An Intelligent Battery Controller Using Bias-Corrected Q-learning

Donghun Lee
Department of Computer Science
Princeton University
35 Olden Street
Princeton, New Jersey 08544

Warren B. Powell
Department of Operations Research
and Financial Engineering
Princeton University
Princeton, New Jersey 08544

Abstract

The transition to renewables requires storage to help smooth short-term variations in energy from wind and solar sources, as well as to respond to spikes in electricity spot prices, which can easily exceed 20 times their average. Efficient operation of an energy storage device is a fundamental problem, yet classical algorithms such as Q-learning can diverge for millions of iterations, limiting practical applications. We have traced this behavior to the max-operator bias, which is exacerbated by high volatility in the reward function, and high discount factors due to the small time steps. We propose an elegant bias correction procedure and demonstrate its effectiveness.

Introduction

As part of its plan to reduce CO2 production, the Department of Energy has decided that energy from renewable sources such as the wind and the sun should represent a growing source of electricity generation. In the U.S., most states have put in place renewable portfolio standards to enforce the transition toward these sources. At the same time, the transition to deregulated power prices have exposed consumers to the high volatility of electricity spot prices, which will only be exacerbated as the variability from renewables grows.

It is now widely recognized that energy storage can play a major role in smoothing the volatility in generation and prices (Barton and Infield 2004; Huggins 2010). Given the high expense of different types of storage, an efficient controller would be an important part of this solution. The problem of optimizing the storage and withdrawal of energy from a battery storage device can be easily formulated as a Markov decision process, yet classical algorithms such as Q-learning, which appear to be well suited to this problem class, exhibit astonishingly slow convergence, and actually demonstrate divergent behavior for millions of iterations. This behavior appears to be due to the bias induced by the max-operator in the Q-learning updating procedure, given by

$$\hat{Q}_t = p_t a_t + \gamma \max_{a'} \bar{Q}_{t-1}(s_{next}, a'), \quad (1)$$

$$\bar{Q}_t(s_t, a_t) = (1-\alpha_{t-1})\bar{Q}_{t-1}(s_t, a_t) + \alpha_{t-1}\hat{Q}_t. \quad (2)$$

where a_t captures the storage $(a_t < 0)$ or withdrawal $(a_t > 0)$ of electricity at a price p_t that evolves over time.

Our problem exhibits two empirical features: high volatility in the rewards $R(S_t, a_t) = p_t a_t$ due to the nature of electricity spot prices, and a discount factor γ that is very close to one because of small time steps (which might on the order of a few minutes due to the nature of the dynamic control and the volatility of the driving processes). Primarily due to the stochastic prices (there can be other forms of uncertainty), the observations $\hat{Q}_t$ exhibit very high variance, which translates into relatively high variance in the estimates $\bar{Q}_t$.

It is well known that the expected value of a maximum of a set of random variables is biased, a property that would exist in any Q-learning application. However, it appears that the characteristics of this particular application, which is a fundamental control problem in energy systems, exacerbate the issue to a degree that has not been observed before. This paper offers a solution that is both practical and elegant.

We formulate our problem as a Markov decision process with state $S_t = (r_t, p_t)$ where r_t is the energy level in the battery (the resource level), and p_t is the price. For our model, we use discretized storage and price processes, so that the state space $\mathcal{S}$ is discrete. In practice, we would not have a true model of the stochastic price process, which is highly volatile; prices may spike from averages of \$50 per megawatt-hour to over \$1000 per MWhr, for periods spanning several minutes to an hour. The discretized action space $\mathcal{A}$ is relatively small, making this a good candidate for Q-learning to find a near-optimal policy.

Our work reports on development of an intelligent battery control problem reflecting the stochastic nature of real time electricity prices and the physical properties of batteries, where the transient divergence in Q-learning estimates can be quantified. We propose a bias-correction term based on an analytical model of a simplified decision process. We then show empirically that the bias correction dramatically improves the rate of convergence for our application.

The model

We formalize the intelligent battery control problem as a MDP as follows. Given the state $S_t \in \mathcal{S}$ with $S_t = (r_t, p_t)$, we assume we have a stochastic prices process $p_{t+1} = p_t + \hat{p}_{t+1}$. Let $\omega = (\hat{p}_1, \hat{p}_2, \ldots)$ be a sample realization of the

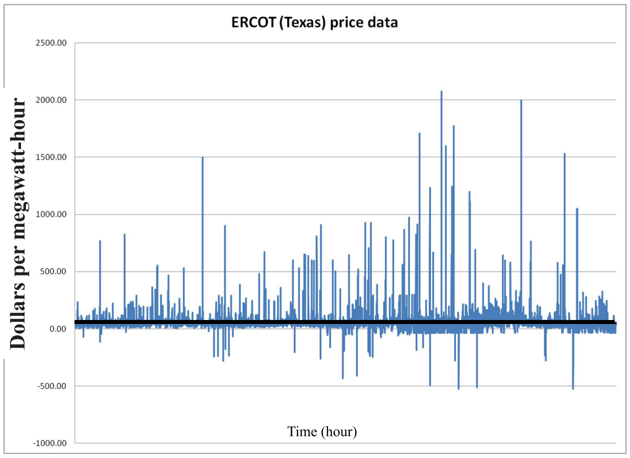

Figure 1: Electricity spot prices for ERCOT Texas.

price process with $\omega \in \Omega$, and where $\mathfrak{F}$ is a sigma-algebra on Ω with filtrations $\mathfrak{F}_t$. We let $(\Omega, \mathfrak{F}, \mathcal{P})$ be our probability space, where $\mathcal{P}$ is a probability measure on $(\Omega, \mathfrak{F})$. The problem is to find a policy $\pi(S_t)$ that returns an action $a_t \in \mathcal{A}$ that solves

$$\max_{\pi \in \Pi} \mathbb{E} \sum_{t=0}^{\infty} \gamma^t R(S_t, \pi(S_t)),$$

where γ is a discount factor and Π is a set of admissible policies adapted to the filtration $\mathfrak{F}_t$. It is well known that an optimal policy is characterized by Bellman's equation

$$V(S_t) = \max_{a \in \mathcal{A}} \left(R(S_t, a) + \gamma \mathbb{E} \left[V(S_{t+1}) | S_t \right] \right), \tag{3}$$

where $S_{t+1} = \mathcal{T}(S_t, a_t, \hat{p}_{t+1})$ is derived from the transition function $\mathcal{T}$. We do not have access to the model $\mathcal{T}$, and nor can we compute the expectation. For this reason, Q-learning, given by equations (1) - (2), is an attractive algorithmic strategy, and is supported by powerful convergence theory (Tsitsiklis 1994; Jaakkola, Jordan, and Singh 1994).

Perhaps the biggest challenge in our research is the price process, which is well known to be heavy-tailed. An average wholesale price is approximately \$50 per MWhr, with spikes that regularly exceed \$1,000 per MWhr, as shown in figure 1. This has been recognized and studied by a number of authors (see, for example, (Eydeland and Wolyniec 2003; R. Huisman 2003). (Kim and Powell 2011) makes an empirical case that the distribution has infinite variance, and is well-modeled by a scaled Cauchy distribution. This heavy-tailed property adds to the overall level of variability in the observations of rewards, and hence the variance of $\hat{Q}$.

We use a mean-reverting jump diffusion model proposed in (Cartea and Figueroa 2005) for our experimental work, which models the log of the price process $Y_t = \log(P_c + c)$ where c is an appropriately chosen shift constant to avoid zero or negative prices. The log-price model is given by

$$dY_t = \alpha(\mu - Y_t)dt + \sigma dW_t + dN_t,$$

where μ is the long term equilibrium price, α is the mean reversion rate, W_t is a Brownian motion and N_t is a Poisson jump process with parameter (mean rate) λ. α, μ, σ and λ are estimated from data on a rolling basis.

Transient bias in Q-learning

As shown in equations (1)-(2), $\bar{Q}_t(s_t, a_t)$ is a statistical estimate, which means that $\max_a \bar{Q}_t(s_t, a_t)$ will be biased upward. As a result, the observations $\hat{Q}_t$ are biased upward, introducing a bias into the estimates $\bar{Q}_t$. When the discount factor γ is close to 1 (as it is in our application), the effect is significant.

The effect of this particular bias in applications has been observed earlier in estimating asset values in finance (Brown 1974), and by authors in reinforcement learning (Thurn and Schwartz 1993; Ormoneit and Sen 2002), management science (Smith and Winkler 2006) and in operations research (Powell 2007). (Kaelbling, Littman, and Moore 1996) discusses the dangers of combining the max operator with function approximations. As of this writing, the bias does not seem to have caused serious practical problems, and relatively few have proposed corrections (exceptions include (Powell 2007) and (van Hasselt 2010)).

Reinforcement learning algorithms that recognize the max-operator bias either explicitly handle the bias as special cases (for example, (da Motta Salles Barreto and Anderson 2008)), or avoid it altogether by decoupling the max-biased sampling step and the stochastic approximation step as in (Reynolds 2001; van Hasselt 2010). For example, double Q-learning keeps track of two sets of estimates of $\bar{Q}_{t-1}$ to use one randomly selected set of estimate to make decisions to generate sample realizations of $\hat{Q}_t$ and use the other estimate to perform stochastic approximation to update $\bar{Q}_t$ (van Hasselt 2010).

In this work, we 1) illustrate the minimal condition to cause the transient max-induced bias in Q-learning, 2) propose bias-corrected Q-learning algorithm that corrects the illustrated bias problem, and 3) report the improved performance of our algorithm in the intelligent battery control problem where the transient bias degrades the performance of Q-learning.

Some initial results

First, we define a convenient term to analyze transient bias:

Definition 1 (Actual max-induced bias). *The actual* max-*bias induced by Q-learning at iteration t for state-action pair (s, a) is defined as:*

$$b_t(s, a) := \bar{Q}_t(s, a) - \bar{Q}_t^*(s, a),$$

where $\bar{Q}_t$ is the Q value after iteration t, and $\bar{Q}_t^*$ is the "oracle" Q estimate calculated with the exactly same algorithm (taking the same action as that used to calculate $\bar{Q}_t$) but with full knowledge of the underlying MDP, including the true reward function.

We first develop bounds on the bias for a simplified system with a single state.

Proposition 2 (Bounds on the bias for a single-state system). *For an MDP with a single state, multiple actions,*

$\hat{R}(a) := \mathbb{E}R(a) + \hat{\varepsilon}, \mathbb{E}\hat{\varepsilon} = 0, \mathrm{Var}\hat{\varepsilon} \neq 0,$

$$b_t \le \left(\hat{R}\left(a_t\right) - R\left(a_t^*\right)\right).$$

We also have

$$\mathbb{E}b_t \ge 0.$$

Proof. For Q-learning with an ϵ-greedy learning policy in iteration $t < \infty$, s_t, the estimates $\bar{Q}_t$ are given as deterministic values. With probability $\epsilon > 0$, a greedy action a_t will be made such that the sample $\hat{Q}_t$ for update in iteration t will be:

$$\hat{Q}_t \leftarrow \max_{a \in \mathcal{A}(s_t)} \left(\hat{R}(s_t, a) + \gamma \max_{a' \in \mathcal{A}(s_{t+1})} \bar{Q}_{t-1}\left(s_{t+1}, a'\right)\right).$$

Here, the single state condition ($|S| = 1$) is used to remove complicated dependency of $\max_{a' \in \mathcal{A}} \bar{Q}_{t-1}\left(s_{t+1}, a'\right)$ and isolate $\max \hat{R}$ as the source of $\max$-bias. The immediate consequence is as follows:

$$\max_{a' \in \mathcal{A}(s_{t+1})} \bar{Q}_{t-1}\left(s_{t+1}, a'\right) \xrightarrow[state]{single} \max_{a' \in \mathcal{A}} \bar{Q}_{t-1}\left(a'\right) =: \bar{Q}^M_{t-1},$$

where $\bar{Q}^M_{t-1}$ is a deterministic value at iteration t, which is independent from $\max_{a \in \mathcal{A}(s_t)}$ operator, without loss of generality. Now, the sample $\hat{Q}_t$ becomes

$$\begin{aligned}\hat{Q}_t &\leftarrow \max_{a \in \mathcal{A}} \left(\hat{R}(a)\right) + \gamma \bar{Q}^M_{t-1} \\ &= \hat{R}(a_t) + \gamma \bar{Q}^M_{t-1},\end{aligned}$$

where $a_t := \arg\max_{a \in \mathcal{A}} \left(\hat{R}(a)\right) + \gamma \bar{Q}^M_{t-1}$.

Denote the corresponding true reward function $R(\cdot) := \mathbb{E}\hat{R}(\cdot)$, the true best action $a_t^* = \arg\max_{a \in \mathcal{A}}(R(a))$, and the oracle Q sample $\hat{Q}^*$. Then the actual $\max$-bias at iteration t shown in Definition 1 becomes:

$$\begin{aligned}b_t &= \bar{Q}_t - \bar{Q}_t^* \\ &= \alpha_{t-1}\left(\hat{Q}_t - \hat{Q}_t^*\right) \\ &= \alpha_{t-1}\left(\hat{R}\left(a_t\right) - R\left(a_t^*\right)\right) \\ &\le \left(\hat{R}\left(a_t\right) - R\left(a_t^*\right)\right).\end{aligned}$$

To account for unobservable randomness in iteration t, take the expectation (over W_t), with the usual case of using nonnegative stepsizes α_t, we obtain

$$\begin{aligned}\mathbb{E}b_t &= \mathbb{E}\left[\alpha_{t-1}\left(\tilde{R}\left(a_t\right) - R_O\left(a_t\right)\right)\right] \\ &= \alpha_{t-1}\mathbb{E}\left[\left(\tilde{R}\left(a_t\right) - R_O\left(a_t\right)\right)\right] \\ &= \alpha_{t-1}\left(\mathbb{E}\left[\tilde{R}\left(a_t\right)\right] - R_O\left(a_t\right)\right) \\ &= \alpha_{t-1}\left(\mathbb{E}\left[\max_{a \in \mathcal{A}}\left(\hat{R}(a)\right)\right] - R_O\left(a_t\right)\right) \qquad (4) \\ &\ge \alpha_{t-1}\left(\mathbb{E}\left[\max_{a \in \mathcal{A}}\left(\hat{R}(a)\right)\right] - \max_{a \in \mathcal{A}}\left(\mathbb{E}\left[\hat{R}\left(a\right)\right]\right)\right) \\ &\ge 0, \qquad (5)\end{aligned}$$

where to get equation 5, apply Jensen's inequality, $\mathbb{E}\left[\max_{a \in \mathcal{A}}\left(\hat{R}(a)\right)\right] \ge \max_{a \in \mathcal{A}}\left(\mathbb{E}\left[\hat{R}\left(a\right)\right]\right)$. $\mathbb{E}b_t \ge 0$ implies the existence of $\max$-induced bias in iteration $t \le \infty$ as in proposition 2. □

An immediate corollary of proposition 2 is that the $\max$-induced bias is transient, and does not affect the asymptotic convergence of Q-learning.

Corollary 3 (Max-bias is transient). *Proposition 2 implies:*

$$\mathbb{E}b_t \xrightarrow{t \to \infty} 0$$

Proof. The result follows immediately from Proposition 2 and the requirement that $\alpha_t \xrightarrow{t \to \infty} 0$. □

This analysis hints at the importance of the stepsize rule as shown in equation 4. Generally, a larger stepsize normally produces faster initial convergence (if no noise is present), but it also increases the bias. A smaller stepsize reduces the effect of the noise which reduces the bias, but it slows convergence in a process without noise.

Bias-corrected Q-learning

Our bias-corrected form of Q-learning introduces a bias correction term B_{t-1} which produces the following algorithm:

$$\begin{aligned}\bar{Q}_t\left(s_t, a_t\right) &\leftarrow \left(1 - \alpha_{t-1}\left(s_t, a_t\right)\right)\bar{Q}_{t-1}\left(s_t, a_t\right), \\ &\quad + \alpha_{t-1}\left(s_t, a_t\right)\left(\hat{Q}_t - B_{t-1}\left(s_t, a_t\right)\right) \\ \hat{Q}_t &\leftarrow \left(\bar{R}_{t-1}(s_t, a_t) + \gamma \max_{a' \in \mathcal{A}} \bar{Q}_{t-1}\left(s_{t+1}, a'\right)\right).\end{aligned}$$

such that the additive correction term B_{t-1} cancels out the $\max$-induced bias in $\bar{Q}_t$ in conditional expectation given all information up to iteration $t-1$ as

$$\mathbb{E}\left[b_t|\mathfrak{F}_{t-1}\right] := \mathbb{E}\left[\bar{Q}_t - \bar{Q}_t^*|\mathfrak{F}_{t-1}\right] = 0$$

which is equivalent to the following condition

$$\mathbb{E}\left[\hat{Q}_t - B_{t-1}|\mathfrak{F}_{t-1}\right] = \mathbb{E}\left[\hat{Q}_t^*|\mathfrak{F}_{t-1}\right]$$

Our bias correction term is designed to satisfy the above condition, with a model which assumes that we have an infinite number of actions, all with the same distribution of rewards. Our bias correction term is computed as

$$B_{t-1}(s_t, a_t) \leftarrow \left(\frac{\xi}{b} + b\right)\bar{\sigma}_{t-1}^{(s_t, a_t^*)},$$

where the terms used above are defined as

$$\begin{aligned}\bar{R}_{t-1}\left(s, a\right) &:= \frac{1}{N_{t-1}(s, a)} \sum_{i=1}^{t-1}\left(\hat{R}_i \mathbb{I}\left(\left(s, a\right) = \left(s_i, a_i\right)\right)\right), \\ N_{t-1}\left(s, a\right) &:= \sum_{i=1}^{t-1}\left(\mathbb{I}\left(\left(s, a\right) = \left(s_i, a_i\right)\right)\right), \\ b &:= \left(2\log|\mathcal{A}| - \log\log|\mathcal{A}| - \log 4\pi\right)^{\frac{1}{2}}, \\ \xi &:\approx 0.5774 \quad \text{(Euler-Mascheroni constant)}, \\ \bar{\sigma}_{t-1}^{(s_t, a_t^*)} &:= \sqrt{\mathrm{Var}\bar{R}_{t-1}(s_t, a_t^*)}, \\ a_t^* &:= \arg\max_{a \in \mathcal{A}} \bar{Q}_{t-1}\left(s_t, a\right)\end{aligned}$$

We next show the asymptotic behavior of bias-corrected Q-learning using the following theorem:

Theorem 4 (Bias-correction). *Assuming that the distribution of $\hat{R}(a)$ is the same for all a, we have*

$$\lim_{|\mathcal{A}|\to\infty} \mathbb{E}\left[\hat{Q}_t - B_{t-1}|\mathfrak{F}_{t-1}\right] = \mathbb{E}\left[\hat{Q}^*_t|\mathfrak{F}_{t-1}\right]. \quad (6)$$

Proof. We use the setting of a problem with a single state, an infinite number of actions and where the action a is chosen greedily. We further assume that the reward for each action is distributionally the same.

Both sides of equation (6) can be simplified further (we also let $\mathbb{E}_{t-1}[\cdot] = \mathbb{E}[\cdot|\mathfrak{F}_{t-1}]$ for notational simplicity), giving us

$$\mathbb{E}_{t-1}(\hat{Q}_t - B_{t-1}) = \mathbb{E}_{t-1}\left[\max_{a\in\mathcal{A}} \bar{R}_{t-1}(a) + \gamma\bar{Q}^M_{t-1} - B_{t-1}\right],$$

$$\mathbb{E}_{t-1}\hat{Q}^*_t \quad = \quad \mathbb{E}_{t-1}\left[\max_{a\in\mathcal{A}} R(a) + \gamma\bar{Q}^M_{t-1}\right].$$

Using the notation a^*_t for the true best action in a single state MDP, denote $\max_{a\in\mathcal{A}} R(a) =: \mu_{a^*}$ which is a deterministic value since $R(a) := \mathbb{E}\hat{R}(a)$. Then simplifying the above equations results in the following equation which we will prove:

$$\mathbb{E}_{t-1}\left[\max_{a\in\mathcal{A}} \bar{R}_{t-1}(a) - B_{t-1}\right] = \mu_{a^*}. \quad (7)$$

Using definition of B_{t-1} in the left-hand side of equation (7) and simplifying (for notational simplicity, $\bar{R}(\cdot) = \bar{R}_{t-1}(\cdot)$, and $\bar{\sigma}_{a^*} = \bar{\sigma}^{a^*_t}_{t-1}$)

$$\begin{aligned}
&\mathbb{E}_{t-1}\left[\max_{a\in\mathcal{A}}\left(\bar{R}(a)\right) - B_{t-1}\right]\\
&= \mathbb{E}_{t-1}\left[\frac{\bar{\sigma}_{a^*}}{b}\left(b\left(\max_{a\in\mathcal{A}}\left(\frac{\bar{R}(a)-\mu_{a^*}}{\bar{\sigma}_{a^*}}\right) - b\right) - \xi\right) + \mu_{a^*}\right]\\
&= \frac{\bar{\sigma}_{a^*}}{b}\left(\mathbb{E}_{t-1}\left[b\left(\max_{a\in\mathcal{A}}\left(\frac{\bar{R}(a)-\mu_{a^*}}{\bar{\sigma}_{a^*}}\right) - b\right)\right] - \xi\right) + \mu_{a^*}\\
&\xrightarrow[|\mathcal{A}|\to\infty]{} \mu_{a^*}. \qquad (8)
\end{aligned}$$

To get equation (8), apply lemma 5 to the expectation and use the fact that standard Gumbel distribution has mean ξ (Euler-Mascheroni constant). □

Lemma 5. *Given i.i.d. normally distributed random variables $X_1, X_2, \dots X_N$, the following convergence in distribution holds:*

$$b_N\left(\max_{i\in\{1,2,\dots,N\}}\left\{\frac{X_i - \mathbb{E}X_i}{\sqrt{\mathrm{Var}X_i}}\right\} - b_N\right) \xrightarrow[N\to\infty]{d} \mathcal{G}(0,1),$$

where

$$b_N := (2\log N - \log\log N - \log 4\pi)^{\frac{1}{2}}$$

and $\mathcal{G}(0,1)$ is the standard Gumbel distribution.

Proof. The proof uses standard concepts from probability and is omitted.

□

On average the bias correction term cancels out the bias in the worst case (infinite number of actions with identical reward distributions). This gives the underlying Q-learning algorithm robustness against abnormally positively biased samples driving $\bar{Q}$ values up, resulting in temporary divergence which takes many iterations to be removed.

In most cases, the bias correction term overestimates the actual $\max$-induced bias and puts negative bias on the stochastic sample. However, this is a much milder form of bias. In a maximization problem, positive bias takes many more iterations to be removed than negative bias because Q-learning propagates positive bias due to the $\max$ operator over $\bar{Q}$ values.

Application: Intelligent Battery Control

The following model of MDP for an intelligent battery control problem is solved with a bias-corrected Q-learning algorithm.

- Battery size: 10 MWh (10 parallel batteries of 1 MWh).
- Battery charge/discharge rate: can charge or discharge 20% of maximum capacity in 15 minutes. Charge level is discretized in 10% increments.
- Electricity price fluctuation: Poisson process based jump diffusion process modeling. Transition function is calculated by Monte Carlo simulating the price process model.
- Reward function is the expense or the revenue made by charging or discharging.
- Time discretization and discount factor: each timestep is 15 minutes long, and $\gamma = 0.99$ is used to sustain reasonable discount level after 96 timesteps.

Rationale behind the battery model

Different storage devices have different characteristics, among which power and capacity are the most relevant in the application in smart grid application modeling. Electrical storage devices, such as supercapacitors, have very high throughput but small capacity. Storage using potential energy, such as hydro-pump storage or compressed air electricity storage, can have high capacity but its construction requires time and large initial cost. Storage using chemical energy, such as lithium-ion batteries, is considered suitable for different applications in the smart grid, because there are different types of batteries with distinct power-capacity, discharge rates, cycle life, efficiency, and cost (Yang et al. 2011). Attractive candidates among chemical-based devices are lead-acid, lithium-ion and sodium-based batteries for their desirable electric properties and viability of real deployment (USDoE-OE 2011). These types of batteries typically have capacity ranging from 100 kWh to 10 MWh, with typical time to fully discharge in the 4 to 8 hour range (Yang et al. 2011). From these numbers, a unit battery in the simulation is modeled to have 1 MWh capacity and charge/discharge rate of 20% of capacity per time unit.

Electricity price process model

The electricity price process model used in the simulation is a jump diffusion process fitted to a real world data of deseasonalized log price of electricity.

The real world data P_t from the PJM western hub is first transformed into log prices as $Y_t = \log(P_t + c)$ with a constant chosen to ensure the argument of log to be positive. Then Y_t is deseasonalized by subtracting a parametric seasonal element term, Y_t^s, and the remaining deseasonalized portion Y_t^{ds} is fitted to a jump diffusion process. Then, the time interval is discretized to get the following discrete time price process equation:

$$Y_t^{ds} - Y_{t-\Delta t}^{ds} = \alpha\left(\mu - Y_{t-\Delta t}^{ds}\right)\Delta t + \sigma\sqrt{\Delta t}\varepsilon_t + N_t \quad (9)$$

where the term with ε is the Brownian motion term and the term N_t is the jump term whose arrival is modeled with a compound Poisson process. Details of the model and the fitting process can be found in (Cartea and Figueroa 2005).

Experimental procedure

For bias-corrected Q-learning and Q-learning, the following simulation procedure is used. First, a price process of length 3 million samples is created from the price process model in equation (9). We then use the price process to execute the Q-learning and bias-corrected Q-learning algorithms. To compare the outcome of the two algorithms, all other parameters and settings are set the same. Wherever necessary, we performed a number of independent repetitions of the simulation procedures and report mean and standard deviation of the resulting values. Experiment-specific details are provided in each section.

To generate the true values $\bar{V}$ of states that satisfies Bellman equation, we performed exact value iteration using the MDP setting described in the previous section. The discrete probability transition matrix used for exact value iteration is generated from Monte Carlo simulation of the price process model in equation (9). All values of any given state reported as "true value" are obtained in this manner, and used to explicitly show the bias in the estimated values from $\bar{Q}$.

Comparison of estimated values between Q-learning and bias-corrected Q-learning

To illustrate the size and the duration of the bias in the value estimate $\bar{V}$, in figure 2 we plot how $\bar{V}$ for a single state changes over the number of samples used to learn $\bar{Q}$. The value estimate $\bar{V}$ of a state s is calculated as $\bar{V}_Q(s) = \arg\max_{a\in\mathcal{A}(s)} \bar{Q}(s,a)$, using the $\bar{Q}$'s from bias-corrected Q-learning and Q-learning. The snapshots of $\bar{Q}_t$ are used to calculate $\bar{V}$ for different value of the iteration counter t, which corresponds to the number of samples used.

Comparison of mean squared bias for states in empirical convergence

In practical cases where computing the true value is not feasible, the convergence of $\bar{Q}$ values or value estimates $\bar{V}$ are often decided by testing the change in the estimates remains smaller than a pre-defined threshold value. Emulating the situation, we impose an arbitrary empirical condition for testing convergence, and report the size of bias remaining in the value estimates that satisfied the empirical convergence criterion. The estimate $\bar{V}(s)$ for a given state s is considered empirically converged at iteration t, if $|\bar{V}_t(s) - \bar{V}_{t-1}(s)| < 10$. At a given iteration t, we denote $\mathcal{S}_{conv,t}$ as a set containing all states that satisfies the criterion and report the evolution of mean-squared bias (MSB) of all empirically convergent states. MSB at iteration t is calculated as

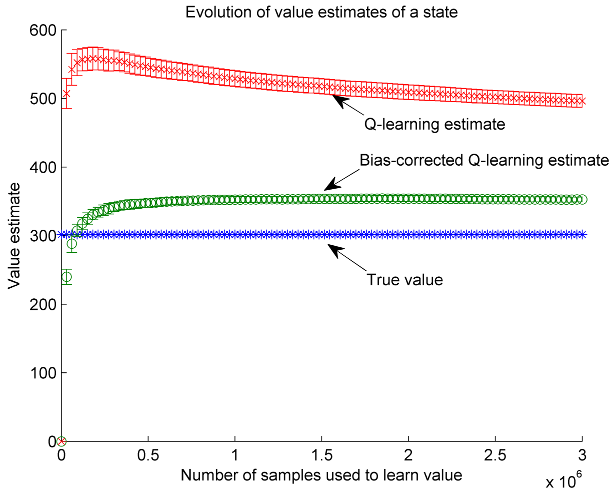

Figure 2: A plot of value estimates of a state (with the battery charged at 50% of capacity and the electricity price of $20 per MWhr), from Q-learning and bias-corrected Q-learning as well as the true value from exact value iteration. Bias in the estimated value is the difference between the estimate and the true value. The estimate from bias-corrected Q-learning shows significantly less bias than that from Q-learning. 50 independent repetitions are performed to compute the mean and the standard deviation values shown.

$$\mathrm{MSB}(t) = \frac{1}{|\mathcal{S}_{conv,t}|}\sum_{s\in\mathcal{S}_{conv,t}}\left(\bar{V}_t(s) - V^*(s)\right)^2$$

where $\bar{V}_t(s)$ is the value estimate of state s calculated from the estimate of $\bar{Q}_t$, and $V^*(s)$ is the true value of state s.

In figure 3, we report the evolution of MSB over the number of samples (or iteration counter t) used to learn $\bar{Q}_t$ for both bias-corrected Q-learning and Q-learning using a harmonic stepsize rule. We also report the result with a constant stepsize rule to account for applications where users prefer a constant stepsize rule to deal with non-stationarity.

Sum of rewards using policy learned

To illustrate how successfully bias-corrected Q-learning learns in the presence of max-induced bias, we report the sum of rewards of using the learned policy to control battery, where the policy is learned from solving the intelligent battery control problem modeled as a MDP.

To calculate the sum of rewards from using the learned policy given $\bar{Q}$ estimates, the following procedure is used. From the estimates of $\bar{Q}_t$ taken at any given iteration t of Q-learning and bias-corrected Q-learning, we construct a deterministic policy as follows: $\pi_t(s) =$

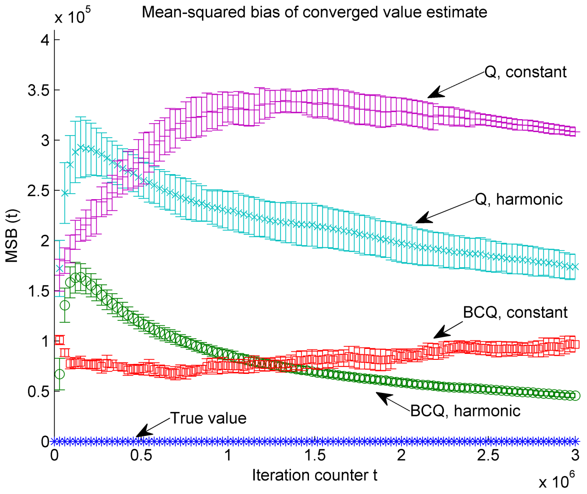

Figure 3: The evolution of mean squared bias for states with empirical convergence, for different learning algorithms and stepsize rules. At any given iteration counter t, the effect of max-induced bias is much more significant for Q-learning than bias-corrected Q-learning, with both harmonic and constant stepsize rules, as illustrated by significantly greater positive deviation from the value estimates from the true value. 50 independent repetitions are performed to compute the mean and the standard deviation values shown.

$\arg\max_{a\in\mathcal{A}(s)} \bar{Q}(s,a)$. Then, we simulate a Markov decision process using an independently generated price process $\{p_1, p_2, \ldots p_N\}$ where $N = 30 \times 96 = 2880$ corresponding to the number of 15-minute-long timesteps over 30 days. At any given simulation iteration n, given a policy π_t, the decision a_n is made using the policy to obtain the corresponding reward as $R_n^{\pi_t} = p_n a_n = p_n \pi_t(r_n, p_n)$.

In figure 4, we report the undiscounted sum of rewards $\sum_{n=1}^{N} R_n^{\pi_t}$ over a range of t up to 3 million samples. The figure shows clearly that the bias-corrected Q-learning policy outperforms the conventional Q-learning policy, with much faster convergence.

Conclusion

Bias corrected Q-learning asymptotically cancels the worst case bias due to the $\max$-operator in Q-learning. In the simulated intelligent battery control problem, bias-corrected Q-learning shows significantly less transient bias in its estimated $\bar{Q}$ than Q-learning. The estimate of $\bar{Q}$ of bias-correct Q-learning shows significant bias correction compared to the estimate of Q-learning even when empirical convergence criteria is used to select subsets of states whose value estimate is deemed stable enough. In addition to reductions in the bias in value estimation, bias-corrected Q-learning learns better than Q-learning, as shown by the learned policy that results in significantly better performance measured in undiscounted finite-horizon sum of reward than that from Q-learning, especially in the earlier iterations where the noise in the estimates of the Q-factors produces the greatest bias.

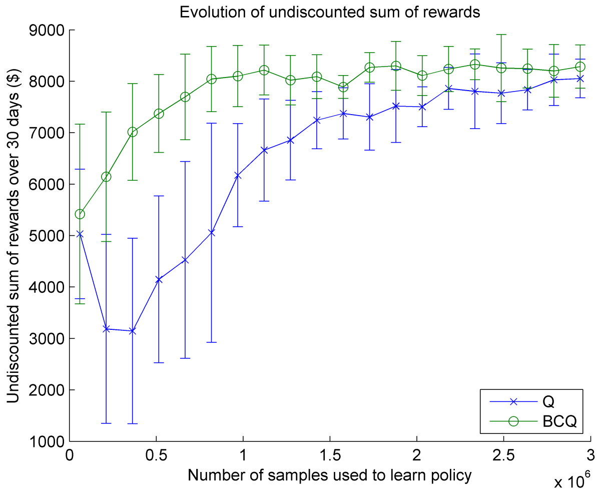

Figure 4: The undiscounted sum of rewards using policies from bias-corrected Q-learning and Q-learning, plotted against the number of samples taken to learn the policies. The learned policy from Q-learning suffers from the transient bias initially (up to 1 million samples), and requires 2 million additional samples to recover to the level where bias-corrected Q-learning reaches before 1 million samples. In the contrary, the learned policy from bias-corrected Q-learning does not show the degraded performance in the initial samples, reflecting its robustness against the transient bias in Q-learning estimates. The reported mean and standard deviation are calculated from 50 independent repetitions.

Acknowledgments

This research was supported in part by grant AFOSR contract FA9550-08-1-0195 and the National Science Foundation grant CMMI-0856153. Many thanks to Ilya O. Ryzhov, Warren R. Scott, and Ramon van Handel for their discussion and assistance in the research project.

References

Barton, J. P., and Infield, D. G. 2004. Intermittent Renewable Energy. *IEEE Transactions on Energy Conversion* 19(2):441–448.

Brown, K. C. 1974. A note on the apparent bias of net revenue estimates for capital investment projects. *The Journal of Finance* 29(4):1215–1216.

Cartea, A., and Figueroa, M. G. 2005. Pricing in Electricity Markets : A Mean Reverting Jump Diffusion Model with Pricing in Electricity Markets : A Mean Reverting Jump Diffusion Model with Seasonality. *Applied Mathematical Finance* 12(4):313–335.

da Motta Salles Barreto, A., and Anderson, C. W. 2008. Restricted gradient-descent algorithm for value-function approximation in reinforcement learning. *Artificial Intelligence* 172(4-5):454 – 482.

Eydeland, A., and Wolyniec, K. 2003. *Energy and Power Risk Management*. Hoboken, NJ: John Wiley and Sons.

Huggins, R. A. 2010. *Energy storage*. New York: Springer.

Jaakkola, T.; Jordan, M.; and Singh, S. P. 1994. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation* 1–31.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237–285.

Kim, J. H., and Powell, W. B. 2011. An hour-ahead prediction model for heavy-tailed spot prices. *Energy Economics* 33:1252–1266.

Ormoneit, D., and Sen, S. 2002. Kernel-based reinforcement learning. *Machine Learning* 49:161–178. 10.1023/A:1017928328829.

Powell, W. B. 2007. *Approximate Dynamic Programming: Solving the curses of dimensionality*. Hoboken, NJ: John Wiley & Sons.

R. Huisman, R. M. 2003. Regime Jumps in Electricity Prices. *Energy Economics* 25:425–434.

Reynolds, S. I. 2001. Optimistic initial q-values and the max operator. University of Edinburgh Printing Services.

Smith, J. E., and Winkler, R. L. 2006. The optimizers curse: Skepticism and postdecision surprise in decision analysis. *Management Science* 52(3):311–322.

Thurn, S., and Schwartz, A. 1993. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*. Lawrence Erlbaum Publisher.

Tsitsiklis, J. N. 1994. Asynchronous stochastic approximation and Q-learning. *Machine Learning* 16:185–202.

USDoE-OE. 2011. Energy storage program planning document.

van Hasselt, H. P. 2010. Double q-learning. In *Advances in Neural Information Processing Systems*, volume 23.

Yang, Z.; Zhang, J.; Kintner-Meyer, M. C. W.; Lu, X.; Choi, D.; Lemmon, J. P.; and Liu, J. 2011. Electrochemical energy storage for green grid. *Chemical Reviews* 111(5):3577–3613.