storage System

In this section, we extend the presentation of the algorithm to handle an M-dimensional portfolio of devices. We let $R_t = (R_{t1}, \ldots, R_{tM})$ be the $M \times 1$ resource vector, x_t be the $(2+4M) \times 1$ decision vector and Φ be a $M \times (2+4M)$ incidence matrix which captures the flow of energy into and out of each device. If we let η^c_m be the charging efficiency of device m and we let $\phi_m = (0, 0, -1, \eta^c_m, \eta^c_m, -1)$ be the column vector that accounts



(a) One-period network. (b) Tracking the marginal value of energy through the multi-period network.

Figure 6: An illustration of the marginal value calculation for the two-dimensional problem. Bold arrows indicate increased flow due to the perturbation $+\delta R$, while the dashed arrow indicates a decrease in flow.

for flow into and out of device m (as in section 2.6), then:

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

We introduce a separable VFA, $\overline{V}_t(S_t^x)$:

$$\begin{aligned} V_t^x(S_t^x) &\approx \overline{V}_t(S_t^x) \\ &= \sum_{m=1}^{M} \overline{V}_{tm}(R_{tm}^x, E_t, P_t), \end{aligned}$$

where $\overline{V}_{tm}(R^x_{tm}, E_t, P_t)$ is a piecewise linear function in the post-decision resource variable for each device m. The optimization problem in (13) may be rewritten as:

$$X_t^\pi(S_t) = \underset{\substack{x_t \in \mathcal{X}_t^n \\ r_t \in \mathcal{R}_t^n}}{\arg\max} \big(C(S_t, x_t) + \sum_{m=1}^{M} \sum_{i=1}^{K_{tm}} \overline{v}_{tim} r_{tim} \big) \ \forall t \in \mathcal{T},$$

where $r_t = (r_{t1}, \dots, r_{tM})$, such that:

$$\begin{pmatrix} \sum_{i=1}^{K_{t1}} r_{ti1} \\ \vdots \\ \sum_{i=1}^{K_{tM}} r_{tiM} \end{pmatrix} - \Phi x_t = R_t,$$
$$0 \le r_{tim} \le \bar{r}_{tim} \ \forall m \ \forall i.$$

We also need to generalize the calculation of the marginal value of energy in storage in (19) for the M-dimensional case. We need to perturb each device by an amount δR one at a time, and now this perturbation may result in energy held in storage from any device at a later time $t' > t$. In figure 6, we illustrate the extension of the algorithm to the two-dimensional case and it is easy to see how the algorithm may be extended to portfolios of dimension higher than two.

We let $S_t^{n,m+} = (R_t^n + e_m \delta R, E_t^n, D_t^n, P_t^n)$, where e_m is the standard M-dimensional unit column vector with a 1 in the mth component. We 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 may compute the marginal value of energy in storage as follows:

$$\hat{v}_t^{n+}(S_t^n) = \begin{cases} \hat{c}_t^{n+}(S_t^n) + \Delta_t^{n+} \hat{v}_{t+\Delta t}^{n+}(S_{t+\Delta t}^n) & \text{for } 0 \leq t < T, \\ \hat{c}_T^{n+}(S_T^n) & \text{otherwise,} \end{cases}$$

where $\hat{v}_t^{n+}(S_t^n)$ is now an $M \times 1$ vector containing observations of the marginal value of energy in each device, and Δ_t^{n+} is an $M \times M$ matrix with the same function as its scalar analog in (19). This matrix may be easily computed as follows:

$$\Delta_t^{n+} = \frac{1}{\delta R} \left(\Phi X^{n+} \right)^T,$$

where $X^{n+} = \begin{pmatrix} x_t^{n,1+} - x_t & \cdots & x_t^{n,M+} - x_t \end{pmatrix}$. The (i,j)th entry of Δ_t^{n+} gives the fraction of energy held in storage j when we perturb the amount of energy in device i by $+\delta R$. Analogously, we compute Δ_t^{n-} and $\hat{v}_t^{n-}(S_t^n)$.

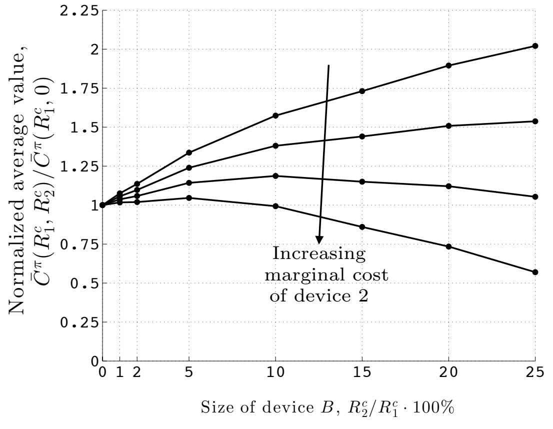
For our analysis, we first consider a grid-level energy storage system consisting of two storage devices, a less expensive primary device with higher energy capacity but lower efficiency such as a lead-acid or lithium-ion battery, and a more expensive secondary device with lower energy capacity but with 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.

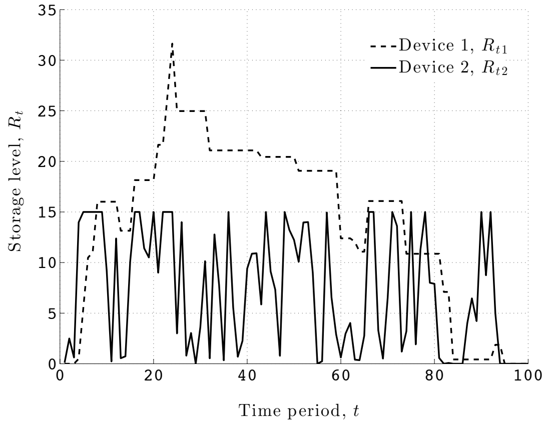
Let device 1 and device 2 be two distinct energy storage devices. Any variable indexed by 1 or 2 represents the component of that variable which corresponds to device 1 and 2, respectively. We let $\eta_1^c \eta_1^d < \eta_2^c \eta_2^d$, $\gamma_1^c < \gamma_2^c$ and $R_2^c \ll R_1^c$, and we define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

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

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



(a) The effect of size of device 2 on the value of the system.



(b) Two-device storage policy simulation.

Figure 7

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:

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

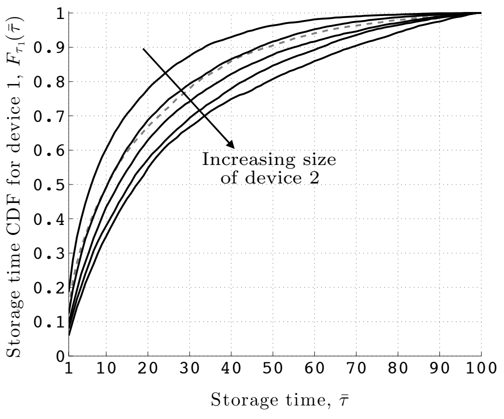
where π is the policy given by $\{\overline{V}_t^N(\cdot)\}_{t \in \mathcal{T}}$, and c_1 and c_2 are the marginal capital costs of storage of type 1 and 2, respectively, per time period. An estimate of C^{π} obtained with K sample paths is denoted $\overline{C}^{\pi}$.

In figure 7a, 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 7a 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 non-zero amount of energy remains in device i (equivalently, the time between time between periods where the device is empty) and we look at its distribution.

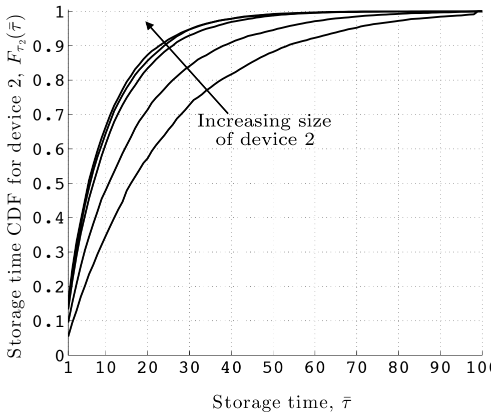
Figure 7b shows the storage profiles for devices 1 and 2 generated by simulating π on a single sample path. It is visually evident that device 1 and device 2 operate on two distinct timescales: the frequency of storage in device 2 is much higher than in device 1. In fact, device 2 is completely emptied 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 $F_{\tau_i}(\bar{\tau}) = \text{Prob}\{\tau_i \leq \bar{\tau}\}$, for $i = 1, 2$.

In figures 8a and 8b, 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.

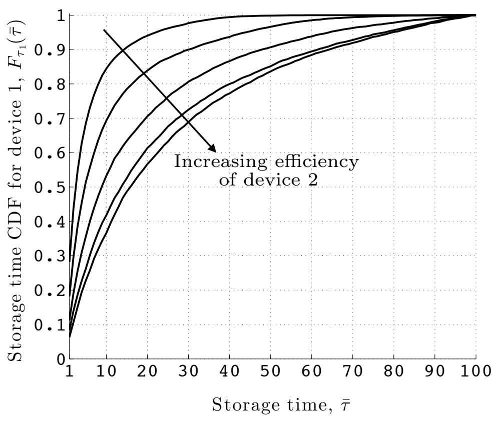
We observed similar behavior of the energy storage system as we varied the efficiency of device 2, $\eta_2^c\eta_2^d$, from 75% to 95% while keeping that of device 1 constant, $\eta_1^c\eta_1^d = 75\%$. Figures 8c and 8d show $F_{\tau_1}(\cdot)$ and $F_{\tau_2}(\cdot)$, respectively, as a function of $\eta_2^c\eta_2^d$. It is easily seen that increasing $\eta_2^c\eta_2^d$ has a substantial effect on the storage timescale of device 1 but the storage timescale of device 2 only varies slightly. It is important to note that the storage timescale is not only affected by the energy capacity and efficiency of the device, but also by



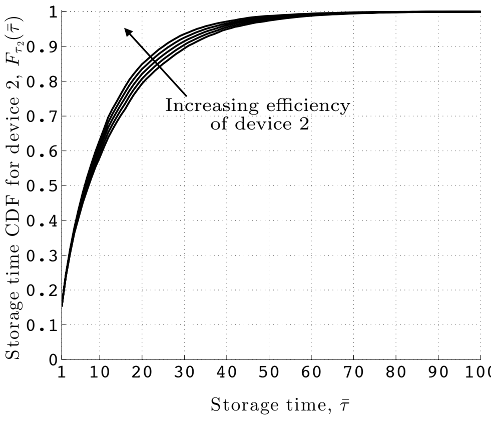
(a) The distribution of the storage timescale for device 1 as a function of the size of device 2.



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



(c) The distribution of the storage timescale for device 1 as a function of the efficiency of device 2.



(d) The distribution of the storage timescale for device 2 as a function of its efficiency.

Figure 8: In figure 8a, the dashed line represents the $F_{\tau_1}(\cdot)$ for the one-device system. In 8a and 8b, the arrow indicates increasing the size of device 2, R_2^c, from 2.5% to 20% of the size of device 1. In 8c and 8d, it indicates increasing the efficiency of device 2, $\eta_2^c\eta_2^d$, from 75% to 95%.

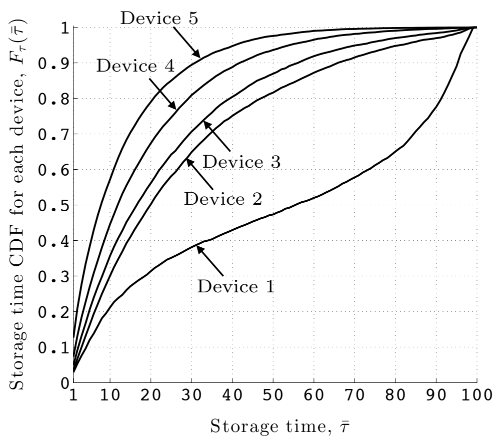


Figure 9: The distribution of the storage timescale for each device in the five-device energy system.

its maximum power rate. Throughout the numerical experiments in this section, we assumed that device 2 always had a higher power capacity than device 1.

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 9 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 timescales 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 linear program, 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.

8 Conclusion

In this paper, we presented in detail an algorithm based on the approximation of the optimal value functions 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 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.

References

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

Barton, J. & Infield, D. (2004), 'Energy Storage and Its Use With Intermittent Renewable Energy', *IEEE Transactions on Energy Conversion* **19**(2), 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 for Sustainable Development* **14**(4), 302–314.

Bertsekas, D. P. (2012), *Dynamic Programming and Optimal Control, Vol. II*, 4th editio edn, Athena Scientific.

Bertsekas, D. & Tsitsiklis, J. (1996), *Neuro-Dynamic Programming*, Athena Scientific; 1 edition.

Bird, L. & Lew, D. (2012), Integrating Wind and Solar Energy in the U.S. Bulk Power System: Lessons from Regional Integration Studies., Technical report, National Renewable Energy Laboratory.

Black, M. & Strbac, G. (2006), 'Value of storage in providing balancing services for electricity generation systems with high wind penetration', *Journal of Power Sources* **162**(2), 949–953.

Camacho, E. F. & Bordons, C. (2004), *Model Predictive Control*, 2nd edn, Springer-Verlag, New York.

Carmona, R. & Ludkovski, M. (2010), 'Valuation of energy storage: an optimal switching approach', *Quantitative Finance* **10**(4), 359–374.

Castronuovo, E. & Lopes, J. (2004), 'On the Optimization of the Daily Operation of a Wind-Hydro Power Plant', *IEEE Transactions on Power Systems* **19**(3), 1599–1606.

Costa, L., Bourry, F., Juban, J. & Kariniotakis, G. (2008), Management of Energy Storage Coordinated with Wind Power under Electricity Market Conditions, *in* 'Proceedings of the 10th International Conference on Probabilistic Methods Applied to Power Systems', pp. 1–8.

Dell, R. & Rand, D. (2001), *Understanding Batteries*, Royal Society of Chemistry; 1 edition.

Dicorato, M., Forte, G., Pisani, M. & Trovato, M. (2012), 'Planning and Operating Combined Wind-Storage System in Electricity Market', *IEEE Transactions on Sustainable Energy* **3**(2), 209–217.

Enders, J. & Powell, W. (2010), 'A dynamic model for the failure replacement of aging high-voltage transformers', *Energy Systems Journal* **1**(1), 31–59.

Enders, J., Powell, W. B. & Egan, D. M. (2010), 'Robust policies for the transformer acquisition and allocation problem', *Energy Systems Journal* **1**(3), 245–272.

Eyer, J. M., Iannucci, J. J. & Corey, G. P. (2004), Energy Storage Benefits and Market Analysis Handbook, A Study for the DOE Energy Storage Systems Program, Technical report, Sandia National Laboratories, SAND2004-6177.

Fleten, S.-E. & Kristoffersen, T. K. (2007), 'Stochastic programming for optimizing bidding strategies of a Nordic hydropower producer', *European Journal of Operational Research* **181**(2), 916–928.

George, A. P. & Powell, W. B. (2006), 'Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming', *Machine Learning* **65**(1), 167–198.

Godfrey, G. & Powell, W. B. (2001), 'An Adaptive, Distribution-Free Algorithm for the Newsvendor Problem with Censored Demands, with Application to Inventory and Distribution Problems', *Management Science* **47**(8), 1101–1112.

Hastie, T., Tibshirani, R. & Friedman, J. H. (2003), *The Elements of Statistical Learning*, Springer.

Kim, J. H. & Powell, W. B. (2011), 'Optimal Energy Commitments with Storage and Intermittent Supply', *Operations Research* **59**(6), 1347–1360.

Kraining, M., Wang, Y., Akuiyibo, E. & Boyd, S. (2011), Operation and Conguration of a Storage Portfolio via Convex Optimization, *in* 'Proceedings IFAC World Congress', pp. 10487–10492.

Kuperman, A. & Aharon, I. (2011), 'Batteryultracapacitor hybrids for pulsed current loads: A review', *Renewable and Sustainable Energy Reviews* **15**(2), 981–992.

Lai, G., Margot, F. & Secomandi, N. (2010), 'An Approximate Dynamic Programming Approach to Benchmark Practice-Based Heuristics for Natural Gas Storage Valuation', *Operations Research* **58**(3), 564–582.

Löhndorf, N. & Minner, S. (2010), 'Optimal day-ahead trading and storage of renewable energiesan approximate dynamic programming approach', *Energy Systems* **1**(1), 61–77.

Mokrian, P. & Stephen, M. (2006), A stochastic programming framework for the valuation of electricity storage, *in* '26th USAEE/IAEE North American Conference', pp. 24–27.

Nascimento, J. & Powell, W. (2013), 'An Optimal Approximate Dynamic Programming Algorithm for the Economic Dispatch Problem with Grid-Level Storage', *IEEE Transactions on Automatic Control (to appear in print)* .

Powell, W. B. (2011), *Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd Edition*, Wiley.

Powell, W. B., George, A., Simao, H., Scott, W., Lamont, A. & Stewart, J. (2011), 'SMART: A Stochastic Multiscale Model for the Analysis of Energy Resources, Technology, and Policy', *INFORMS Journal on Computing* **24**(4), 665–682.

Puterman, M. L. (1994), *Markov Decision Processes*, John Wiley and Sons, New York.

Ru, Y., Kleissl, J. & Martinez, S. (2013), 'Storage Size Determination for Grid-Connected Photovoltaic Systems', *IEEE Transactions on Sustainable Energy* **4**(1), 68–81.

Simao, H. P., Day, J., George, A. P., Gifford, T., Nienow, J. & Powell, W. B. (2008), 'An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application', *Transportation Science* **43**(2), 178–197.

Simao, H. & Powell, W. B. (2009), 'Approximate dynamic programming for management of high-value spare parts', *Journal of Manufacturing Technology Management* **20**(2), 147–160.

Sioshansi, R. & Denholm, P. (2013), Benefits of Co-Locating Concentrating Solar Power and Wind.

Sioshansi, R., Denholm, P., Jenkin, T. & Weiss, J. (2009), 'Estimating the value of electricity storage in PJM: Arbitrage and some welfare effects', *Energy Economics* **31**(2), 269–277.

Sutton, R. & Barto, A. (1998), *Reinforcement Learning: An Introduction*, A Bradford Book, MIT Press, Cambridge, MA.

Teleke, S., Baran, M. E., Bhattacharya, S. & Huang, A. Q. (2010), 'Rule-Based Control of Battery Energy Storage for Dispatching Intermittent Renewable Sources', *IEEE Transactions on Sustainable Energy* **1**(3), 117–124.

Topaloglu, H. & Powell, W. B. (2005), 'A Distributed Decision-Making Structure for Dynamic Resource Allocation Using Nonlinear Functional Approximations', *Operations Research* **53**(2), 281–297.

Van Roy, B., Bertsekas, D., Lee, Y. & Tsitsiklis, J. (1997), A Neuro-Dynamic Programming Approach to Retailer Inventory Management, *in* 'Proceedings of the 36th IEEE Conference on Decision and Control', Vol. 4, IEEE, pp. 4052–4057.

Vanderbei, R. (2007), *Linear Programming: Foundations and Extensions*, 3rd edn, Springer, New York.

Vazquez, S., Lukic, S. M., Galvan, E., Franquelo, L. G. & Carrasco, J. M. (2010), 'Energy Storage Systems for Transport and Grid Applications', *IEEE Transactions on Industrial Electronics* **57**(12), 3881–3895.

Xi, X., Sioshansi, R. & Marano, V. (2013), A Stochastic Dynamic Programming Model for Co-optimization of Distributed Energy Storage.

Zhou, Y., Scheller-Wolf, A. & Secomandi, N. (2013), Managing Wind-based Electricity Generation in the Presence of Storage and Transmission Capacity.

Zhou, Y., Scheller-Wolf, A., Secomandi, N. & Smith, S. (2012), Is It More Valuable to Store or Destroy Electricity Surpluses?