Submitted to *Operations Research*
manuscript (Please, provide the mansucript number!)

Approximate Dynamic Programming for Energy Storage with New Results on Instrumental Variables and Projected Bellman Errors

Warren R. Scott
Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544,
wscott@princeton.edu

Warren B. Powell
Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544,
powell@princeton.edu

We address the problem of developing near-optimal policies for an energy system that combines energy from an exogenously varying supply (wind) and price (the grid) to serve time dependent and stochastic loads in the presence of a storage device. The goal is to consider three different types of uncertainty in a stochastic control problem that arises when using energy from renewables. With a few simplifications, the problem can be solved optimally using classical dynamic programming methods, but the full problem can only be solved approximately. We compare several approximate dynamic programming algorithms, including approximate policy iteration using least-squares Bellman error minimization, Bellman error minimization using instrumental variables, and least-squares projected Bellman error minimization. We show for the first time that Bellman error minimization using instrumental variables is mathematically equivalent to projected Bellman error minimization, previously thought to be fundamentally different algorithmic strategies. We show that Bellman error minimization using instrumental variables, implemented in an approximate policy iteration algorithm, significantly outperforms classical least-squares policy iteration, but underperforms direct policy search. All of these are tested using realistic data, and are compared against optimal benchmarks.

Key words: energy storage, electricity market, wind turbines, approximate dynamic programming, dynamic programming, Bellman error minimization, projected Bellman error minimization, instrumental variables, approximate policy iteration, direct policy search

1. Introduction

Incorporating large amounts of energy from intermittent resources into the power grid creates many complications due to both variability and uncertainty. For example, if the wind power in the system drops suddenly, expensive ancillary services are required to satisfy the load. We also have

to deal with electricity prices including time-varying contract prices as well as highly volatile spot prices. We need to manage our system to meet a time-varying load which has its own sources of uncertainty due to weather. Drawing on power from wind requires that we deal with an exogenously varying supply that introduces both short-term volatility with a daily cycle which is out of sync with loads. An electricity storage device can be used to mitigate the effects of the intermittency and uncertainty of wind as well as providing other services to a grid operator. Potential uses for an electricity storage device include electricity price arbitrage, generation capacity, ancillary services, transmission support, electricity service reliability, time-of-use energy cost management, regulation of energy production from renewables, and time-shifting of renewable energy (see Eyer et al. (2004)).

Many recent papers discuss the benefits of combining energy storage devices with renewables. Costa et al. (2008) describes a virtual power plant which uses a dynamic programming algorithm to operate an energy storage facility and a wind farm. Sørensen (1981) describes the potential benefits of combining wind power with hydro storage. Greenblatt et al. (2007) and Swider (2007) discuss combining wind with compressed air energy storage. Sioshansi et al. (2009) investigates the potential value of a storage device in the PJM network used for arbitrage. Mokrian and Stephen (2006) uses stochastic programming to operate a storage device which buys and sells in an electricity spot market. Kempton and Tomic (2005) discusses the value of the ability of electric vehicles to provide peak power, spinning reserves, and regulation. Zhou et al. (2011) examine a dual-threshold policy for a wind, storage, and transmission system. Lai et al. (2010) discusses how approximate dynamic programming can be used to bound the value of natural gas storage, and Secomandi (2010) derives optimal policies for storing natural gas under certain assumptions on natural gas prices. Carmona and Ludkovski (2005) uses stochastic impulse control to operate a gas storage device. A thorough review of this growing literature is beyond the scope of this paper.

We address the problem of optimally controlling the power flows among a source with intermittent supply, a grid which offers infinite supply at a variable price, and a variable load in the presence of a storage device. A byproduct of this research will be the ability to estimate the economic value

of storage for both long term investment as well as day-ahead tactical planning. The answers to these questions require knowing how the energy storage device will be used. In general, deciding how to charge and discharge the storage device is a difficult problem to solve optimally due to the uncertainty in the wind, electricity prices, and electricity demand.

The primary contribution of the paper is the development of high quality, scalable algorithms for the near-optimal control of an energy storage device in the presence of complicating side variables such as prices, loads, and energy from renewables. We develop an optimal benchmark for a simplified problem and use this to evaluate an approximate policy iteration algorithm using least-squares policy iteration, an algorithmic strategy with strong theoretical support. We demonstrate the importance of using instrumental variables in this algorithmic strategy (see Durbin (1954); Kendall and Stuart (1961); Söderström and Stoica (1983); Bowden and Turkington (1984)). Recent research has also focused attention on the use of projected Bellman error minimization. We show for the first time that this is mathematically equivalent to Bellman error minimization when instrumental variables. Despite the strong theoretical support enjoyed by this algorithmic strategy, we also show that direct policy search still produces much better policies.

This paper is organized as follows. Section 2 gives a description and model of wind, electricity prices, electricity demand, and energy storage. Section 3 sets up the dynamic program that combines stochastic wind, stochastic electricity prices from the grid, and an energy storage device to satisfy a stochastic load. Section 4 summarizes approximate policy iteration for solving the dynamic program. Within policy iteration, we focus on several policy evaluation algorithms based on minimizing Bellman error: (1) instrumental variables Bellman error minimization, (2) least-squares projected Bellman error minimization, (3) instrumental variables projected Bellman error minimization. We show that these three policy evaluation algorithms are equivalent under certain full rank assumptions and converge when using off-policy sampling under certain conditions. Section 5 describes an alternative strategy to fit the parameters of a value function approximation using direct policy search. Finally, in Section 6 we analyze the performance of the approximate dynamic programming policies on a series of simplified, discretized problems for which we have obtained an

optimal benchmark, and then on the full, multidimensional problem with continuous variables. A byproduct of this research is a set of benchmark problems which can be used by the algorithmic community to test approximate algorithms with an exact solution and finally the full model.

2. Models

We wish to address the problem of combining power from the grid with stochastic prices, wind with stochastic supply, and storage to meet a stochastic demand for electricity as shown in Figure 1. We begin by describing the models we use for wind, electricity prices, electricity demand, and energy storage.

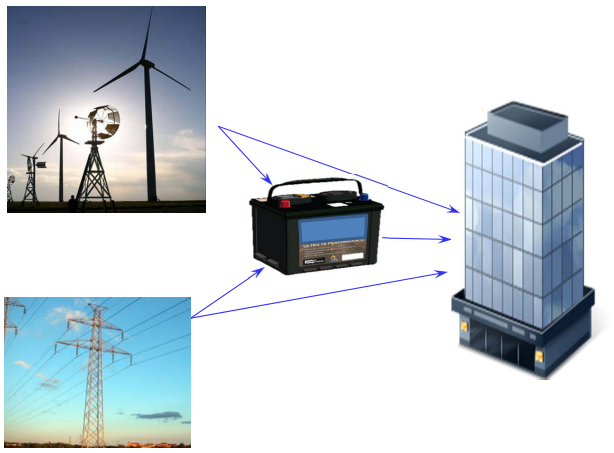


Figure 1 The energy flow diagram.

2.1. Wind

Brown et al. (1984) suggests modeling the square root of the wind speed with an autoregressive (AR) model, while Chen et al. (2010) suggests using a more general ARIMA model. Let W_t be the wind speed in (m/s). We define Y_t to be the de-meaned square root of the wind speeds; $Y_t = \sqrt{W_t} - \mathbb{E}[\sqrt{W_t}]$. We use the wind speeds at Maryneal, TX every fifteen minutes to fit an AR model to Y_t. For the purpose of keeping the state space small we use an AR(1),

$$Y_t = \phi_1 Y_{t-\Delta t} + \epsilon_t, \tag{1}$$

where $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Using the Yule-Walker equations (see Carmona (2004)) and setting $\Delta t = 15$ minutes, we find the following estimates: $\mathbb{E}[\sqrt{W_t}] = 1.4781$; $\phi_1 = 0.7633$; $\sigma_\epsilon = 0.4020$. Now we can simulate Y_t and then transform back to the corresponding wind speed W_t. Once we have the wind speed we can convert to the power produced by a wind turbine using a typical power curve equation (see Burton et al. (2001); Anaya-Lara et al. (2009)),

$$P_t = .5 C_p \rho A {W_t}^3. \tag{2}$$

Here, C_p is the power coefficient that is less than the Bentz limit of .593 (corresponding approximately to $C_p = .45$), ρ is the density of air ($\rho = 1.225 kg/m^3$). A is the area swept by the rotor blades of the turbine ($A = \pi 50^2 m^2$ for a typical turbine), W_t is the velocity of the wind in m/s, and P_t is the power output from the turbine in watts (1 watt = 1 kg $\cdot m^2/s$). Typically there is a cut-in wind speed that is the minimum speed necessary to produce power, a rated wind speed (beyond which the wind turbine does not produce any extra power), and, finally, a very large speed called the cut-out wind speed, above which the turbine must be shut off.

2.2. Electricity Prices

In the PJM day-ahead market, PJM receives offers and bids for the next operating day, and at 4pm the day-ahead prices are determined with the scheduling, pricing, and dispatch program. In addition, there is an hourly real-time (spot) market that has even more extreme prices than the day-ahead market. The real-time prices at the PJM Western Hub average $42.11 per MWh over 2009-2010, although the prices are occasionally negative and have a maximum of $362.90 per MWh. Figure 2 shows that the prices are lowest at night; they begin to increase around 5am and are typically the highest in the evening around 6pm.

We fit a jump diffusion process to the deseasonalized real-time electricity prices (see Cartea and Figueroa (2005)). We first take the electricity prices, P_t, and convert to log prices,

$$Y_t = log(P_t + c). \tag{3}$$

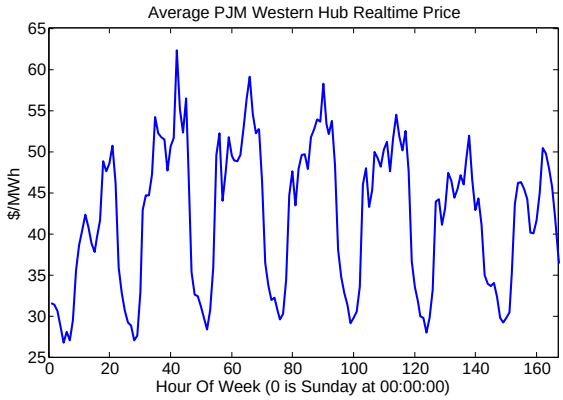


Figure 2 The average PJM Real-Time price at the Western Hub as a function of the hour of the week.

In Equation (3), we add a constant c before taking the natural log to ensure we do not take the log of a negative number (we set c to one minus the minimum value of P_t). We next calculate the deseasonalized log electricity prices, $Y_t^{ds} = Y_t - Y_t^s$, where Y_t^s is the seasonal component and is a deterministic periodic function of t. When calibrating Y_t^s, we use an hour of week and month of year component. We then fit a jump diffusion process to the deseasonalized log prices,

$$dY_t^{ds} = \lambda(\mu - Y_t^{ds})dt + \sigma dW_t + dN_t, \tag{4}$$

where μ is the long term equilibrium price, λ is the mean reversion rate, W_t is a Brownian motion, and N_t is the jump process. Discretizing, we can write

$$Y_t^{ds} - Y_{t-\Delta t}^{ds} = \lambda(\mu - Y_{t-\Delta t}^{ds})\Delta t + \sigma\sqrt{\Delta t}\epsilon_t + J_t, \tag{5}$$

where $\{\epsilon_{t+n\Delta t}\}_{n=0}^N$ are i.i.d. standard normal random variables, and J_t is the jump over the interval $(t-\Delta t, t]$. If we were to model the jumps with a compound Poisson process, we could write $J_t = \sum_{k=0}^{X_t - X_{t-\Delta t}} J_{t,k}$, where X_t is a Poisson process with intensity $\lambda^{Poisson}$ (hence the number of arrivals $X_t - X_{t-\Delta t} \sim \text{Poisson}(\lambda^{Poisson}\Delta t)$). However, for calibration purposes, Cartea and Figueroa (2005) models the jumps as the i.i.d. process,

$$J_t = \epsilon_t^{jump}\mathbf{1}(U_t < p^{jump}), \tag{6}$$

where ϵ_t^{jump} is the size of a jump, $U_t \sim \text{unif}(0,1)$, and p^{jump} is the probability of a jump over a time interval of Δt. We identify the nonzero jumps as in Cartea and Figueroa (2005) by locating times where the absolute value of the return is more than three times the standard deviation of the returns. We can then fit p^{jump} as the fraction of time jumps occur (we divide this estimate by two because most jumps are immediately followed by jumps in the opposite direction). In addition, we model $\{\epsilon_{t+n\Delta t}^{jump}\}_{n=0}^{N}$ as i.i.d. normal random variables with mean zero and standard deviation σ^{jump}.

At this point we can obtain estimates of λ, μ, and σ using least-squares linear regression on Equation (5); $Y_t^{ds} - Y_{t-\Delta t}^{ds}$ are the observations, and $\sigma\sqrt{\Delta t}\epsilon_t + J_t$ are the centered residuals. The variance of the residuals is,

$$\text{Var}(\sigma\sqrt{\Delta t}\epsilon_t + J_t) = \sigma^2\Delta t + \text{Var}(J_t) = \sigma^2\Delta t + p^{jump}(\sigma^{jump})^2, \tag{7}$$

which gives an equation which can be used for estimating σ.

2.3. Electricity Demand

Eydeland and Wolyniec (2003) outlines typical models for residential, commercial, and industrial power demand. Industrial power demand is relatively stable while residential power demand is highly dependent upon the temperature. For example, Pirrong and Jermakyan (2001) models the load with a reflected Brownian motion that incorporates a seasonal component. Feinberg and Genethliou (2005) summarizes the main approaches to forecasting load such as an end-use model that incorporates appliances and customers, various regression models (based on temperature, time, and other factors), time series, and heuristics made by experts. Eydeland and Wolyniec (2003) prefers the method of modeling the load as a function of temperature; additional factors could be used such as the temperature-humidity index and wind chill index (see Feinberg and Genethliou (2005)).

We use the actual total ERCOT energy loads every hour over 2010 (we can convert this to power by assuming the power consumption is constant over a time interval and using $E = P\Delta t$). The load clearly exhibits some hourly and daily features as shown in Figure 3.

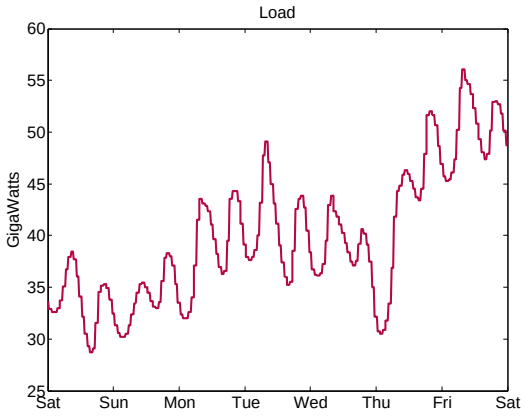


Figure 3 The total ERCOT load over the first full week of 2010.

In general, the ERCOT load starts ramping up in the morning around 5am and peaks in the evening around 6pm, although the patterns vary greatly based on the day of the week and the month of the year. We can deseasonalize the loads and then analyze the residuals of the loads. We write the deterministic seasonal component of the load, m_t, as the sum of an hour-of-week and monthly component. For any time, t, the hour of the week is an integer from $0, 1, ..., 167$; zero corresponds to Sunday morning between 00:00:00 and 00:59:59, and 167 corresponds to Saturday night between 23:00:00 and 23:59:59 military time. To calculate the seasonal component, we calculate the average load over each of the hours of the week and call this the hour-of-week seasonal component, m_t^{hour}. We take the residuals and calculate the average load over each of the months and call this the month-of-year seasonal component, m_t^{month}. The residuals are then called the deseasonalized load, D_t^{ds}. We can write the decomposed load as,

$$D_t = m_t + D_t^{ds}, \tag{8}$$

where the seasonal component m_t is defined as

$$m_t = m_t^{hour} + m_t^{month}.$$

2.4. Energy Storage

Brunet (2011) explains that stationary lead-acid batteries with tubular plates need a small amount

of maintenance and can last up to 12 years when operated properly, costing approximately $300 per kWh of capacity. They typically have a round trip efficiency of 70% and a self-discharge rate of 2% per month and should not be discharged below 20% of their capacity (see Brunet (2011)). The lifespan of the batteries can be maximized by limiting the depth of discharge to 15% per day. A typical lead-acid battery may have a C/10 maximum discharge rate, meaning it can be fully discharged over 10 hours using the maximum discharge rate (see DOE Handbook (1995)). In our work, we do not consider the effect of the storage rate on storage efficiency, as governed by Peukert's Law (see Baert and Vervaet (1999)); this would be a nice extension for handling lead-acid batteries, but is beyond the scope of our presentation.

3. Dynamic Programming Problem

We now address the problem of combining power from the grid with a stochastic price, wind with a stochastic supply, and storage to meet a stochastic demand for electricity. We call D_t the total energy demand (in MWh) over the time period starting at $t - \Delta t$ and ending at t. This energy demand must be met at every time period from either wind energy, energy from the battery, or energy from the grid. We fix a time step, Δt, of fifteen minutes. The full model is described below.

3.1. State Variable

The state variable, S_t, the fraction of the storage that is full (R_t), the current amount of wind energy in MWh (E_t), the current energy demand in MWh (D_t), and the current spot price of electricity to and from the grid in $\$/MWh$ (P_t). We solve both steady-state and time-dependent applications. For time-dependent problems, we also include time t in the state variable. We can write $S_t = (R_t, E_t, D_t, P_t)$.

3.2. Decision Variables

For a fixed time t, the flows in Figure 1 can be represented by the vector $\{x_t^{WR}, x_t^{GR}, x_t^{RD}, x_t^{WD}, x_t^{GD}\}$, where W refers to wind, R refers to the battery resource, G refers to

the grid, and D refers to the demand. At the wind node, the wind energy must either go to the storage or to the demand (we assume the storage can dissipate energy if necessary),

$$x_t^{WR} + x_t^{WD} = E_t^{wind}.$$

At the demand node, the energy demand is satisfied by the grid, the storage, and the wind,

$$D_t = x_t^{GD} + \eta^{discharge} x_t^{RD} + x_t^{WD}.$$

Now we define the constants ΔR^{min} and ΔR^{max} as the minimum and maximum fraction of the storage you can charge over Δt (negative values correspond to discharging). For example, if we have a lead acid battery with a $C/10$ maximum charge and discharge rate, and $\Delta t = 15$min, then $\Delta R^{min} = -1/40$ and $\Delta R^{max} = 1/40$. Now, the feasible actions must satisfy,

$$\frac{\Delta R^{min} R^{capacity}}{\eta^{discharge}} \leq x_t^{GR} \leq \frac{\Delta R^{max} R^{capacity}}{\eta^{charge}}, \tag{9}$$

$$0 \leq x_t^{RD} \leq \Delta R^{max} R^{capacity}. \tag{10}$$

Equation (9) ensures that we do not charge or discharge the storage device faster than the storage device allows. In Equation (9) we could require $0 \leq x_t^{GR}$ if we did not want to allow selling from the storage to the grid. Equation (10) guarantees that we do not discharge the storage device faster than allowed when sending energy from the storage to demand. In our problem the demand must always be satisfied so it is easy to see how to optimally use the wind energy. We send as much wind as possible to demand, and the remaining wind is sent to the storage device for future use,

$$x_t^{WD} = min(E_t^{wind}, D_t), \tag{11}$$

$$x_t^{WR} = E_t^{wind} - x_t^{WD}, \tag{12}$$

$$x_t^{GD} = D_t - \eta^{discharge} x_t^{RD} - x_t^{WD}. \tag{13}$$

Equations (12) and (13) are the flow constraints at the wind and demand node. Equations (11), (12), and (13) effectively reduce the size of our action space from 5 dimensions to 2 dimensions. In addition we require that $R_{t+\Delta t} > 0$ because the battery cannot go negative (in the case of lead-acid batteries we require $R_{t+\Delta t} > .2$ to prevent the battery from becoming fully discharged).

3.3. Exogenous Information Process

We define the exogenous information process as the random changes in the state of the system, $W_{t+\Delta t} = \{\hat{E}_{t+\Delta t}, \hat{D}_{t+\Delta t}, \hat{P}_{t+\Delta t}\}$, which refer to exogenous changes in the energy from the wind E_t, loads D_t and electricity spot prices P_t. These exogenous changes may be state dependent as well as time dependent.

3.4. State Transition

We write the state transition function as, $S_{t+\Delta t} = S^M(S_t, x_t, W_{t+\Delta t})$. The updated state variables can be written,

$$\begin{aligned} E_{t+\Delta t} &= E_t + \hat{E}_{t+\Delta t}, \\ D_{t+\Delta t} &= D_t + \hat{D}_{t+\Delta t}, \\ P_{t+\Delta t} &= P_t + \hat{P}_{t+\Delta t}. \end{aligned}$$

We assume extra energy can be dissipated at the storage device, and our next resource state can be computed,

$$R_{t+\Delta t}(x_t) = min\left(\frac{R_t R^{capacity} + \eta^{charge}(x_t^{GR} + x_t^{WR}) - x_t^{RD}}{R^{capacity}}, 1\right).$$

3.5. Contribution and Objective

The contribution function is simply the dollar value of energy sold minus the amount bought from the grid,

$$C(S_t, x_t) = P_t D_t - P_t(x_t^{GR} + x_t^{GD}).$$

We consider the ergodic infinite horizon problem where the goal is to find the policy, $X^\pi(S_t)$, which maximizes the expected discounted future rewards,

$$\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t C(S_t, X^\pi(S_t))\right]. \tag{14}$$

In our model, the policy $X^\pi(S_t)$ may be stationary (where the *function* does not vary over time) or time-dependent, as might occur when the function depends on the time of day. Time-dependent

functions may be written using $X_t^\pi(S_t)$, or by modifying the state variable to include time (from which we might compute hour of day).

4. Approximate Policy Iteration

The objective function (Equation (14)) can be solved approximately using several algorithmic strategies such as rolling horizon procedures (also known as model predictive control), stochastic programming, or some form of parameterized policy, but we are going to focus on policies based on value function approximations. To do this, we start with Bellman's optimality equation, which allows us to write

$$V_t(S_t) = \max_x C(S_t, x) + \gamma \mathbb{E}\{V_{t+1}(S_{t+1})|S_t\},$$

where $S_{t+1} = S^M(S_t, x, W_{t+1})$ and the expectation is over the random variable W_{t+1}. Since the state variable is multidimensional and continuous, this equation cannot be solved exactly, and as a result a large field of research has evolved to develop approximations of the value function (see Powell (2011); Bertsekas and Tsitsiklis (1996); Sutton and Barto (1998); Puterman (1994)).

We focus on the most widely studied class of approximations that uses a linear model with pre-specified basis functions. We further take advantage of the strategy of using a post-decision state variable, denoted S_t^x, which is the state immediately after a decision but before any exogenous information has been revealed. For our problem, the post-decision state is given by

$$S_t^x = (R_{t+\Delta t}(x), E_t, D_t, P_t).$$

We then approximate the value function using the general form

$$\bar{V}_t(S_t^x) = \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t^x), \tag{15}$$

where $(\phi_f(s))_{f \in \mathcal{F}}$ is a user-specified set of features.

In approximate policy iteration there is typically an outer loop called policy improvement along with an inner loop where policy evaluation is run for a fixed policy. For approximate policy iteration algorithms, the policy improvement loop is fairly standard, but there are many variations of the policy evaluation algorithm.

In the remainder of this section, we review a class of algorithms based on a concept known in the reinforcement learning community known as least squares temporal difference (LSTD) learning. We review the theory, and then describe the algorithm based on Bellman error minimization. We next present two variants: the first uses the concept of projected Bellman error minimization, and the second uses instrumental variables, after which we demonstrate that these two methods are equivalent.

4.1. Theory

Many approximate dynamic programming algorithms can be classified under the category of projected equation methods (see Goodwin and Sin (1984); Watkins (1989); Bradtke and Barto (1996); Koller and Parr (2000); Lagoudakis and Parr (2003); Bertsekas (2011)). Much of the literature focuses on finite states, and typically these methods attempt to find the best value function within a class of value functions. For example, if a linear model (also called linear architecture) is used to approximate the value function, the objective may be to find the weights which minimize the L_2 norm of the Bellman residual. We focus on the approximating the post-decision value function (Lagoudakis and Parr (2003) approximates the pre-decision value function). Letting $\mathcal{S}^x$ be the post-decision state space, this can be written

$$\min_{\theta} \sum_{s \in \mathcal{S}^x} \left(V(s) - \sum_{f \in \mathcal{F}} \theta_f \phi_f(s) \right)^2 = \min_{\theta} \|V - \Phi\theta\|_2^2, \tag{16}$$

where Φ is a matrix of fixed basis functions (each row corresponds to a state, and each column corresponds to a basis function), and θ is a column vector of weights.

Using a linear model for the value function, Bradtke and Barto (1996) presents the least-squares temporal difference learning algorithm for the policy evaluation of a fixed policy which will be presented below. The weights chosen with the least-squares approach will converge with probability one to the true weights if the correct basis functions are used (the true value function happens to be in the span of the basis functions) and a few other technical conditions are satisfied.

Also assuming finite states and actions, Lagoudakis and Parr (2003) introduces least-squares policy iteration which approximates the value of state-action pairs (Q-factors) with a linear model.

When doing policy evaluation, they choose to use least-squares to minimize the fixed-point approximation error instead of the Bellman residual. The paper references the approximate policy iteration theory from Bertsekas and Tsitsiklis (1996) which bounds the L_∞ norm of the difference between the true and approximated Q-factors.

Bradtke and Barto (1996) explains that TD(λ) uses information inefficiently relative to the Least-Squares approach to TD policy evaluation (LSTD). The LSTD policy evaluation algorithm described in Bradtke and Barto (1996) is an on-policy algorithm which approximates the true value function with a linear model with fixed basis functions. The algorithm uses instrumental variables to obtain an estimate of the value function which converges with probability one as the number of transitions increases to infinity.

Lagoudakis and Parr (2003) expands upon the LSTD algorithm from Bradtke and Barto (1996) by using a linear architecture to approximate the value function over the higher dimension state-action pairs. Furthermore, they give the geometric interpretation of several different methods of approximately solving Bellman's equation. Once the value function in Bellman's equation has been replaced by a linear model, Bellman's equation is typically an over-determined system which cannot be solved exactly. When solving for the weights of the value function, the Bellman residuals can be minimized in a least-squares or weighted least-squares sense (Bellman error minimizing approximation). An alternative approach is to project the Bellman residuals down into the space spanned by the basis functions of the value function and then minimize the Bellman residuals. Lagoudakis and Parr (2003) explains that in general the approximate value function is a fixed point of the projected Bellman operator, not the Bellman operator (see De Farias and Van Roy (2000) for a nice discussion).

4.2. Algorithm

We first summarize approximate policy iteration based on Bellman error minimization (see Bradtke and Barto (1996), Lagoudakis and Parr (2003)). We use a modified version of Bellman's equation

based on the post-decision state variable (see Powell (2011), Bertsekas (2011)). Typically, Bellman's equation for an infinite horizon problem is written around the pre-decision value function,

$$V(S_t) = \max_x \mathbb{E}[C(S_t, x) + \gamma V(S_{t+1})|S_t]. \tag{17}$$

The post-decision state, S_t^x, is the state immediately after being in the pre-decision state S_t and taking the action x, but before you observe the randomness from the state transition or receive the contribution (see Powell (2011) [Chapter 4] for a thorough discussion of post-decision states). The post-decision value $V^x(S_t^x)$ is the value of being in post-decision state S_t^x and is defined as $V^x(S_t^x) = \mathbb{E}[V(S_{t+1})|S_t^x]$. Equation (17) can be written as

$$V(S_t) = \max_x \{C(S_t, x) + \gamma V^x(S_t^x)\}.$$

Using only post-decision states, Bellman's equation can be written as

$$V^x(S_{t-1}^x) = \mathbb{E}[\max_x \{C(S_t, x) + \gamma V^x(S_t^x)\}|S_{t-1}^x]. \tag{18}$$

In addition to bringing the expectation outside of the maximum in Bellman's equation, the post-decision value function has the advantage that the post-decision state is often of lower dimension than the pre-decision state.

Next, we replace the post-decision value function with a parametric linear model, $V^x(S_t^x) = \phi(S_t^x)^T\theta$, where $\phi(\cdot)$ is a column vector of pre-determined, real-valued basis functions, $\phi_1(\cdot), ..., \phi_k(\cdot)$, and θ is a column vector of weights for the basis functions. Plugging this approximation into Equation (18) for a fixed policy π we get

$$\phi(S_{t-1}^x)^T\theta = \mathbb{E}[C(S_t, X^\pi(S_t|\theta)) + \gamma\phi(S_t^x)^T\theta|S_{t-1}^x]. \tag{19}$$

If we could find a value of the θ where this equation were exactly satisfied for all states, we would have the true value function for the policy $X^\pi(S_t|\theta)$. In general, we are only able to find a value of θ which approximately solves Equation (19). We outline the approximate policy iteration algorithm in Figure 4 which combines an inner loop which performs policy evaluation for a fixed policy with an outer loop which improves the policy. We now summarize several techniques for finding θ which approximately solves Equation (19).

4.3. Policy Evaluation using Bellman Error Minimization

We draw on the foundation provided in Bradtke and Barto (1996), but adapted for the post-decision state in Ma and Powell (2010). We focus on the off-policy case where a set of post-decision states, $\{S^x_{t-1}\}^n_{i=1}$, are generated randomly and then, for each sample, $i = 1, ..., n$, we simulate the contribution and next post-decision state,$\{S^x_t\}_i$. We rewrite Equation (19) as

$$\underbrace{C(S_t, X^\pi(S_t|\theta))}_{C_{t,i}} = \underbrace{\left(\phi(S^x_{t-1}) - \gamma\mathbb{E}[\phi(S^x_t)|S^x_{t-1}]\right)^T}_{X_{t,i}}\theta + \underbrace{C(S_t, X^\pi(S_t|\theta)) - \mathbb{E}[C(S_t, X^\pi(S_t|\theta))|S^x_{t-1}]}_{C_{t,i}-\bar{C}_{t,i}}. \tag{20}$$

This is now in the form of a linear regression problem. Using simulation, we are able to get observations of $C(S_t, X^\pi(S_t|\theta))$ and $\left(\phi(S^x_{t-1}) - \gamma\mathbb{E}[\phi(S^x_t)|S^x_{t-1}]\right)^T$ in Equation (20). We can write this in matrix form as

$$\underbrace{C_t}_{n\times 1} = \underbrace{(\Phi_{t-1} - \gamma\Phi_t)}_{n\times k}\underbrace{\theta}_{k\times 1} + \underbrace{C_t - \bar{C}_t}_{n\times 1}, \tag{21}$$

where

$$C_t = \begin{bmatrix} C(\{S_t\}_1, \pi(\{S_t\}_1)) \\ \vdots \\ C(\{S_t\}_n, \pi(\{S_t\}_n)) \end{bmatrix}, \tag{22}$$

$$\Phi_{t-1} = \begin{bmatrix} \phi(\{S^x_{t-1}\}_1)^T \\ \vdots \\ \phi(\{S^x_{t-1}\}_n)^T \end{bmatrix}, \tag{23}$$

$$\Phi_t = \begin{bmatrix} \phi(\{S^x_t\}_1)^T \\ \vdots \\ \phi(\{S^x_t\}_n)^T \end{bmatrix}, \tag{24}$$

and

$$\bar{C}_t = \begin{bmatrix} \mathbb{E}[C(\{S_t\}_1, X^\pi(\{S_t\}_1|\theta))|\{S^x_{t-1}\}_1] \\ \vdots \\ \mathbb{E}[C(\{S_t\}_n, X^\pi(\{S_t\}_n|\theta))|\{S^x_{t-1}\}_n] \end{bmatrix}. \tag{25}$$

We have used subscripts $t-1$ and t to explicitly keep track of which vectors are known at time $t-1$ and t, respectively. We refer to $C_t - \bar{C}_t$ as the Bellman errors or Bellman residuals, although the terms may be defined slightly differently in other contexts.

For least-squares Bellman error minimization, the objective is to minimize the L_2 norm of the Bellman errors in Equation (21), $\frac{1}{n}(C_t - \bar{C}_t)^T(C_t - \bar{C}_t)$. Throughout this paper we use the following assumption which assumes the basis functions are linearly independent and certain matrices have full rank:

ASSUMPTION 1. *$\Phi_{t-1}, (\Phi_{t-1} - \gamma\Phi_t)$, and $(\Phi_{t-1})^T(\Phi_{t-1} - \gamma\Phi_t)$ have full column rank, and $k \leq n$.*

These assumptions can be interpreted as needing to visit enough different states such that the model can identified. The typical least-squares equation yields the following estimator for θ which we refer to as least-squares Bellman error minimization,

$$\hat{\theta} = \left[(\Phi_{t-1} - \gamma\Phi_t)^T(\Phi_{t-1} - \gamma\Phi_t)\right]^{-1}(\Phi_{t-1} - \gamma\Phi_t)^T C_t. \tag{26}$$

The matrix of regressors, $(\Phi_{t-1} - \gamma\Phi_t)$, is not deterministic ($\Phi_t$ is not deterministic because we cannot calculate $\mathbb{E}[\phi(S_t^x)|S_{t-1}^x]$); we can only simulate $\phi(S_t^x)$ given S_{t-1}^x, and, as a result, the least-squares estimator for θ will typically be inconsistent. Due to the structure of the problem, we use the method of instrumental variables instead (see Bradtke and Barto (1996) and Ma and Powell (2010)). An instrumental variable is a variable that is correlated with the regressors, but uncorrelated with the errors in the regressors and the observations (see Appendix A or Durbin (1954); Kendall and Stuart (1961); Söderström and Stoica (1983); Bowden and Turkington (1984)). This results in what we call instrumental variables Bellman error minimization (called LSTD in Bradtke and Barto (1996)),

$$\hat{\theta} = [(\Phi_{t-1})^T(\Phi_{t-1} - \gamma\Phi_t)]^{-1}(\Phi_{t-1})^T C_t. \tag{27}$$

Bradtke and Barto (1996) gives conditions such that Equation (27) is a consistent estimator ($\lim_{n\to\infty}\hat{\theta} = \theta$ with probability one) for the on-policy case. The proof references the consistency properties of the method of instrumental variables by showing that the columns of Φ^n are appropriate instrumental variables (see Appendix A). One interesting comment is that the matrix $[(\Phi_{t-1})^T(\Phi_{t-1} - \gamma\Phi_t)]$ could have negative eigenvalues, unlike $[(\Phi_{t-1} - \gamma\Phi_t)^T(\Phi_{t-1} - \gamma\Phi_t)]$.

Approximate Policy Iteration
(01) Initialize θ.
(02) **for** j = 1:M **(Policy Improvement Loop)**
(03) Define the policy $X^\pi(S_t\|\theta) = \text{argmax}_x[C(S_t,x) + \gamma\phi(S_t^x)^T\theta]$
(04) **for** i =1:N **(Policy Evaluation Loop)**
(05) Simulate a random post-decision state, S_{t-1}^x.
(06) Record $\phi(S_{t-1}^x)$.
(07) Simulate the state transition to get S_t.
(08) Determine the decision, $x = X^\pi(S_t\|\theta)$.
(09) Record $C_{t,i} = C(S_t,x)$.
(10) Record $\phi(S_t^x)$, the observation of $\mathbb{E}[\phi(S_t^x)\|S_{t-1}^x]$.
(11) **End**
(12) Update θ with Equation (26), (27), (30), or (31). **(Policy Evaluation)**
(13) **End**

Figure 4 Summary of approximate policy iteration. The inner loop simulates transitions from a fixed policy in order to approximately evaluate the fixed policy. The outer loop improves the policy.

4.4. Policy Evaluation using Projected Bellman Error Minimization

Again we start with the rearranged form of Bellman's equation using post-decision states in matrix form (see Equation (21)),

$$C_t = (\Phi_{t-1} - \gamma\Phi_t)\theta + (C_t - \bar{C}_{t-1}). \tag{28}$$

The idea of projected Bellman error minimization (also called least-squares fixed-point approximation in Lagoudakis and Parr (2003)) is to first project the Bellman errors into the space spanned by the basis functions of the value function and then minimize them (see Lagoudakis and Parr (2003) and Sutton et al. (2009)). Projecting the left and right hand sides of Equation (28) down into the space spanned by Φ_{t-1} (with respect to the L_2 norm), we get

$$\Pi_{t-1}C = \Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t)\theta + \Pi_{t-1}(C_t - \bar{C}_{t-1}), \tag{29}$$

where $\Pi_{t-1} = \Phi_{t-1}((\Phi_{t-1})^T\Phi_{t-1})^{-1}(\Phi_{t-1})^T$ is the projection operator onto the space spanned by the basis functions (see Tsitsiklis and Van Roy (May 1997) for the original derivation of this mapping or Powell (2011) [Section 8.2.3]). For completeness, we note that Π_{t-1} is the L_2 projection which solves the problem, $\min_\theta \|\Phi_{t-1}\theta - b\|_2 = \|\Pi_{t-1}b - b\|_2$. In other words, if you want to get from an arbitrary vector b, to the closest vector (in the L_2 sense) that is in the span of the columns of Φ_{t-1},

just apply the projection Π_{t-1} to the vector b. By Assumption 1, Φ_{t-1} has full column rank so Π_{t-1} is well defined.

Typically, Equation (29) is an over-determined set of equations. Taking a least squares approach, we find θ by minimizing the norm of the Projected Bellman Error

$$\min_{\theta} \|\Pi_{t-1}(C_t - \bar{C}_{t-1})\|_2 = \min_{\theta} \|\Pi_{t-1}C - \Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t)\theta\|_2.$$

The least-squares estimator of θ yields what we refer to as least-squares projected Bellman error minimization,

$$\hat{\theta} = \Big[(\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))^T (\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))\Big]^{-1} (\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))^T \Pi_{t-1}C_t. \tag{30}$$

However, this is the classic errors-in-variables model due to the randomness in our observations Φ_t, and instrumental variables can be used to construct a consistent estimator for θ (see Appendix A.1). We show Φ_{t-1} can be used as instrumental variables as before in Equation (27), and the proof is similar to that in Bradtke and Barto (1996). The resulting estimator is what we call the projected Bellman error minimization with instrumental variables,

$$\hat{\theta} = \big[(\Phi_{t-1})^T\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t)\big]^{-1} (\Phi_{t-1})^T\Pi_{t-1}C_t. \tag{31}$$

For completeness, we note that $\Pi_{t-1}\Phi_{t-1}$ could have been used for the instrumental variables instead of Φ_{t-1}, but linear algebra can be used to show the estimator would be equivalent to Equation (31).

4.5. Consistency of Projected Bellman Error Minimization with Instrumental Variables

We show that projected Bellman error minimization with instrumental variables is consistent (converges in probability to the true weights); the result holds even when the state space is continuous or the discount factor is one. For notation consistent with Appendix A, we let $X = \Pi_{t-1}(\Phi_{t-1} - \gamma\mathbb{E}[\gamma\Phi_t|\{S^x_{t-1}\}])$, $X' = \Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t)$, $X'' = X' - X$, $Y'' = \Pi_{t-1}(C_t - \bar{C}_{t-1})$, and $Z = \Phi_{t-1}$. Bradtke and Barto (1996) proves a similar result for the on-policy case.

We first assume that the covariance matrix between the instrumental variables and regressors has full rank, and we assume we restrict ourselves to the off-policy case:

ASSUMPTION 2. *Σ has full rank k, where $\Sigma_{jl} = Cov[Z_j, X_l]$.*

ASSUMPTION 3. *The rows of Φ_{t-1} are i.i.d. (off-policy).*

The following assumptions will be necessary to use the law of large numbers, which guarantees sample means converge to their true means:

ASSUMPTION 4. $\mathbb{E}[|Z_{ij}Y_i^{''}|] < \infty, \qquad \forall j = 1, ..., k.$

ASSUMPTION 5. $\mathbb{E}[|Z_{ij}X_{il}^{''}|] < \infty, \quad \forall j, l \in \{1, ..., k\}.$

COROLLARY 1. *Under Assumptions 1, 2, 3, 4, 5, $\hat{\theta} = ((\Phi_{t-1})^T \Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))^{-1} (\Phi_{t-1})^T \Pi_{t-1} C_t$ is a consistent estimator for θ defined in Equation* (29).

The proof follows from Proposition 1 in Appendix A. The following lemmas show that the assumptions in Appendix A for Proposition 1 hold. We first show Assumption 6 holds,

LEMMA 1. $\mathbb{E}[Y_i^{''}] = 0, \qquad \forall i.$

Proof: See Appendix B.1.

We next show Assumption 7 holds, which states that the mean of the noise in the observation of the explanatory variables is zero.

LEMMA 2. $\mathbb{E}[X_{ij}^{''} = 0], \qquad \forall i, j.$

Proof: See Appendix B.2.

We next show Assumption 8 holds, which states that the instrumental variables are uncorrelated with the noise in the observations of the response variable.

LEMMA 3. $Cov[Z_{ij}, Y_i^{''}] = 0, \qquad \forall i, j.$

Proof: See Appendix B.3.

We define e_i as a column vector of zeros with a one at the i'th position. We next show Assumption 9 holds,

LEMMA 4. $Cov[Z_{ij}, X_{il}^{''}] = 0, \qquad \forall i, j, l.$

Proof: See Appendix B.4.

Finally, Assumption 10 holds by Assumption 2, and Assumptions 11, 12, and 13 follow trivially from Assumptions 2, 3, 4, and 5 by the law of large numbers (see Appendix A). Therefore Proposition 1 applies. Q.E.D.

One interesting comment is that this proof holds even if the discount factor $\gamma = 1$. However, for Assumption 1 to hold when $\gamma = 1$, it is not hard to see that a constant basis function cannot be used because $(\Phi_{t-1} - \gamma\Phi_t)$ would not have full column rank.

4.6. Equivalence of Instrumental Variable Bellman Error Minimization and Projected Bellman Error Minimization

In Section 4.3 we summarized least-squares Bellman error minimization (Equation (26)) and instrumental variables Bellman error minimization (Equation (27)). In Section 4.4 we summarized least-squares projected Bellman error minimization (Equation (30)) and instrumental variables projected Bellman error minimization (Equation (31)). It turns out instrumental variables Bellman error minimization, least-squares projected Bellman error minimization, and instrumental variables projected Bellman error minimization are equivalent.

THEOREM 1. *Under Assumption 1, the following policy evaluation algorithms are equivalent:*

Instrumental Variables Bellman Error Minimization (LSTD)

$$\hat{\theta} = [(\Phi_{t-1})^T(\Phi_{t-1} - \gamma\Phi_t)]^{-1}(\Phi_{t-1})^T C_t, \tag{32}$$

Least-Squares Projected Bellman Error Minimization (Least-Squares Fixed Point Approx.)

$$\hat{\theta} = \left[(\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))^T (\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))\right]^{-1} (\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))^T \Pi_{t-1} C_t, \tag{33}$$

Instrumental Variables Projected Bellman Error Minimization

$$\hat{\theta} = \left[(\Phi_{t-1})^T \Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t)\right]^{-1} (\Phi_{t-1})^T \Pi_{t-1} C_t. \tag{34}$$

Proof: See Appendix C.

4.7. On-Policy Versus Off-Policy

For evaluating a fixed policy, Tsitsiklis and Van Roy (1997) proves that off-policy TD(λ) algorithms with a linear function approximation of the value function may not converge. In this case, off-policy means that the distribution of the states visited during a single infinite trajectory is not equal to the distribution of the states visited if we followed the fixed policy of interest. For a fixed policy, Tsitsiklis and Van Roy (1997) gives conditions under which on-policy temporal-difference learning will converge to the true value function projected onto the space of value function approximations (with respect to a weighted norm).

Bradtke and Barto (1996) gives conditions for on-policy policy evaluation based on Bellman error minimization to converge to a fixed value function when using a linear model for the value function. Lagoudakis and Parr (2003) explains that on-policy LSTD biases the value function and may do a very poor job of fitting the value function at states that are rarely visited. Another major disadvantage of on-policy is that, if the policy does not explore enough states, Assumption 1 may not hold. An important point to keep in mind is that the value of the greedy policy determined by the final value function approximation may be significantly different from the true value function.

5. Direct Policy Search

An alternative to Bellman error minimization for finding the regression vector θ is direct policy search. As before, we consider policies parameterized by θ of the form,

$$X^{\pi}(S_t|\theta) = \text{argmax}_x[C(S_t,x) + \gamma\phi(S_t^x)^T\theta],$$

where the post-decision value function $V(S^x)$ has been replaced by the linear model $\phi(S^x)^T\theta$. The goal of dynamic programming is to find a value function which satisfies Bellman's equation; the optimal post-decision value function easily translates into an optimal policy which maps a state to an action (this may not be true for pre-decision value functions). Unlike policy iteration or value iteration, the objective of direct policy search is not necessarily to find a value function that is close to the true value function (with respect to some norm); our objective is to find a value of θ for

which the policy $X^\pi(s|\theta)$ performs well. Additionally, we only need to consider features which are a function of the decisions; for this reason, the "value function approximation" is typically much simpler than what is required if we use Bellman error minimization. The challenge for direct policy search arises as the dimension of θ grows; randomly trying different values of θ is highly inefficient. However, direct policy search can use classic stochastic optimization algorithms to sequentially choose policies to simulate.

5.1. The Knowledge Gradient for Direct Policy Search

Our objective is to find a value of θ which solves the following stochastic optimization problem,

$$\max_\theta V^\pi(S_0), \tag{35}$$

given the policy $X^\pi(S_t|\theta)$. For a fixed value of θ we can obtain a noisy observation of the objective in Equation (35) by simulating $\hat{V}^\pi(S_0) = C_0(S_0, X^\pi(S_0|\theta)) + \gamma^1 C_1(S_1, X^\pi(S_1|\theta)) + \gamma^2 C_2(S_2, X^\pi(S_2|\theta)) + \cdots$. We can sequentially simulate the value for many different values of θ before determining which value of θ gives the best policy, $X^\pi(S_t|\theta)$. Unfortunately, the optimization problem given by Equation (35) is typically non-convex and non-separable. When the dimension of θ is small, the knowledge gradient for continuous parameters (KGCP) policy has been shown to work well for efficiently optimizing θ (see Scott et al. (2011a)).

The KGCP policy for optimizing θ combines a model of $\mu(\theta) = V^\pi(S)$ with a criterion which chooses the next value of θ for which a noisy observation of $\mu(\theta)$ will be simulated. In particular, the objective $\mu(\theta)$ is modeled using Gaussian process regression which can be viewed as a linear smoother. The KGCP quantifies how much we expect the maximum of the objective to increase by getting an additional noisy observation of $\mu(\theta)$ at a particular value of θ. More formally, we let $\mathcal{F}^n$ be the sigma-algebra generated by $\theta^0, ..., \theta^{n-1}$ and the corresponding noisy observations of $\mu(\theta^0), ..., \mu(\theta^{n-1})$. $\mu^n(\theta)$ is the updated Gaussian process regression function after n observations (see Scott et al. (2011b)). The KGCP is defined as

$$\bar{\nu}^{KG,n}(\theta) \triangleq \mathbb{E}\left[\max_{i=0,..,n} \mu^{n+1}(\theta^i) \middle| \mathcal{F}^n, \theta^n = \theta\right] - \max_{i=0,..,n} \mu^n(\theta^i)|_{\theta^n=\theta}.$$

In the Gaussian process regression framework, $\mu^{n+1}(\theta)$ given $\mathcal{F}^n$ is normally distributed for each value of θ, and the KGCP can be calculated exactly (see Scott et al. (2011b)). The KGCP can be viewed as a generalization of the expected improvement criterion from Jones et al. (1998) to the case with noisy observations (see Scott et al. (2011b)). The next sampling decision will be chosen to maximize the KGCP,

$$\theta^n \in \arg\max_{\theta} \bar{\nu}^{KG,n}(\theta).$$

After N observations, the implementation decision (the value of θ we believe is best) can be chosen by maximizing the regression function,

$$\theta^\star \in \arg\max_{\theta} \mu^N(\theta).$$

One additional challenge for using direct policy search is determining the feasible domain for θ; the domain of θ is typically restricted to a hypercube or simplex, because a true global search over all of $\mathcal{R}^k$ without any structure is typically an arbitrarily hard problem even with smoothness assumptions. The value of θ which maximizes Equation (35) produces the best policy within the class of polices, $X^\pi(S_t|\theta)$. Direct policy search has the potential to choose the best θ of any algorithm when choosing a policy $X^\pi(S_t|\theta)$, although in practice there is always a limited budget (primarily due to time) of how many policies we can simulate.

6. Numerical Experiments

Our main objective is to compare approximate policy iteration (API) with least-squares Bellman error minimization to API with instrumental variables Bellman error minimization to see if instrumental variables add value in practice. We first compare the algorithms on discretized benchmark problems with known solutions so we can report how well they perform relative to optimal. Additionally we run direct policy search on the discretized benchmark problems to see if we can find an even better policy. Finally, we run approximate policy iteration on a problem with a state consisting of five continuous dimensions and actions consisting of five continuous dimensions.

6.1. Creating Benchmark Problems

We first consider a finite, discretized state and action space with a fixed probability transition matrix. One solution technique for finding the exact solution is value iteration (see Puterman (1994)). $V^0(S)$ is initialized to a constant for all S, and at each iteration, n, the algorithm updates the values of each state,

$$V^n(s) = \max_x\{C(s,x) + \gamma \sum_{s'} V^{n-1}(s')\mathbb{P}(s'|s,x)\}, \quad \forall s \in \mathcal{S}. \tag{36}$$

The algorithm will converge to the true value function of the optimal policy which satisfies Bellman's equation,

$$V(S) = \max_x\{C(S,x) + \gamma \mathbb{E}\left[V(S'(S,x))|S\right]\}. \tag{37}$$

We discretized the state space in the benchmark test problems and then created fixed probability transition matrices for the exogenous information process in order to create a true discrete process (this is different than simply simulating a continuous process and then discretizing as you progress).

In Table 1 we summarize a list of the benchmark problems described in Section 3 with exact solutions. "Full" refers to the problem shown in Figure 1 with energy from wind and the grid serving a load. "BA" refers to a battery arbitrage problem without wind or a load, where you buy and sell electricity from the grid using storage. We include how finely each state variable is discretized (the size of the state space for a particular problem is the product of each of the discretization levels). We then list the wind capacity divided by the load, the storage capacity divided by the load over an hour, the round trip efficiency (RTE) of the storage device, and the max charge and discharge rate of the storage device. For example, C/10 means the storage device can be fully charged or discharged in 10 hours. The transition matrix of the electricity prices was fit using the PJM Western Hub real time prices (with and without time of day). The transition matrix of the load was fit using the load of the PJM Mid-Atlantic Region (with time of day). The transition matrix for the wind was fit using data from wind speeds near the Sweetwater Wind Farm. For Problems $1-16$ the state space is resource level, wind energy, and electricity price, $S_t = (R_t, E_t, P_t)$ (time and

demand are fixed). In these problems, the load is held constant in order to keep the benchmark problems computationally tractable (exact value iteration, even for this simplified problem, requires approximately 2 weeks on a 3Ghz processor). Later, we demonstrate the approximate algorithms on problems with stochastic, time-dependent loads. For Problems $17-20$, the state space is the time-of-day, τ_t, (1-96 corresponding to fifteen minute intervals in a day), the resource level, and the electricity price, giving us the state variable $S_t = (\tau_t, R_t, P_t)$. Δt is fixed to fifteen minutes for all the problems. We use a discount factor, $\gamma = .999$. We found that discount factors of $\gamma = .99$ or smaller produce policies that are relatively myopic, and do not allow us to hold energy in storage for extended periods.

		# of Discretization Levels								
Prob	Type	Time	Resource	Price	Load	Wind	Wind	Storage	RTE	Charge Rate
1	Full	1	33	20	1	10	0.1	2.5	.81	C/10
2	Full	1	33	20	1	10	0.1	2.5	.81	C/1
3	Full	1	33	20	1	10	0.1	2.5	.70	C/10
4	Full	1	33	20	1	10	0.1	2.5	.70	C/1
5	Full	1	33	20	1	10	0.2	2.5	.81	C/10
6	Full	1	33	20	1	10	0.2	2.5	.81	C/1
7	Full	1	33	20	1	10	0.2	2.5	.70	C/10
8	Full	1	33	20	1	10	0.2	2.5	.70	C/1
9	Full	1	33	20	1	10	0.1	5.0	.81	C/10
10	Full	1	33	20	1	10	0.1	5.0	.81	C/1
11	Full	1	33	20	1	10	0.1	5.0	.70	C/10
12	Full	1	33	20	1	10	0.1	5.0	.70	C/1
13	Full	1	33	20	1	10	0.2	5.0	.81	C/10
14	Full	1	33	20	1	10	0.2	5.0	.81	C/1
15	Full	1	33	20	1	10	0.2	5.0	.70	C/10
16	Full	1	33	20	1	1	0.2	5.0	.70	C/1
17	BA	96	33	20	1	1	-	-	.81	C/10
18	BA	96	33	20	1	1	-	-	.81	C/1
19	BA	96	33	20	1	1	-	-	.70	C/10
20	BA	96	33	20	1	1	-	-	.70	C/1

Table 1 Set of benchmark problems specifying the type (Full or Battery Arbitrage), the number of discretization levels for time (1=steady state), resource, price, load (1=deterministic) and wind. The remaining columns specify average maximum wind divided by the load, storage capacity divided by hourly load, round trip efficiency (RTE), and the maximum charge/discharge rate (C/10 means it takes hours to charge/discharge).

6.2. Comparing to the Benchmark

In order to choose how long to run the inner policy evaluation loop and outer policy improvement loop (see Figure 4), we ran approximate policy iteration using instrumental variables Bellman error minimization several times on one of the problems. For the test problems, we found most of the improvement has occurred before $M = 30$ policy improvements and policy evaluations of length $N = 5000$.

In Figure 5 we compare approximate policy iteration with instrumental variables Bellman error minimization, approximate policy iteration with least-squares Bellman error minimization, and direct policy search based on KGCP (described in Section 5) to see if the method of instrumental variables adds value as the theory suggests. In addition, we show the performance of the myopic policy which discharges the battery as quickly as possible and then leaves it empty. The value of the myopic policy is still positive due to the wind power as well as the initial energy in the battery. In Figure 5, approximate policy iteration with instrumental variables Bellman error minimization and least-squares Bellman error minimization use quadratic basis functions, and we run each algorithm 100 times. For each run of the algorithms, the final policies produced by each algorithm are then evaluated on the same sample path, $\omega \in \Omega$, where ω is generated from the discretized exogenous information process. We then record the average percent of optimal and the standard deviation of the average percent of optimal across the 100 runs. The average percentage of optimal for a policy π is computed as

$$\% \text{ of optimal} = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \frac{\hat{F}^{\pi}(\omega)}{V^{\star}(S_0(\omega))},$$

where ω is a sample path of the randomness in the state transitions, and $S_0(\omega)$ is the starting state which has been randomly generated from a uniform distribution. $\hat{F}^{\pi}(\omega)$ is a realization of value of the policy π run on the sample path ω, starting at the state $S_0(\omega)$, and $V^{\star}(S_0(\omega))$ is the true value of the optimal policy for state $S_0(\omega)$ which is computed using Equation (36). We ran the approximate policy iteration with other sets of basis functions (first-order, third-order, fourth-order), but quadratic basis functions performed the best (see Appendix D).

When we perform direct policy search using KGCP, we budget ourselves to simulating 50 sequentially chosen policies, after which the KGCP algorithm must choose what it believes to be the best policy. This is repeated 100 times and the average percent of optimal and standard deviation of the average percent of optimal are given in Figure 5. Direct policy search produced solutions that were on average 91.8 percent of optimal, and were always at least 70 percent of optimal, for problems 1 through 16. One interesting observation is that direct policy search performed at least 70% of optimal for problems 1 through 16, which suggests direct policy search is robust. In particular, direct policy search did much better on many of the problems and should only improve if the algorithm is allowed to run longer (although the algorithm becomes very time consuming). However, direct policy search quickly becomes intractable as the number of basis functions increases. Choosing the search domain for direct policy search is another significant complication as the number of basis functions increases. We suggest using approximate policy iteration to find good values of the regression parameters, and then use direct policy search to improve the policy in the region of the fitted regression parameters.

In Figure 6 we show a sample path of a policy produced by approximate policy iteration on Problem 1 in Table 1. We see that the resource is charged when electricity prices are low and discharged when electricity prices are high. We also note that the battery fully discharges (down to 20 percent) relatively infrequently.

One way to reduce the number of basis functions used by the algorithms is to ignore dimensions of the post-decision state when constructing the value function approximation. In Figure 7, we show the results using three value function approximations: 1) resource level, wind power and electricity price, 2) resource level only, and 3) resource level and electricity price. We observe that using the resource level alone for the domain of the post-decision value function appears to do quite poorly for most problems. Using both resource level and electricity price appears to do fairly well overall, although using all the dimensions of the state variable appears to do the best. For certain problems, it may actually be advantageous to leave variables out of the state space in order to have a smaller number of weights to estimate for the value function.

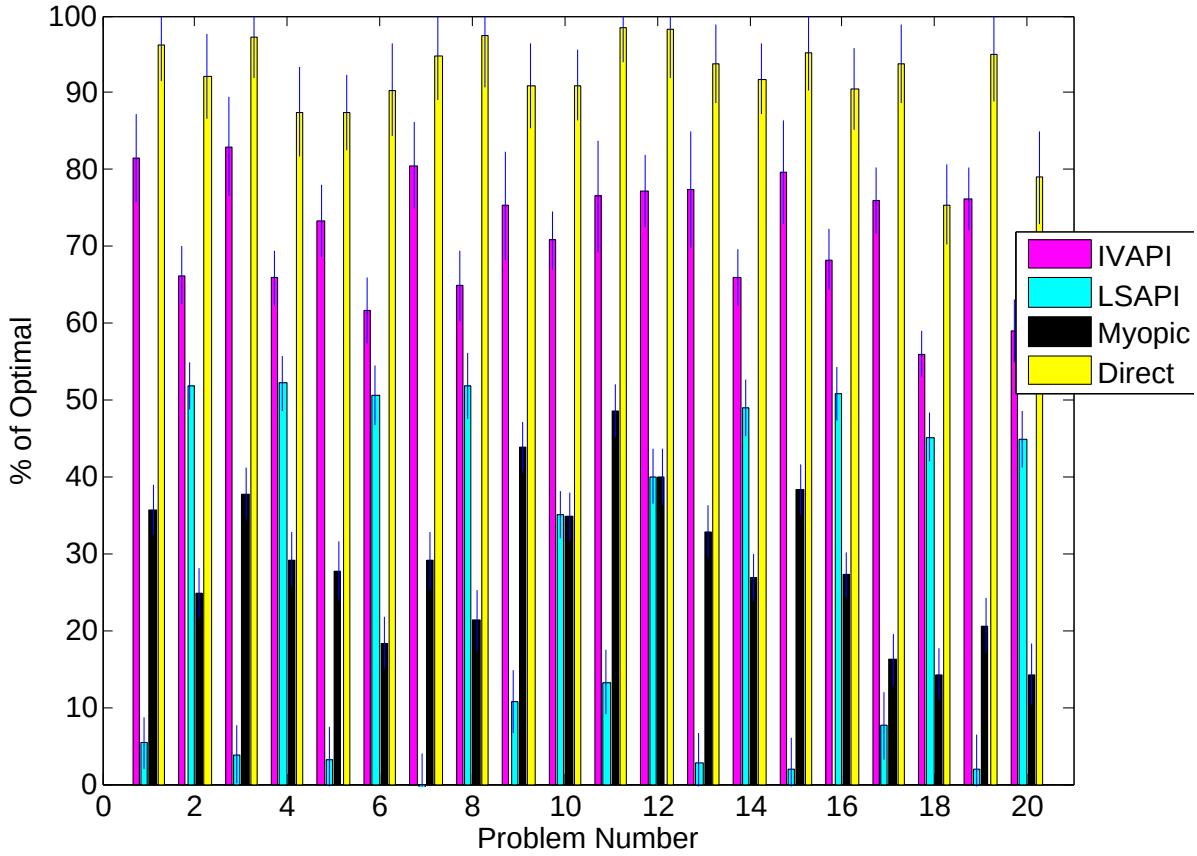
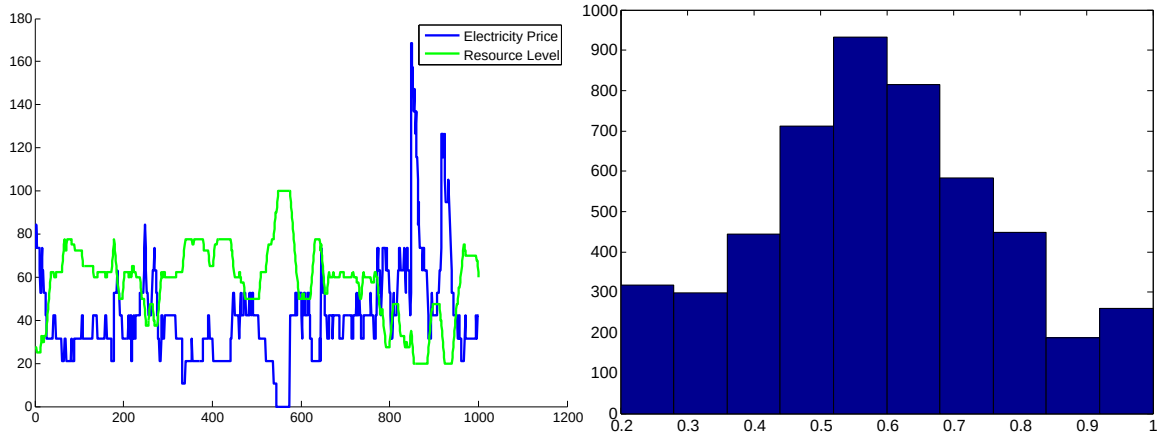


Figure 5 Performance as a percent of the benchmark optimization solution using API with instrumental variables, least-squares API, a myopic policy and direct policy search.



(a) (b)

Figure 6 We plot a 10 day sample path of a policy produced by approximate policy iteration with instrumental variables Bellman error minimization using quadratic basis functions on Problem 1. (a) We plot the electricity price and resource level. (b) We plot a histogram of the resource level.

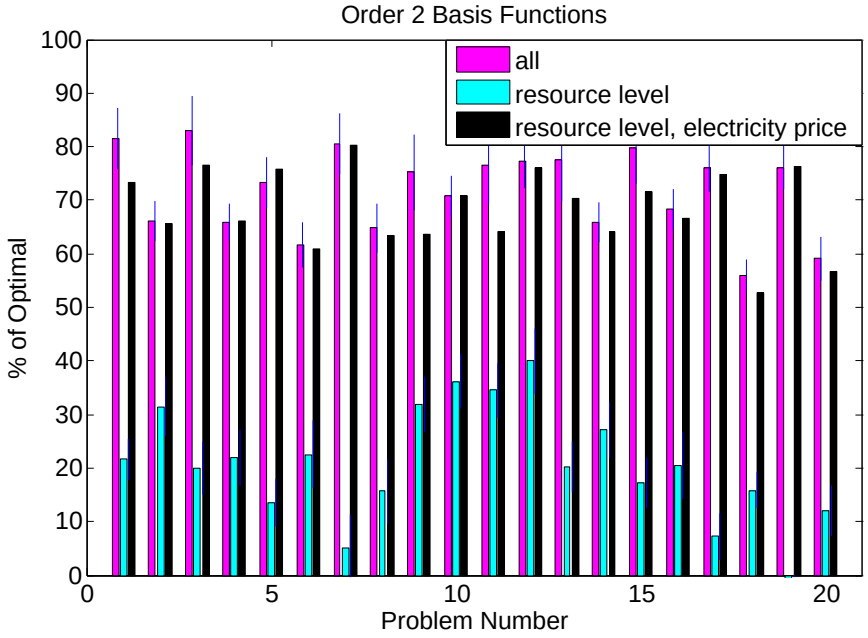


Figure 7 The algorithms use quadratic basis functions. We show the percentage of optimal along with 95% confidence intervals for the average percentage of optimal for Bellman error minimization using instrumental variables (IV) when only certain dimensions of the post-state are included in the post-state value function approximation.

6.3. A Continuous Problem

In this section we consider problems with continuous states as well as a larger state space. We compare both approximate policy iteration algorithms on the continuous problems described in Table 2, although an optimal solution will no longer be available. These problems now have continuous states and continuous actions and the state transitions correspond the models in Section 2. The electricity prices and loads are now time-dependent and stochastic for Problems 1-3. Problems 4-10 are continuous steady-state problems.

Figure 8 shows that least-squares Bellman error minimization performs very poorly and the instrumental variables do indeed add value. Although all the dimensions of the state variable and action space are difficult to visualize, in Figure 9 we use a policy produced by approximate policy

		# of Discretization Levels								
Prob	Type	Time	Resource	Price	Load	Wind	Wind	Storage	RTE	Charge Rate
1	Full	96	Cont.	Cont.	Cont.	Cont.	0.1	2.5	.81	C/10
2	Full	96	Cont.	Cont.	Cont.	Cont.	0.1	5.0	.81	C/10
3	BA	96	Cont.	Cont.	1	1	-	-	.81	C/10
4	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	5.0	.81	C/10
5	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	2.5	.81	C/1
6	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	2.5	.70	C/1
7	BA	1	Cont.	Cont.	1	1	-	-	.81	C/10
8	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	5.0	.81	C/1
9	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	5.0	.70	C/1
10	Full	1	Cont.	Cont.	Cont.	Cont.	0.2	2.5	.81	C/1

Table 2 Parameter settings for problems with continuous states. Problems 1-3 have time-dependent stochastic loads and electricity prices. Problems 4-10 are steady-state.

iteration with instrumental variables Bellman error minimization to show the electricity price and the percent of the storage which is full on one particular sample path. We can see that the policy tends to start charging the battery at night when electricity prices are low and then discharges the battery throughout the day when electricity prices are higher. Approximate policy iteration with instrumental variables Bellman error minimization is a promising scalable algorithm which is designed for problems where the states are continuous and the transition probabilities are unknown.

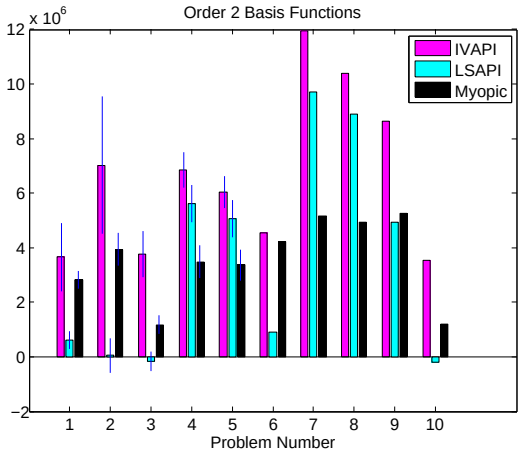


Figure 8 We plot the average objective of both approximate policy iteration algorithms on the continuous problems shown in Table 2.

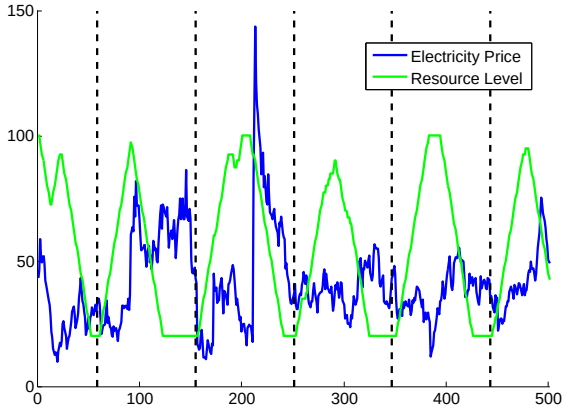


Figure 9 A sample path of the electricity spot price and resource level. The vertical lines correspond to midnight.

7. Conclusion

We have created a series of difficult benchmark problems that arise in a class of energy storage problems that represent a difficult algorithmic challenge with respect to identifying good control policies. The idea is to create (discretized) problems that can be solved optimally, use these benchmark problems to evaluate scalable approximation algorithms which can then be used on more complex problems.

We quickly found that considerable care has to be used in creating benchmark problems. For example, we found that using a discount factor of .99 produced problems where myopic policies worked well. As a result, substantial errors in the value function approximation still produced results that were within a few percent of optimal. The same result occurred if the battery was small relative to the amount of available wind energy. Our problems were chosen both to model realistic systems, but also to provide an algorithmic challenge, as evidenced by the poor performance of a myopic policy.

We compared three strategies based on Bellman error minimization (classical least squares approximate policy iteration, and variants that use instrumental variables and projected Bellman error minimization), to one based on direct policy search. This work produced several surprising results. First, we found that the performance using instrumental variables and projected Bellman error were not just similar - they were the same, an observation that led to a mathematical proof of

this result. Second, we were somewhat surprised and impressed at how much better Bellman error minimization performed using instrumental variables, a technique that does not seem to be widely used in the reinforcement learning literature and virtually unknown in other ADP communities. But third, we were also surprised and a bit disappointed at how poorly Bellman error minimization, even with instrumental variables, worked relative to both the optimal solution as well as the performance of direct policy search.

This research suggests that direct policy search should be used, perhaps in conjunction with approximate policy iteration. The challenge is that in its derivative-free form, it does not scale easily to large numbers of parameters. This may be a major limitation in time-dependent applications where we may need to estimate a different set of parameters for each time period.

Appendix A: The Instrumental Variable Method

The instrumental variable method is a well known technique for dealing with errors in the explanatory variables (errors-in-variables) of a regression problem. In this section we summarize explanatory variables and the consistency properties mentioned in Durbin (1954), Bradtke and Barto (1996), Young (1984), and Kendall and Stuart (1961). We consider the linear model in matrix form:

$$Y_i = \sum_{j=1}^{k} X_{ij}\beta_j \qquad i = 1, ..., n, \tag{38}$$

or

$$Y = X\beta,$$

where Y is a $n \times 1$ vector of the response variable, X is a $n \times k$ matrix of explanatory variables, and β is a $k \times 1$ vector of weights. Suppose we can only observe X' and Y', not the true values X and Y. Following similar notation as Durbin (1954),

$$X'_{ij} = X_{ij} + X''_{ij},$$

$$Y'_i = Y_i + Y''_i,$$

and in matrix form,

$$X' = X + X'', \tag{39}$$

$$Y' = Y + Y'', \tag{40}$$

where X'' and Y'' are the errors in the observed values of X and Y. Our linear regression model can now be written as:

$$Y' = X\beta + Y''.$$

As explained in Kendall and Stuart (1961), we treat X and Y as random variables. Unlike a standard linear regression problem, Equation (38) is a structural relation which relates two random variables (X is not deterministic). This is different than a regression line which gives a functional relationship that relates the mean of the dependant variable to the regressor variable (see Kendall and Stuart (1961)).

The first assumptions are that the noise in X and Y have mean zero,

ASSUMPTION 6. $\mathbb{E}[Y_i^{''}] = 0, \qquad i = 1, ..., n.$

ASSUMPTION 7. $\mathbb{E}[X_{ij}^{''}] = 0, \qquad \forall i, j.$

A.1. Example of Bias in Ordinary Least Squares

Kendall and Stuart (1961) and Durbin (1954) show that least squares regression encounters problems with the model given by Equations (38), (39), (40). The source of the problem is the correlation between the X' and X'', since the observation of X is typically correlated with the error in X. If β is a scalar ($k = 1$), this is easy to show. Starting with the least squares estimate of β_1,

$$\begin{aligned}
\hat{\beta_1} &= ((X^{'})^T X^{'})^{-1} (X^{'})^T Y^{'} \\
\hat{\beta_1} &= \frac{\sum_{i=1}^n X_{i1}^{'} Y_i^{'}}{\sum_{i=1}^n (X_{i1}^{'})^2} \\
&= \frac{\sum_{i=1}^n X_{i1}^{'} (X_{i1}\beta_1 + Y_i^{''})}{\sum_{i=1}^n (X_{i1}^{'})^2} \qquad (41) \\
&= \frac{\sum_{i=1}^n X_{i1}^{'} \left((X_{i1}^{'} - X_{i1}^{''})\beta_1 + Y_i^{''} \right)}{\sum_{i=1}^n (X_{i1}^{'})^2} \qquad (42)
\end{aligned}$$

$$= \beta_1 - \beta_1 \frac{\sum_{i=1}^n X'_{i1} X''_{i1}}{\sum_{i=1}^n (X'_{i1})^2} + \frac{\sum_{i=1}^n X'_{i1} Y''_i}{\sum_{i=1}^n (X'_{i1})^2} \tag{43}$$

In Equation (41) we substituted in Equation (38) and (40). In Equation (42) we used Equation (39). Now taking the limit as n goes to infinity, Equation (43) becomes

$$\begin{aligned}
\lim_{n\to\infty} \hat{\beta}_1 &= \beta_1 - \beta_1 \lim_{n\to\infty} \left(\frac{\sum_{i=1}^n X'_{i1} X''_{i1}}{\sum_{i=1}^n (X'_{i1})^2} \right) + \lim_{n\to\infty} \left(\frac{\sum_{i=1}^n X'_{i1} Y''_i}{\sum_{i=1}^n (X'_{i1})^2} \right) \\
&= \beta_1 - \beta_1 \lim_{n\to\infty} \left(\frac{\sum_{i=1}^n X'_{i1} X''_{i1}}{\sum_{i=1}^n (X'_{i1})^2} \right) + \frac{\mathrm{Cov}[X'_{i1}, Y''_i]}{\mathbb{E}[(X'_{i1})^2]} \qquad (44) \\
&= \beta_1 - \beta_1 \lim_{n\to\infty} \left(\frac{\sum_{i=1}^n X'_{i1} X''_{i1}}{\sum_{i=1}^n (X'_{i1})^2} \right). \qquad (45)
\end{aligned}$$

Equation (45) holds if $\mathrm{Cov}[X'_{i1}, Y''_i] = 0$. For many problems, X'_{i1} and X''_{i1} are positively correlated. Hence Equation (45) shows that typically the least squares estimate of β_1 is inconsistent and too small in magnitude for these problems. This problem can be overcome if an instrumental variable is available.

In Figure 10, we generated the regressors X and Z from a multivariate normal distribution with correlation 0.7. We then added independent Gaussian noise to X, and Y, where $Y = X\beta$. The the various regression techniques are plotted Figure 10. The errors in the X have violated the assumptions necessary for least-squares, and the least-squares regression line is too flat, as the theory predicts. The least-absolute-deviations regression (L1) is also too flat in this example. The instrumental variables method is consistent for this problem and this can be observed in the figure.

If an instrumental variable is know to exist, why not just add it as an additional regressor? If our main goal is to determine the value of β, adding dimensions to the regression problem could cause β to lose its meaning. As we see below, a properly chosen instrumental variable can yield a consistent estimator for β.

A.2. Consistency of Estimate using Instrumental Variables

An instrumental variable, Z_j, should be correlated with the true X_j but uncorrelated with the errors in the observations of X and Y (see Durbin (1954) and Kendall and Stuart (1961)). We use the notation X_j to indicate the j'th column of X. Let Σ be the $k \times k$ matrix where $\Sigma_{jl} = \mathrm{Cov}[Z_j, X_l]$.

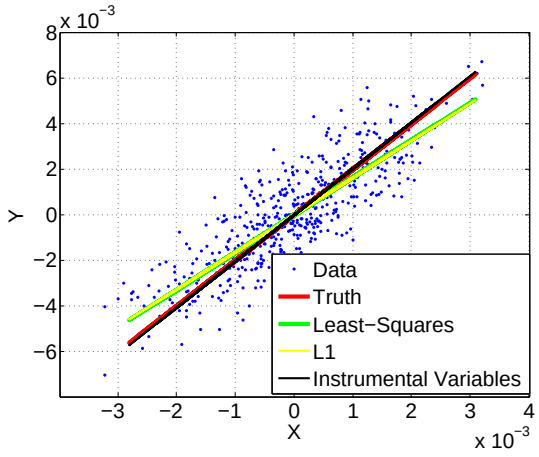


Figure 10 When there are errors in the regressors, instrumental variables can be used to solve the problem.

Below we extend the consistency proof from Kendall and Stuart (1961) to use multiple instrumental variables ($k > 1$). We assume an instrumental variable exists with the following properties:

ASSUMPTION 8. $Cov[Z_{ij}, Y_i''] = 0, \qquad \forall i, j.$

ASSUMPTION 9. $Cov[Z_{ij}, X_{il}''] = 0, \qquad \forall i \in \{1, ..., n\}, \quad j, l \in \{1, ..., k\}.$

ASSUMPTION 10. Σ *has full rank* k*, where* $\Sigma_{jl} = Cov[Z_j, X_l]$.

ASSUMPTION 11. $\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} Z_{ij} Y_i'' = 0, \qquad \forall j = 1, ..., k.$

ASSUMPTION 12. $\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} Z_{ij} X_{il}'' = 0, \quad \forall j, l \in \{1, ..., k\}.$

ASSUMPTION 13. $\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} Z_{ij} X_{il} = Cov[Z_j, X_l], \quad \forall j, l \in \{1, ..., k\}.$

Assumptions 11, 12, and 13 do not follow trivially from assumptions 8, 9, 10 without additional assumptions such as independence across the n observations (because the law of large numbers does not apply). The method of instrumental variables defines the estimator $\hat{\beta}$ as the solution to

$$Z^T X' \hat{\beta} = Z^T Y', \tag{46}$$

where Z is a $n \times k$ matrix. Note that $\hat{\beta}$ is uniquely defined when $Z^T X'$ has full rank k.

PROPOSITION 1. *For the model given by Equations* (38), (39), (40)*, if Assumptions 6, 7, 8, 9, 10, 11, 12, 13 hold, then* $\hat{\beta} = [Z^T X']^{-1} Z^T Y'$ *is a consistent estimator of* β.

Simplifying Equation (46) we get

$$Z^T(X+X^{''})\hat{\beta} = Z^T(X\beta+Y''), \tag{47}$$
$$(Z^TX+Z^TX^{''})\hat{\beta} = Z^TX\beta+Z^TY''.$$

In Equation (47), we used Equations (38), (39), and (40). Now, taking the limit as n goes to infinity,

$$\lim_{n\to\infty}\frac{1}{n}\left(Z^TX+Z^TX^{''}\right)\hat{\beta} = \lim_{n\to\infty}\frac{1}{n}\left(Z^TX\beta+Z^TY''\right) \tag{48}$$
$$\lim_{n\to\infty}\frac{1}{n}Z^TX\hat{\beta} = \lim_{n\to\infty}\frac{1}{n}Z^TX\beta \tag{49}$$
$$\Sigma\left(\lim_{n\to\infty}\hat{\beta}\right) = \Sigma\beta \tag{50}$$
$$\lim_{n\to\infty}\hat{\beta} = \beta. \tag{51}$$

In Equation (49), we used Assumptions 8 and 11 which imply $\lim_{n\to\infty}\frac{1}{n}Z^TY^{''}=\vec{0}$ and Assumptions 9 and 12 which imply $\lim_{n\to\infty}\frac{1}{n}Z^TX^{''}=\mathbf{0}$. We also used Slutsky's theorem when taking the limit. In Equation (50) the sample covariances converge in probability to the true covariances by Assumption 13. Getting to Equation (51), we used Assumption 10 which ensures that $\hat{\beta}$ is unique. Q.E.D.

Appendix B: Proof of Lemmas in Section 4.5

B.1. Proof of Lemma 1

$$\mathbb{E}[Y^{''}] = \mathbb{E}[\Pi_{t-1}(C_t-\bar{C}_{t-1})] \tag{52}$$
$$= \mathbb{E}\left[\mathbb{E}[\Pi_{t-1}(C_t-\bar{C}_{t-1})|\{S^x_{t-1}\}]\right] \tag{53}$$
$$= \mathbb{E}[\Pi_{t-1}\underbrace{\mathbb{E}[(C_t-\bar{C}_{t-1})|\{S^x_{t-1}\}]}_{=\vec{0}}] \tag{54}$$
$$= \vec{0}. \tag{55}$$

B.2. Proof of Lemma 2

$$\mathbb{E}[X^{''}] = \mathbb{E}[X^{'}-X] \tag{56}$$

$$= \mathbb{E}\left[\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t) - \Pi_{t-1}(\Phi_{t-1} - \mathbb{E}[\gamma\Phi_t|\{S^x_{t-1}\}])\right] \tag{57}$$

$$= \gamma\mathbb{E}\left[\Pi_{t-1}(\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \Phi_t)\right] \tag{58}$$

$$= \gamma\mathbb{E}\left[\mathbb{E}[\Pi_{t-1}(\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \Phi_t)|\{S^x_{t-1}\}]\right] \tag{59}$$

$$= \gamma\mathbb{E}\left[\Pi_{t-1}\mathbb{E}[\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \Phi_t|\{S^x_{t-1}\}]\right] \tag{60}$$

$$= \gamma\mathbb{E}[\Pi_{t-1}(\underbrace{\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \mathbb{E}[\Phi_t|\{S^x_{t-1}\}]}_{=\mathbf{0}})]. \tag{61}$$

$$\tag{62}$$

B.3. Proof of Lemma 3

$$\mathrm{Cov}[Z_{ij}, Y_i^{''}] = \mathbb{E}[Z_{ij}Y_i^{''}] - \mathbb{E}[Z_{ij}]\underbrace{\mathbb{E}[Y_i^{''}]}_{=0} \tag{63}$$

$$= \mathbb{E}[\Phi_{t-1,ij}\{\Pi_{t-1}(C_t - \bar{C}_{t-1})\}_i] \tag{64}$$

$$= \mathbb{E}[\Phi_{t-1,ij}e_i^T\Pi_{t-1}(C_t - \bar{C}_{t-1})] \tag{65}$$

$$= \mathbb{E}\left[\mathbb{E}[\Phi_{t-1,ij}e_i^T\Pi_{t-1}(C_t - \bar{C}_{t-1})|\{S^x_{t-1}\}]\right] \tag{66}$$

$$= \mathbb{E}[\Phi_{t-1,ij}e_i^T\Pi_{t-1}\underbrace{\mathbb{E}[C_t - \bar{C}_{t-1}|\{S^x_{t-1}\}]}_{=\vec{0}}] \tag{67}$$

$$= 0. \tag{68}$$

B.4. Proof of Lemma 4

$$\mathrm{Cov}[Z_{ij}, X_{il}^{''}] = \mathbb{E}[Z_{ij}X_{il}^{''}] - \mathbb{E}[Z_{ij}]\underbrace{\mathbb{E}[X_{il}^{''}]}_{=0} \tag{69}$$

$$= \mathbb{E}[Z_{ij}(X_{il}^{'} - X_{il})] \tag{70}$$

$$= \mathbb{E}[Z_{ij}e_i^T(X^{'} - X)e_l] \tag{71}$$

$$= \gamma\mathbb{E}[\Phi_{t-1,ij}e_i^T\Pi_{t-1}(\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \Phi_t)e_l] \tag{72}$$

$$= \gamma\mathbb{E}[\mathbb{E}[\Phi_{t-1,ij}e_i^T\Pi_{t-1}(\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \Phi_t)e_l|\{S^x_{t-1}\}]] \tag{73}$$

$$= \gamma\mathbb{E}[\Phi_{t-1,ij}e_i^T\Pi_{t-1}\mathbb{E}[\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \Phi_t|\{S^x_{t-1}\}]e_l] \tag{74}$$

$$= \gamma\mathbb{E}[\Phi_{t-1,ij}e_i^T\Pi_{t-1}\underbrace{(\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \mathbb{E}[\Phi_t|\{S^x_{t-1}\}])}_{=0}e_l] \tag{75}$$

$$= 0. \tag{76}$$

Appendix C: Proof of Theorem 1

We first show Equations (32) and (34) are equivalent. Starting with Equation (34) and recalling that, by definition, $\Pi_{t-1} = \Phi_{t-1}((\Phi_{t-1})^T\Phi_{t-1})^{-1}(\Phi_{t-1})^T$, we can write

$$\begin{aligned}
&\left[(\Phi_{t-1})^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)\right]^{-1}(\Phi_{t-1})^T\Pi_{t-1}C_t \\
&= \left[(\Phi_{t-1})^T(\underbrace{\Pi_{t-1}\Phi_{t-1}}_{\Phi_{t-1}}-\gamma\Pi_{t-1}\Phi_t)\right]^{-1}\underbrace{(\Phi_{t-1})^T\Phi_{t-1}((\Phi_{t-1})^T\Phi_{t-1})^{-1}}_{I_{k\times k}}(\Phi_{t-1})^TC_t \\
&= \left[(\Phi_{t-1})^T\Phi_{t-1}-\gamma(\Phi_{t-1})^T\Pi_{t-1}\Phi_t\right]^{-1}(\Phi_{t-1})^TC_t \\
&= \left[(\Phi_{t-1})^T\Phi_{t-1}-\gamma\underbrace{(\Phi_{t-1})^T\Phi_{t-1}((\Phi_{t-1})^T\Phi_{t-1})^{-1}}_{I_{k\times k}}(\Phi_{t-1})^T\Phi_t\right]^{-1}(\Phi_{t-1})^TC_t \\
&= \left[(\Phi_{t-1})^T\Phi_{t-1}-\gamma(\Phi_{t-1})^T\Phi_t\right]^{-1}(\Phi_{t-1})^TC_t \\
&= \left[(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)\right]^{-1}(\Phi_{t-1})^TC_t.
\end{aligned}$$

Hence Equations (32) and (34) are equivalent. Next we show Equations (32) and (33) are equivalent. We start by writing

$$\begin{aligned}
&(\Phi_{t-1}-\gamma\Phi_t)^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t) \\
&= (\Phi_{t-1}-\gamma\Phi_t)^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t) \\
\implies\quad &(\Phi_{t-1}-\gamma\Phi_t)^T\Phi_{t-1}[(\Phi_{t-1})^T\Phi_{t-1}]^{-1}(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t) \\
&= (\Phi_{t-1}-\gamma\Phi_t)^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t) \\
\implies\quad &(\Phi_{t-1}-\gamma\Phi_t)^T\Phi_{t-1}[(\Phi_{t-1})^T\Phi_{t-1}]^{-1}\underbrace{(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)[(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}}_{I_k}(\Phi_{t-1})^T \\
&= (\Phi_{t-1}-\gamma\Phi_t)^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)[(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1})^T \\
\implies\quad &(\Phi_{t-1}-\gamma\Phi_t)^T\underbrace{\Phi_{t-1}[(\Phi_{t-1})^T\Phi_{t-1}]^{-1}(\Phi_{t-1})^T}_{\Pi_{t-1}} \\
&= (\Phi_{t-1}-\gamma\Phi_t)^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)[(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1})^T \\
\implies\quad &(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))^T \\
&= (\Phi_{t-1}-\gamma\Phi_t)^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)[(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1})^T \qquad (77) \\
\implies\quad &[(\Phi_{t-1}-\gamma\Phi_t)^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))^T
\end{aligned}$$

$$= \underbrace{[(\Phi_{t-1}-\gamma\Phi_t)^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1}-\gamma\Phi_t)^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)}_{I_k}[(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1})^T$$

$$\implies \quad [(\Phi_{t-1}-\gamma\Phi_t)^T \underbrace{\Pi_{t-1}}_{(\Pi_{t-1})^T\Pi_{t-1}} (\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))^T$$

$$= [(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1})^T$$

$$\implies \quad [(\Phi_{t-1}-\gamma\Phi_t)^T(\Pi_{t-1})^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1}-\gamma\Phi_t)^T \underbrace{(\Pi_{t-1})^T}_{(\Pi_{t-1})^T\Pi_{t-1}}$$

$$= [(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1})^T$$

$$\implies \quad [(\Phi_{t-1}-\gamma\Phi_t)^T(\Pi_{t-1})^T\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1}-\gamma\Phi_t)^T(\Pi_{t-1})^T\Pi_{t-1}$$

$$= [(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1})^T$$

$$\implies \quad \left[(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))^T(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))\right]^{-1}(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))^T\Pi_{t-1}$$

$$= [(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1})^T$$

$$\implies \quad \left[(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))^T(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))\right]^{-1}(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))^T\Pi_{t-1}C_t$$

$$= [(\Phi_{t-1})^T(\Phi_{t-1}-\gamma\Phi_t)]^{-1}(\Phi_{t-1})^TC_t.$$

Equation (77) uses the fact that $(\Pi_{t-1})^T = \Pi_{t-1}$. Hence Equations (32), (33), and (34) are equivalent.

Appendix D: Performance of Algorithms with Different Basis Functions

Figure 11 and 12 show the performance of the approximate dynamic programming algorithms using first order and third order basis functions, respectively.

References

Anaya-Lara, O., N. Jenkins, J. Ekanayake, P. Cartwright, M. Hughes. 2009. *Wind energy generation: modelling and control*. Wiley.

Baert, D., A. Vervaet. 1999. Lead-acid battery model for the derivation of peukert's law. *Electrochimica acta* **44**(20) 3491–3504.

Bertsekas, D.P. 2011. *Dynamic Programming and Optimal Control 3rd Edition, Volume II*. CRC.

Bertsekas, D.P., J.N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Bowden, R.J., D.A. Turkington. 1984. *Instrumental variables*. Cambridge University Press.

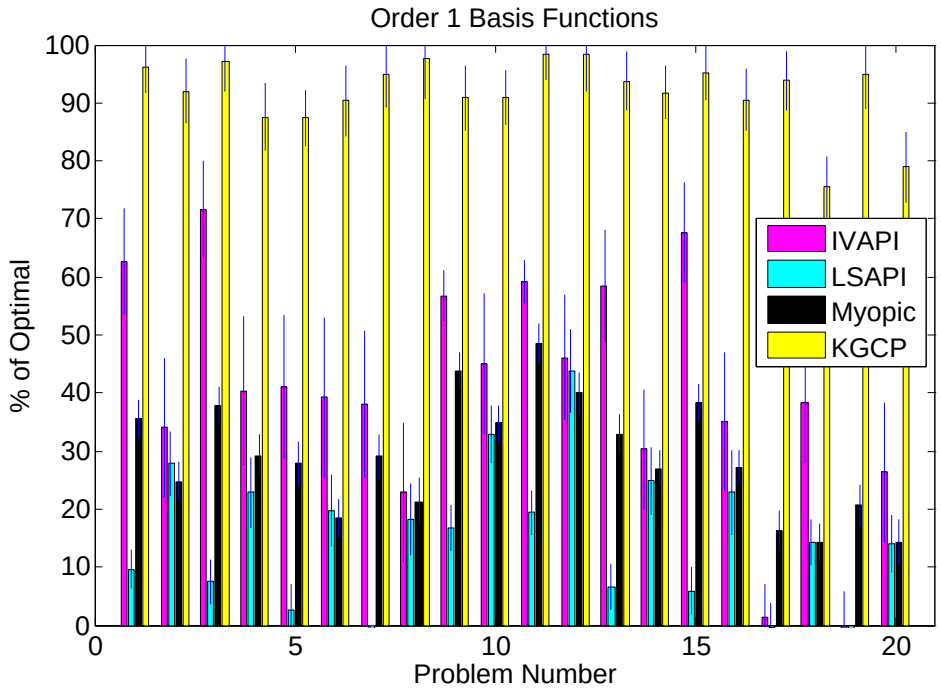


Figure 11 The algorithms use linear basis functions. We show the performance of Bellman error minimization using instrumental variables (IV) and least-squares Bellman error minimization (LS) along with direct policy search (KGCP).

Bradtke, Steven J., Andrew G. Barto. 1996. Linear least-squares algorithms for temporal difference learning. *Machine Learning* **22** 33–57.

Brown, Barbara G., Richard W. Katz, Allan H. Murphy. 1984. Time series models to simulate and forcast wind speed and power. *Journal of Climate and Applied Meteorology* **23** 1184–1195.

Brunet, Yves. 2011. *Energy Storage*. Wiley.

Burton, T., D. Sharpe, N. Jenkins, E. Bossanyi. 2001. *Wind energy: handbook*. Wiley.

Carmona, R. 2004. *Statistical analysis of financial data in S-Plus*. Springer Verlag.

Carmona, R., M. Ludkovski. 2005. Gas storage and supply guarantees: an optimal switching approach. *submitted to Management Science* .

Cartea, A., M.G. Figueroa. 2005. Pricing in electricity markets: a mean reverting jump diffusion model with seasonality. *Applied Mathematical Finance* **12**(4) 313–335.

Chen, Peiyuan, T. Pedersen, B. Bak-Jensen, Zhe Chen. 2010. Arima-based time series model of stochastic

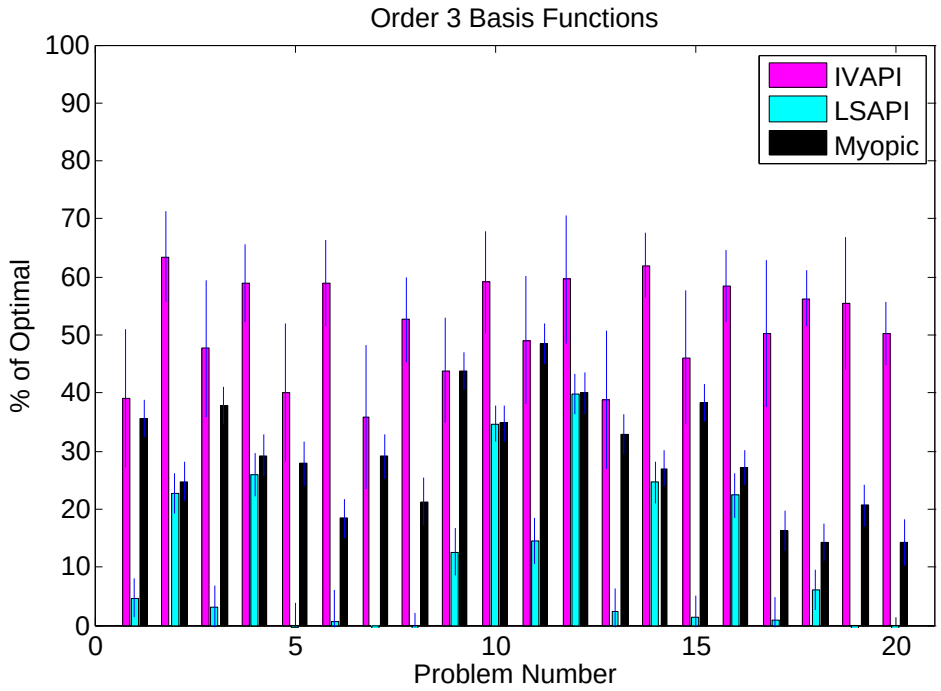


Figure 12 Third order basis functions. We show the performance of Bellman error minimization using instrumental variables (IV) and least-squares Bellman error minimization (LS).

wind power generation. *Power Systems, IEEE Transactions on* **25**(2) 667 –676. doi:10.1109/TPWRS.2009.2033277.

Costa, L.M., F. Bourry, J. Juban, G. Kariniotakis. 2008. Management of energy storage coordinated with wind power under electricity market conditions. *Probabilistic Methods Applied to Power Systems, 2008. PMAPS'08. Proceedings of the 10th International Conference on*. IEEE, 1–8.

De Farias, D.P., B. Van Roy. 2000. On the existence of fixed points for approximate value iteration and temporal-difference learning. *Journal of Optimization theory and Applications* **105**(3) 589–608.

DOE Handbook. 1995. Primer on Lead-Acid Storage Batteries.

Durbin, J. 1954. Errors in variables. *Revue de l'Institut international de statistique* 23–32.

Eydeland, A., K. Wolyniec. 2003. *Energy and power risk management: New developments in modeling, pricing, and hedging*. John Wiley & Sons Inc.

Eyer, J.M., J.J. Iannucci, G.P. Corey. 2004. Energy Storage Benefits and Market Analysis Handbook, A

Study for the DOE Energy Storage Systems Program. *Sandia National Laboratories, SAND2004-6177* .

Feinberg, E.A., D. Genethliou. 2005. Load forecasting. *Applied Mathematics for Restructured Electric Power Systems* 269–285.

Goodwin, G. C., K. S. Sin. 1984. *Adaptive Filtering and Control*. Prentice-Hall, Englewood Cliffs, NJ.

Greenblatt, J.B., S. Succar, D.C. Denkenberger, R.H. Williams, R.H. Socolow. 2007. Baseload wind energy: modeling the competition between gas turbines and compressed air energy storage for supplemental generation. *Energy Policy* **35**(3) 1474–1492.

Jones, D.R., M. Schonlau, W.J. Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* **13** 455–492.

Kempton, W., J. Tomic. 2005. Vehicle-to-grid power fundamentals: Calculating capacity and net revenue. *Journal of Power Sources* **144**(1) 268–279.

Kendall, M.G., A. Stuart. 1961. *The Advanced Theory of Statistics: Inference and Relationship*, vol. 2. Hafner Publishing Company.

Koller, D., R. Parr. 2000. Policy iteration for factored mdps. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-00)*. 326–334.

Lagoudakis, M.G., R. Parr. 2003. Least-squares policy iteration. *Journal of Machine Learning Research* **4** 1107–1149.

Lai, G., F. Margot, N. Secomandi. 2010. An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Operations research* **58**(3) 564–582.

Ma, Jun, Warren B. Powell. 2010. Convergence analysis of kernel-based on-policy approximate policy iteration algorithms for markov decision processes with continuous, multidimensional states and actions. *Working Paper* .

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

Pirrong, C., M. Jermakyan. 2001. The price of power: The valuation of power and weather derivatives. *October* **12** 2001.

Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the curses of dimensionality*. 2nd ed. John Wiley and Sons, New York.

Puterman, M. L. 1994. *Markov Decision Processes*. John Wiley & Sons, New York.

Scott, Warren R., Peter Frazier, Warren B. Powell. 2011a. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization* **21**(3).

Scott, Warren R., Peter Frazier, Warren B. Powell. 2011b. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization* **21**(3).

Secomandi, N. 2010. Optimal commodity trading with a capacitated storage asset. *Management Science* **56**(3) 449–467.

Sioshansi, R., P. Denholm, T. Jenkin, J. Weiss. 2009. Estimating the value of electricity storage in pjm: Arbitrage and some welfare effects. *Energy Economics* **31**(2) 269–277.

Söderström, T., P. Stoica. 1983. *Instrumental variable methods for system identification*, vol. 161. Springer-Verlag Berlin.

Sørensen, B. 1981. A combined wind and hydro power system. *Energy Policy* **9**(1) 51–55.

Sutton, R.S., A.G. Barto. 1998. *Reinforcement Learning*. The MIT Press, Cambridge, Massachusetts.

Sutton, R.S., H.R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, E. Wiewiora. 2009. Fast gradient-descent methods for temporal-difference learning with linear function approximation. *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 993–1000.

Swider, D.J. 2007. Compressed air energy storage in an electricity system with significant wind power generation. *IEEE transactions on energy conversion* **22**(1) 95.

Tsitsiklis, J., B. Van Roy. 1997. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control* **42** 674–690.

Tsitsiklis, J.N., B. Van Roy. May 1997. An analysis of temporal-difference learning with function approximation. *Automatic Control, IEEE Transactions on* **42**(5) 674–690. doi:10.1109/9.580874.

Watkins, C.J.C.H. 1989. Learning from delayed rewards. Ph.d. thesis, Cambridge University, Cambridge, UK.

Young, Peter. 1984. *Recursive Estimation and Time-Series Analysis*. Springer-Verlag, Berlin, Heidelberg.

Zhou, Y., A.A. Scheller-Wolf, N. Secomandi, S. Smith. 2011. Managing wind-based electricity generation with storage and transmission capacity. *Working Paper* .