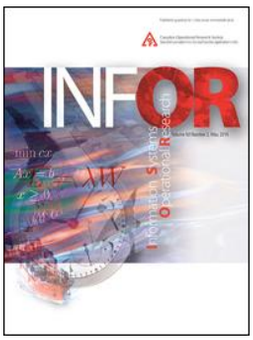

INFOR: Information Systems and Operational Research

ISSN: 0315-5986 (Print) 1916-0615 (Online) Journal homepage: https://www.tandfonline.com/loi/tinf20

Least squares policy iteration with instrumental variables vs. direct policy search: comparison against optimal benchmarks using energy storage

Somayeh Moazeni, Warren R. Scott & Warren B. Powell

To cite this article: Somayeh Moazeni, Warren R. Scott & Warren B. Powell (2019): Least squares policy iteration with instrumental variables vs. direct policy search: comparison against optimal benchmarks using energy storage, INFOR: Information Systems and Operational Research, DOI: 10.1080/03155986.2019.1624491

To link to this article: https://doi.org/10.1080/03155986.2019.1624491

Published online: 19 Jun 2019.

INFOR: INFORMATION SYSTEMS AND OPERATIONAL RESEARCH
https://doi.org/10.1080/03155986.2019.1624491

Least squares policy iteration with instrumental variables vs. direct policy search: comparison against optimal benchmarks using energy storage

Somayeh Moazeni[a], Warren R. Scott[b] and Warren B. Powell[b]

[a]School of Business, Stevens Institute of Technology, Hoboken, NJ, USA; [b]Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, USA

ABSTRACT

This article studies least-squares approximate policy iteration (API) methods with parametrized value-function approximation. We study several variations of the policy evaluation phase, namely, Bellman error minimization, Bellman error minimization with instrumental variables, projected Bellman error minimization, and projected Bellman error minimization with instrumental variables. For a general discrete-time stochastic control problem, Bellman error minimization policy evaluation using instrumental variables is equivalent to both variants of the projected Bellman error minimization. An alternative to these API methods is direct policy search based on knowledge gradient. The practical performance of these three approximate dynamic programming methods, (i) least squares API with Bellman error minimization, (ii) least squares API with Bellman error minimization with instrumental variables, and (iii) direct policy search, are investigated in the context of an application in energy storage operations management. We create a library of test problems using real-world data and apply value iteration to find their optimal policies. These optimal benchmarks are then used to compare the developed approximate dynamic programming policies. Our analysis indicates that least-squares API with instrumental variables Bellman error minimization prominently outperforms least-squares API with Bellman error minimization. However, these approaches underperform our direct policy search implementation.

ARTICLE HISTORY
Received 6 June 2018
Accepted 28 April 2019

KEYWORDS
Dynamic programming; approximate dynamic programming; approximate policy iteration; Bellman error minimization; direct policy search; energy storage

1. Introduction

It is long recognized that the powerful theory of discrete Markov decision processes (Puterman 1994) is limited by the well-known curse of dimensionality, which can be traced to the need to compute the value of being in each discrete state or, more problematically, the probability of transitioning from one discrete state to another given an action. Known as a *flat representation* in computer science, even small problems quickly blow up using this model. The field of approximate dynamic programming

CONTACT Somayeh Moazeni smoazeni@stevens.edu School of Business, Stevens Institute of Technology, Hoboken, NJ 07030, USA

(Bertsekas and Tsitsiklis 1996; Powell 2011; Bertsekas 2012) and reinforcement learning (Sutton and Barto 1998; Szepesvari 2010) have offered the hope of partially overcoming this problem by replacing the value function using a statistical approximation (in particular, linear regression), allowing us to draw on approximate versions of powerful algorithmic strategies such as value iteration and policy iteration (Puterman 1994).

Somewhat surprisingly, while we have found in our own work that approximate value iteration (AVI) works well for very specific problem classes (Topaloglu and Powell 2006; Simao et al. 2009; He et al. 2012; Nascimento and Powell 2013), it does not work as a general algorithmic strategy; our success with AVI has always been in the context of problems where we could exploit convexity. The computer science community has focused considerable efforts on a strategy called *Q*-learning, which involves learning the value of a state-action pair rather than just the value of being in a state (Sutton and Barto 1998). *Q*-learning enjoys rigorous convergence theory for lookup table representations (Tsitsiklis 1994), but this does not scale even to small problems. Approximating *Q*-factors with linear models is more art than science. Convergence results (Sutton et al. 2009b; Maei et al. 2009) do not contain any performance guarantees and are focused on establishing convergence of a particular approximating architecture. In addition, empirical comparisons against optimal benchmarks are scarce.

Perhaps for this reason, approximate policy iteration (API) has attracted considerable recent attention, see, e.g. (Bertsekas 2012; Busoniu et al. 2012) and the review in Powell and Ma (2011). API avoids the need to approximate the value of a state-action pair, and enjoys stronger convergence theory, although this always involves assumptions that are unlikely to be perfectly satisfied in practice. However, we are still unaware of any comparisons against optimal benchmarks.

This article uses the setting of a class of energy storage problems that requires balancing power from the grid and power from a stochastic, renewable source, to serve a load (that is sometimes time varying), with access to an energy storage device. The software containing these benchmark problems is available at https://castlelab.princeton.edu/datasets. We solve the created problems to obtain optimal policies, which provide us with rigorous benchmarks to accurately assess the quality of the solutions produced by different algorithmic strategies.

In this article, we focus on the use of linear architectures for approximating the value function, a strategy that has received the most attention in the literature. We model our problem in steady state, which allows us to use a powerful algorithmic strategy called *least-squares policy iteration* (LSPI), introduced by Lagoudakis and Parr (2003). This approach builds on the *least-squares temporal-difference* (LSTD) learning algorithm (see, e.g. Section 8 of Powell (2011)), proposed by Bradtke and Barto (1996) to estimate the value of a fixed policy. The LSTD method is one of the batch variants of the temporal difference (TD) learning. LSTD tends to be statistically efficient and extracts more information from training experiences and converges faster, compared to other typical TD learning methods. Lagoudakis and Parr (2003) introduce the idea of using sample experiences and linear approximation architectures for incremental policy improvement within a policy-iteration framework. To learn the

state-action value function, Lagoudakis and Parr (2003) discusses and compares two policy evaluation methods: (i) Bellman error minimizing approximation which minimizes the L_2 norm of the Bellman error, i.e., the difference between the left-hand side and the right-hand side of the Bellman equation, (ii) least-squares fixed-point approximation which seeks an approximate fixed point of the Bellman operator considering the orthogonal projection. The least-squares fixed-point approximation minimizes the projection of the distance that the Bellman error minimizing approximation minimizes. We investigate integrating instrumental variables into these two policy evaluation methods. We formally show that the policy evaluation with instrumental variables is equivalent to the policy evaluation with projected Bellman error minimization, as well as the hybrid policy evaluation combining both instrumental variables and projected Bellman error minimization.

Although our focus is on using exact benchmarks to derive insights into different algorithmic strategies, our choice of energy storage is motivated by the importance of this problem class. This dynamic optimization problem can be considered an extension of the stochastic inventory management problem, see, e.g. Porteus (2002). Growing interests in renewables and advances in energy storage technology have increased the interest in the energy storage operation optimization. For example, Barton and Infield (2004) analyzes three control policies and their corresponding expected revenues, assuming probabilistic models for load and wind, and a load-price curve for the electricity price. Carmona and Ludkovski (2005) and Lai et al. (2010) investigate the operation and valuation of a gas storage device. Greenblatt et al. (2007) and Swider (2007) discuss incorporating a compressed air energy storage with wind power generation in energy systems. Maximization of the expected market profit of a wind farm and hydro pumped storage over a finite horizon to comply with commitments in the market is addressed in Gonzalez et al. (2008). This problem is formulated as a two-stage stochastic optimization problem with uncertain prices and wind generation. Hu and Defourny (2017) investigate optimization of grid-level battery storage under battery aging consideration. A computationally efficient nonstationary direct policy search approach is developed in Moazeni et al. (2017) to optimize the operation of energy storage in the presence of a renewable resource to serve a load, while taking market risk into consideration. The problem of commodity storage management using high-dimensional models for forward prices is studied in Nadarajah et al. (2015), where approximate linear programing approaches for the resulting dynamic programming problem are investigated. Optimal operation an energy storage unit under random allowed operation times over a finite time horizon is studied in Moazeni and Defourny (2018). For a thorough review of this growing literature, we refer the reader to Moazeni et al. (2015), Halman et al. (2018), and the references therein.

For our energy storage application, we created a library of test problems using realistic data, constrained by the goal of creating optimal benchmarks. Optimal policies for these problems are computed using exact value iteration (Puterman 1994). While these problems are relatively simple, CPU times to estimate these policies typically ranged around two weeks. These benchmarks are then used to compare the approximate policies based on least-squares API and direct policy search. These

experiments show that least-squares API with instrumental variables Bellman error minimization performs significantly better than the least-squares API with the Bellman error minimization without instrumental variables. Yet even this advanced algorithmic strategy falls far short of optimal. Direct policy search outperforms (in terms of achieving percentage of optimality) both least-squares API policies by a wide margin. Since the structure of the policy in the policy improvement phase is identical in all algorithms, the issue is not the accuracy of the approximating architecture for the value function, but rather the estimation procedure, calling into question the validity of Bellman error minimization.

This article makes the following contributions. 1) We introduce and prove the consistency of a hybrid policy based on least squares API that combines instrumental variables and projected Bellman error minimization. 2) We develop a set of benchmark problems using the important context of energy storage where we are able to derive optimal policies. 3) We calculate approximate policies using least-squares API with basic Bellman error minimization, least-squares API with instrumental variables Bellman error minimization, and direct policy search, and compare the results against optimal benchmark policies and the myopic policy.

We provide an overview of API in Section 2, where several policy evaluation methods based on Bellman error minimization are also discussed. Direct policy search and knowledge gradient are presented in Section 3. The energy storage management problem, its underlying stochastic processes, and its stochastic dynamic optimization formulation are explained in Section 4. Performance of the computed approximate dynamic programming policies is investigated and compared with benchmark problems in Section 5. Limitations of the present study are discussed in Section 6. Concluding remarks are given in Section 7.

2. Approximate policy iteration algorithm

Stochastic dynamic programming for maximizing expected revenues relies on the Bellman optimality equation given by

$$V(S_t) = \max_{x \in \mathcal{X}_t} \mathbb{E}_{W_{t+1}}\left[C(S_t, x) + \gamma \; V(S_{t+1}) | S_t\right]. \tag{1}$$

Here, S_t refers to the state variable at time step t, $C(S_t, x)$ is the contribution function at state S_t and action x, $\mathcal{X}_t$ is the feasible region for decisions at time t, $V(\cdot)$ is the value function (around the pre-decision state S_t), and $0 \leq \gamma < 1$ is the discount factor. The expectation in Equation (1) is over the exogenous random changes, denoted by W_{t+1}, in the state of the system. The state variable at the next step is then obtained by the transition function $\mathcal{T}$, i.e., $S_{t+1} = \mathcal{T}(S_t, x, W_{t+1})$. Throughout, we use the convention that any variable indexed by t is known at time t.

Computing the expected value of $V(S_{t+1})$ is often challenging and needs to be approximated. To avoid this, a modified version of the Bellman equation based on the *postdecision state variables* can be adopted, see, e.g. Judd (1998), Bertsekas (2012) and Section 4.6 of Powell (2011) for a thorough discussion of postdecision states. The postdecision state variable, denoted by S_t^x, refers to the state immediately after being

in the pre-decision state S_t and taking the action x, but before any exogenous information (randomness) from the state transition has been revealed. The value of being in the postdecision state S_t^x is denoted by $V^x(S_t^x)$, and is related to the value function as follows:

$$V^x\left(S_t^x\right) \stackrel{\text{def}}{=} \mathbb{E}_{W_{t+1}}\left[V(S_{t+1}) \mid S_t^x\right].$$

Thus, the Bellman Equation (1) around the postdecision state variables can be written as

$$V^{x\prime}\left(S_{t-1}^{x\prime}\right) = \mathbb{E}_{W_t}\left[\max_{x\in\mathcal{X}_t} \left\{C(S_t,x) + \gamma\ V^x\left(S_t^x\right)\right\} \mid S_{t-1}^{x\prime}\right]. \tag{2}$$

The expectation being outside of the maximum operator allows us to solve the inner maximization problem using deterministic optimization techniques. The policy optimal with respect to the postdecision value function is then given by

$$X^\pi(S_t) \in \text{argmax}_{x\in\mathcal{X}_t}\ \left\{C(S_t,x) + \gamma\ V^x\left(S_t^x\right)\right\}. \tag{3}$$

For most applications, however, when the state variable is multidimensional and continuous, Bellman Equations (1) or (2) cannot be solved exactly; as a result a large field of research on approximation techniques has evolved, see, e.g. Bertsekas and Tsitsiklis (1996), Sutton and Barto (1998), Szepesvari (2010), Powell (2011). We focus on the widely used approach of approximating the value function with linear architectures of the form

$$\hat{V}^x\left(S_t^x\right) \stackrel{\text{def}}{=} \sum_{k=1}^{K} \theta_k\ \phi_k\left(S_t^x\right) = \theta^\top \phi\left(S_t^x\right), \tag{4}$$

where $\{\phi_k(S_t^x)\}_{k=1}^K$ is a set of K given basis functions, $\phi(S_t^x)$ is the column feature vector with elements $\phi_k(S_t^x)$, and θ is the column parameter vector associated with the basis functions. For further discussion on basis functions, see, e.g. Menache et al. (2005), Heuberger et al. (2005), Konidaris et al. (2011). Substituting approximate postdecision value functions (4) into Equations (2) and (3), we get

$$\theta^\top \phi\left(S_{t-1}^{x\prime}\right) \approx \mathbb{E}\left[C(S_t,x) + \gamma\ \theta^\top \phi\left(S_t^x\right) \mid S_{t-1}^{x\prime},\ x = X^{\hat{\pi}}(S_t|\theta)\right], \tag{5}$$

where

$$X^{\hat{\pi}}(S_t|\theta) \in \text{argmax}_{x\in\mathcal{X}_t}\ \left[C(S_t,x) + \gamma\ \theta^\top \phi\left(S_t^x\right)\right]. \tag{6}$$

A value of the weight vector θ, at which Equation (5) holds for all states, would yield the optimal value function and an optimal policy. However, in general, a fixed point satisfying this equation does not exist (De Farias and Van Roy 2000), and only a value of θ which approximately solves Equation (5) can be sought.

The approximate policy iteration (API) approach alternates between a *policy evaluation phase* which approximately evaluates the current policy by estimating the value function, and a *policy improvement phase* in which a new (improved) policy is generated. For API with linear architectures to approximate the value functions, the policy improvement phase employs Equation (6), while the development of a policy evaluation step relies on Equation (5). The procedure is repeated for M iterations, where M is a user-controlled parameter, allowing to generate improving policies that converge to the optimal policy.

2.1. Least-squares approximate policy iteration algorithm

A family of API algorithms, referred to as least-squares approximate policy iteration (LSPI), was introduced in Lagoudakis and Parr (2003). Assuming finite states and actions, LSPI approximates the value of state-action pairs (Q-factors) with a linear architecture, and incrementally improves the policy within a policy-iteration framework. Several methods of approximately solving the Bellman equation and their geometric interpretations are discussed in Lagoudakis and Parr (2003). Instances of the LSPI algorithmic family differ in the specific policy evaluation procedure employed. This algorithm extends the least-squares temporal-difference (LSTD) learning algorithm of Bradtke and Barto (1996) to control problems. For further discussion on the convergence of TD learning and the LSPI method, the reader is referred to Tsitsiklis and Van Roy (1997), De Farias and Van Roy (2000), Lagoudakis and Parr (2003).

We draw on the foundation provided in Bradtke and Barto (1996), adopted for the postdecision state. We focus on the off-policy case, where a set of N postdecision states $\{S^{x'_n}_{t-1,n}\}_{n=1}^{N}$ are generated randomly, and then for each sample n, we simulate the state $S_{t,n}$ and the corresponding next state decision $x_n \overset{\text{def}}{=} X^{\hat{\pi}}(S_{t,n}|\theta)$ computed by Equation (6). For a set of N samples $\{(S^{x'_n}_{t-1,n}, S_{t,n}, x_n, C(S_{t,n}, x_n), S^{x_n}_{t,n}) \mid n = 1, 2, ..., N\}$, define

$$C_t \overset{\text{def}}{=} \begin{bmatrix} C(S_{t,1}, x_1) \\ \vdots \\ C(S_{t,N}, x_N) \end{bmatrix}, \qquad \Phi_t \overset{\text{def}}{=} \begin{bmatrix} \phi\left(S^{x_1}_{t,1}\right)^\top \\ \vdots \\ \phi\left(S^{x_N}_{t,N}\right)^\top \end{bmatrix}. \tag{7}$$

Here, C_t is a column vector of dimension N, while Φ_t is a matrix of size $N \times K$. Each row of Φ_t contains the value of all basis functions for a certain post decision state variable $S^{x_n}_{t,n}$. Similarly, the nth row of the matrix Φ_{t-1} is $\phi(S^{x'_n}_{t-1,n})^\top$. Variations of the least-squares API algorithm aim to find a weight vector θ that satisfies the Bellman equation as closely as possible by solving a least-squares problem (see Section 2.2) for the following over-constrained linear system

$$C_t \approx (\Phi_{t-1} - \gamma\Phi_t)\theta. \tag{8}$$

For a given value of θ, we refer to $C_t - (\Phi_{t-1} - \gamma\Phi_t)\theta$ as *Bellman errors* or *Bellman residuals*, which expresses the difference between the left-hand side and the right-

hand side of the Bellman equation. An overview of the least-squares API algorithm is given in Figure 1.

Next, we elaborate on the policy evaluation step.

2.2. Policy evaluation procedures

We focus on policy evaluation approaches based on LSTD and investigate variants of policy evaluation algorithms using Bellman error minimization and projected Bellman error minimization. The following assumption is made to rule out redundant parameters in the value function approximation architecture:

Assumption 2.1. *The matrices* Φ_{t-1}, $(\Phi_{t-1}-\gamma\Phi_t)$, *and* $\Phi_{t-1}^{\top}(\Phi_{t-1}-\gamma\Phi_t)$ *have full rank, and* $K \leq N$.

For least-squares Bellman error minimization, the objective is to minimize the Euclidean norm of the Bellman errors,

$$\min_{\theta} \ \|C_t - (\Phi_{t-1} - \gamma\Phi_t)\theta\|_2^2. \tag{9}$$

Applying the typical method of least-squares, a solution of Equation (9) equals

$$\hat{\theta}_{\text{LSBEM}} \stackrel{\text{def}}{=} \Big((\Phi_{t-1}-\gamma\Phi_t)^{\top}(\Phi_{t-1} - \gamma\Phi_t)\Big)^{-1}(\Phi_{t-1}-\gamma\Phi_t)^{\top}C_t, \tag{10}$$

to which we refer as the *least-squares bellman error minimization* (LSBEM) estimator. 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.

A class of simple and computationally efficient techniques to obtain consistent estimates, without modeling the noise, is *instrumental variable* methods. An instrumental variable is a variable that is correlated with the regressors, but uncorrelated with the errors in the regressors and the observations, see, e.g. Durbin (1954), Kendall and Stuart (1961), Söderström and Stoica (1983), Bowden and Turkington (1984), Young (2011). Appendix A provides a brief overview of instrumental variable methods. Instrumental variables have been previously used in the context of API algorithms,

Least-Squares Approximate Policy Iteration Algorithm

(01) Initialize θ.
(02) **for** $m = 1$ **to** M **(Policy Improvement Iterations)**
(03) Given θ, define the policy $x = X^{\pi}(S_t|\theta)$ using equation (6)
(04) **for** $n = 1$ **to** N
(05) Simulate a random post-decision state, $S_{t-1,n}^{x'_n}$
(06) Compute $\phi(S_{t-1,n}^{x'_n})$
(07) Simulate the state transition to get $S_{t,n}$
(08) Determine the decision $x_n = X^{\pi}(S_{t,n}|\theta)$
(09) Compute $C(S_{t,n}, x_n)$ and $\phi(S_{t,n}^{x_n})$ corresponding to decision x_n
(10) **End**
(11) Update θ using equation (10), or (11), or (14), or (15). **(Policy Evaluation)**
(12) **End**

Figure 1. Least-squares API algorithm.

see, e.g. Bradtke and Barto (1996), but otherwise have not received much attention, even in the reinforcement learning literature. The method of instrumental variables is used in LSTD to compensate for the use of Monte Carlo simulation to approximate the expected cost in $\bar{C}_{t,i} \stackrel{\text{def}}{=} \mathbb{E}[C(S_{t,i}, X\tilde{\pi}(S_{t,i}|\theta))|S^x_{t-1,i}]$. This results in the *instrumental variables bellman error minimization* (IVBEM) vector,

$$\hat{\theta}_{\text{IVBEM}} \stackrel{\text{def}}{=} \left(\Phi_{t-1}^\top(\Phi_{t-1} - \gamma\Phi_t)\right)^{-1}\Phi_{t-1}^\top C_t. \tag{11}$$

It can be proved that the linear least-squares function approximation with the instrumental variables method leads to a consistent estimator ($\hat{\theta}_{\text{IVBEM}} \to \theta$ as $N \to \infty$, with probability one). The proof references the consistency properties of the method of instrumental variables by showing that the columns of Φ_{t-1} are appropriate instrumental variables (see Lemma 2 in Bradtke and Barto (1996) and Appendix A). Note that the matrix $(\Phi_{t-1}^\top(\Phi_{t-1} - \gamma\Phi_t))$ could have negative eigenvalues, unlike $((\Phi_{t-1} - \gamma\Phi_t)^\top(\Phi_{t-1} - \gamma\Phi_t))$.

The idea of projected Bellman error minimization, also called *least-squares fixed-point approximation*, is to first project the Bellman errors down onto the space spanned by the basis functions defining the value function and then minimize the Bellman errors, see Lagoudakis and Parr (2003), Sutton et al. (2009a). Define the projection operator

$$\Pi_{t-1} = \Phi_{t-1}\left(\Phi_{t-1}^\top\Phi_{t-1}\right)^{-1}\Phi_{t-1}^\top, \tag{12}$$

on the space spanned by the basis functions; see Tsitsiklis and Van Roy (1997) for the original derivation of this mapping, or Section 8.2.3 of Powell (2011). It follows from Assumption 2.1 that the matrix Φ_{t-1} has full column rank, and hence Π_{t-1} is well-defined. We refer to $\Pi_{t-1}C_t - \Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t)\theta$ as the *projected Bellman error*. Taking a least squares approach, we find θ by minimizing the norm of the projected Bellman error

$$\min_\theta \ \|\Pi_{t-1}C_t - \Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t)\theta\|_2. \tag{13}$$

The least-squares estimator of θ then yields the *least-squares projected bellman error minimization* (LSPBEM) estimator, given by

$$\hat{\theta}_{\text{LSPBEM}} \stackrel{\text{def}}{=} \left((\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))^\top(\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))\right)^{-1}(\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t))^\top\Pi_{t-1}C_t, \tag{14}$$

To establish a consistent estimator for θ, similar to the derivation of Equation (11), $Z = \Phi_{t-1}$ can be used as an instrumental variable, see Appendix A or the proof in Bradtke and Barto (1996). We refer to the resulting estimator as the *instrumental variables projected bellman error minimization* (IVPBEM),

$$\hat{\theta}_{\text{IVPBEM}} \stackrel{\text{def}}{=} \left(\Phi_{t-1}^\top\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t)\right)^{-1}\Phi_{t-1}^\top\Pi_{t-1}C_t. \tag{15}$$

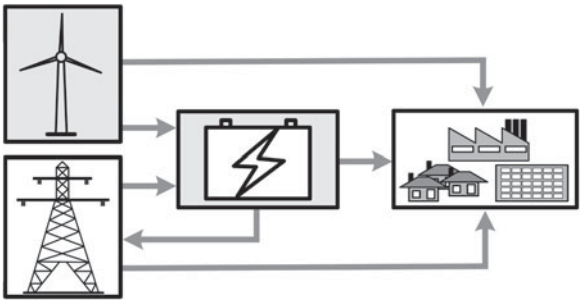

Figure 2. The energy system diagram.

Consistency of the IVPBEM estimator is established in Theorem B.1 in Appendix B. For similar results on the consistency of LSTD methods, the reader is referred to Kolter (2011), Yu (2012), Dann et al. (2014). We note that $\Pi_{t-1}\Phi_{t-1}$ could also have been used as the instrumental variable instead of Φ_{t-1}. However, it is easy to see that the obtained estimator would be equivalent to that in Equation (15).

The following proposition formalizes the relationship among the three estimators in Equations (11), (14), and (15), under Assumption 2.1.

Proposition 2.1. *The policy evaluation algorithms IVBEM, LSPBEM, and IVPBEM are equivalent.*

A proof of Proposition 2.1 is provided in Appendix C. This result was also noted in Antos et al. (2008), and further discussed in Szepesvari (2010) and Dann et al. (2014). Our numerical results in Section 5 indicate that the LSBEM is not equivalent to IVBEM, and whence, the two others.

3. Direct policy search

An alternative approximate dynamic programming approach to find the policy parameter vector θ is direct policy search. Consider policies parameterized by θ of the form in Equation (6), in which the value function has been replaced by a function approximator linear in adjustable parameters and a feature vector representing states. In contrast to policy iteration or value iteration methods, the goal in direct policy search is not necessarily to estimate the value at every state which is close (with respect to some norm) to the true value function; the objective is to find a parameter vector θ for which the parametrized policy performs well, i.e., it solves the following stochastic optimization problem

$$\max_{\theta} \; V^{\pi}(S_0), \tag{16}$$

given the policy structure $X^{\pi}(S_t|\theta)$, and the initial state S_0. The value of θ which maximizes Equation (16) produces the best policy within the class of polices, $X^{\pi}(S_t|\theta)$. Solving this problem becomes challenging, particularly as the dimension of θ grows. Furthermore, the optimization problem given by Equation (16) is typically nonconvex and nonseparable. For further discussion on direct policy search

methods and computational approaches, see Moazeni et al. (2016, 2017). Note that, in direct policy search, we only need to consider features which are functions of the decisions.

Classic stochastic optimization algorithms can be used to sequentially choose policies to simulate. 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, e.g. Scott et al. (2011). In our experiments, θ is limited to two or three dimensions. The KGCP approach is explained next.

3.1. The knowledge gradient for direct policy search

For a given θ^i, a noisy observation of the objective in (16) can be obtained by simulating

$$\mu(\theta^i) \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} \gamma^t C_t\big(S_t, X^\pi\big(S_t|\theta^i\big)\big). \tag{17}$$

The KGCP policy for optimizing θ treats the objective function $\mu(\theta)$ as a Gaussian process regression. This policy relies on a criterion which chooses the next value of θ for which a noisy observation of $\mu(\theta)$ is simulated. The KGCP quantifies how much the maximum of the objective is expected to increase by getting an additional noisy observation of $\mu(\theta)$ at a particular value of θ.

More formally, let $\mathcal{F}^n$ be the sigma-algebra generated by $\theta^0, \ldots, \theta^{n-1}$ and the corresponding noisy observations of $\mu(\theta^0), \ldots, \mu(\theta^{n-1})$. Denote the updated expected values of μ at θ^i, conditioned on $\mathcal{F}^n$, by $\mu^n(\theta^i)$. Define the KGCP quantity as

$$\nu^{\text{KG},n}(\theta) \stackrel{\text{def}}{=} \mathbb{E}\left[\max_{i=0,\ldots,n} \mu^{n+1}(\theta^i)|\mathcal{F}^n, \theta^n = \theta\right] - \max_{i=0,\ldots,n} \mu^n(\theta^i)|_{\theta^n=\theta}. \tag{18}$$

The next sampling decision is then chosen to maximize the KGCP quantity,

$$\theta^n \in \arg\max_{\theta} \ \nu^{\text{KG},n}(\theta). \tag{19}$$

After N observations, the implementation decision θ^* is chosen by maximizing $\mu^N(\theta)$, i.e.,

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

In the Gaussian process regression framework, $\mu^{n+1}(\theta)$ given $\mathcal{F}^n$ is normally distributed for each value of θ, and consequently the KGCP quantity and the KGCP policy can be calculated exactly, see Scott et al. (2011) or Chapter 16 of Powell and Ryzhov (2012). The KGCP policy converges asymptotically to the optimal value of θ for problem (16).

4. Benchmark application: energy storage operation

Consider a power system as shown in Figure 2, involving an intermittent energy supply, an electricity demand, an interconnecting grid, and a battery storage device. At time t, the energy flows are given by the vector $x_t \overset{\text{def}}{=} (x_t^{WR}, x_t^{GR}, x_t^{RD}, x_t^{WD}, x_t^{GD})$, where x_t^{IJ} denotes the amount of energy transferred from I to J at time step t. The superscript W stands for energy source (wind), D for demand, R for storage, and G for grid. These entities are assumed to be nonnegative except x_t^{GR}. A negative value for x_t^{GR} refers to selling electricity from the storage to the grid.

Denote the total electricity demand (in MWh) over the time period starting at $t-\Delta t$ and ending at t, by D_t. At every time step, the demand D_t must be served through the wind energy, available energy from the storage device, or energy purchased from the grid,

$$x_t^{GD} + \eta^{\text{discharge}} x_t^{RD} + x_t^{WD} = D_t. \tag{20}$$

Here, $\eta^{\text{discharge}} \in (0, 1)$ denotes the *discharging efficiency rate*.

The wind energy generated during the time period $[t-\Delta t,\ t)$, denoted by E_t, first serves the demand and the surplus is charged into the storage device for the future use, i.e.,

$$x_t^{WD} = \min\{E_t,\ D_t\}, \tag{21}$$

$$x_t^{WR} + x_t^{WD} = E_t. \tag{22}$$

Let R^{cap} indicate the total capacity of the storage device. Define the constants $\Delta R^{\min}$ and $\Delta R^{\max}$ as the minimum and maximum fractions of the storage device that can be charged over Δt. For example, for a lead acid battery with a $C/10$ maximum charge and discharge rates, and $\Delta t = 15\text{min}, \Delta R^{\min} = -1/40$ and $\Delta R^{\max} = 1/40$. To avoid charging or discharging the storage device faster than the permitted rates, x_t^{GR} must satisfy:

$$\frac{\Delta R^{\min} R^{\text{cap}}}{\eta^{\text{discharge}}} \le x_t^{GR} \le \frac{\Delta R^{\max} R^{\text{cap}}}{\eta^{\text{charge}}}, \tag{23}$$

where $\eta^{\text{charge}} \in (0, 1)$ is the *charging efficiency rate*. Similarly, to ensure that the storage device is not discharged faster than allowed when sending energy from the storage unit to the demand, we include the constraint

$$0 \le x_t^{RD} \le \Delta R^{\max} R^{\text{cap}}. \tag{24}$$

Since both $\eta^{\text{discharge}} < 1$ and $\eta^{\text{charge}} < 1$, transmitting energy from the grid to the demand via the storage would be less efficient and more costly than directly sending energy from the grid to the demand. The energy flow x_t yield the next storage state $R_{t+\Delta t}(x_t)$ equal to

$$R_{t+\Delta t}(x_t) = \max\left\{R^{\min},\ \min\left\{R_t + \frac{\eta^{\text{charge}}\left(x_t^{GR} + x_t^{WR}\right) - x_t^{RD}}{R^{\text{cap}}}, 1\right\}\right\}, \tag{25}$$

where $R^{\min}$ is the minimum fraction of the capacity of the storage device that must remain full. For example, stationary lead-acid batteries with tubular plates, which are one of the lowest cost technologies for energy storage, should not be discharged below 20% of their capacity, i.e., $R^{\min} = 0.20$, see, e.g. Brunet (2011).

Constraints (20)–(25), together with the standard measurability conditions, define the set of admissible policies. Uncertainties in the energy supply (wind), demand, and prices are expressed through stochastic processes, to be explained in the following subsections.

4.1. Wind energy

The energy output from the wind turbine over $[t,\ t+\Delta t)$ is computed by $E_t = \frac{10^{-8}}{72}\ C_p\ \rho\ 50^2\pi\ w_t^3\ \Delta t$, see, e.g. MacKay (2009), Moazeni et al. (2015). Here, $\rho = 1.225\ \text{kg/m}^3$ is the density of air, $C_p = 0.45$ is the power coefficient, w_t denotes the wind speed measured in meters per second, and Δt is stated in seconds. Velocity of the wind, w_t is given by $w_t = (Y_t^E + \mu_E)^2$. Here, Y_t^E evolves by an AR(1) model (see, e.g. Brown et al. (1984)), $Y_t^E = \phi_E Y_{t-\Delta t}^E + \sigma_E\sqrt{\Delta t}\tilde{\epsilon}_t$, where $\tilde{\epsilon}_t \sim \mathcal{N}(0,1)$. Using 15-min data from the wind speeds at Maryneal, Texas and applying the Yule-Walker equations (see, e.g. Carmona (2004)) to fit the above model, we obtain $\mu_E = 1.4781, \phi_E = 0.7633, \sigma_E = 0.4020$.

4.2. Electricity prices

Similar to Cartea and Figueroa (2005), we model the real-time electricity prices by $P_t = \exp(Y_t^s + Y_t^{ds}) - c$, with a deterministic seasonal component Y_t^s. The deseasonalized log prices are modeled by a discretized mean reverting jump diffusion process $Y_t^{ds} = Y_{t-\Delta t}^{ds} + \lambda_P(\mu_P - Y_{t-\Delta t}^{ds})\Delta t + \sigma_P\sqrt{\Delta t}\ \tilde{\epsilon}_t + J_t$, where μ_P is the long term equilibrium price, λ_P is the mean reversion rate, $\tilde{\epsilon}_t \sim \mathcal{N}(0,1)$, and J_t denotes the jump over the interval $[t-\Delta t,\ t)$. The jumps are modeled by the i.i.d. process, $J_t = \tilde{\alpha}_t^J 1(\tilde{u}_t < p_J)$. Here, $\tilde{\alpha}_t^J \sim \mathcal{N}(0, \sigma_J)$ is the jump size, p_J is the probability of a jump over a time interval of length Δt, and $\tilde{u}_t \sim \text{unif}(0,1)$. The constant parameter c equals one minus the minimum value of P_t in the data set. We use the approach in Cartea and Figueroa (2005) to estimate the model parameters. For the real time electricity prices at the PJM Western Hub dataset, we get $\sigma_J = 0.4229,\ p_J = 0.0170,\ \lambda_P = 1800.9, \mu_P = 4.1995, \sigma_P = 11.0971$, and $c = 27.2531$.

4.3. Electricity demand

Eydeland and Wolyniec (2003) outline typical models for residential, commercial, and industrial power demands. While industrial power demand is relatively stable, residential power demand is highly dependent upon the temperature and exhibits seasonal variations. Various load models and forecasting methods are discussed in Feinberg and

Genethliou (2010). Similar to Pirrong and Jermakyan (2008), Moazeni et al. (2015), we adopt a demand model with seasonality components $D_t = m_t^{hour} + m_t^{month} + D_t^{ds}$. Here, m_t^{hour} and m_t^{month} indicate the hour-of-week seasonal and the month-of-year seasonal components, and D_t^{ds} is the deseasonalized load. The deseasonalized load D_t^{ds} evolves with a linear autoregressive model $Y_t^D = \phi_D Y_{t-\Delta t}^D + \sigma_D \sqrt{\Delta t}\tilde{\epsilon}_t$ For the hourly ERCOT energy load data we obtained, $\phi_D = 0.9636, \sigma_D^2 = 914870$.

4.4. Stochastic dynamic optimization formulation

We formulate this multistage stochastic optimization problem using stochastic dynamic programming. The contribution function at every time step t then is the dollar value of energy sold minus the amount bought from the grid, assuming that an identical energy price is used for both withdrawal and injection:

$$C(S_t, x_t) = P_t D_t - P_t\left(x_t^{GR} + x_t^{GD}\right). \tag{26}$$

The goal is to find a policy which maximizes the accumulated expected discounted future rewards,

$$\max_{\pi} \ \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t C(S_t, X^{\pi}(S_t))\right]. \tag{27}$$

We define the state variable, $S_t = (R_t,\ E_t,\ D_t,\ P_t)$, and the postdecision state variable, $S_t^x = (R_{t+\Delta t}(x),\ E_t,\ D_t,\ P_t)$. Recall that R_t is the fraction of the storage device that is full, E_t is the current amount of wind energy, D_t indicates the current energy demand, and P_t is the current spot price of electricity selling to (or purchasing from) the electrical grid. The exogenous information process is defined as the random changes in the state of the system, $W_{t+\Delta t} = \{\hat{E}_{t+\Delta t}, \hat{D}_{t+\Delta t}, \hat{P}_{t+\Delta t}\}$, explaining exogenous changes in E_t, D_t and P_t, that can may be state dependent as well as time dependent.

Similar power system models have been studied in Moazeni et al. (2015) and Moazeni et al. (2017) for illustrating other algorithmic strategies. Proposition 3 in Moazeni et al. (2017) establishes a closed-form solution for the exact stochastic dynamic optimization in the finite time horizon setting, under the assumptions that charging and discharging efficiency rates are one $\eta^{\text{charge}} = \eta^{\text{discharge}} = 1$ and the storage device can be fully charged and discharged over Δt. Under these assumptions, Proposition 3 in Moazeni et al. (2017) shows that the optimal policy is a two-threshold policy, ruled by the sign of $P_t - \mathbb{E}[P_{t+1}|S_t]$, i.e., charge the storage device if $P_t \leq \mathbb{E}[P_{t+1}|S_t]$ and discharge if $P_t \geq \mathbb{E}[P_{t+1}|S_t]$.

Next section explains our computational results from the least-squares API algorithms in Section 2, the direct policy search described in Section 3, for problem (27), and compares the performance of the computed policies to the optimal policies from exact stochastic dynamic optimization.

Table 1. Benchmark problems with discrete state spaces: number of discretization levels for time (1 = steady state) and load (1 = deterministic).

Problem		Number of discretization levels					Parameters			
Number	Type	Time τ_t	Resource R_t	Price P_t	Demand D_t	Wind E_t	Wind $\frac{E_t}{D_t}$	Storage $\frac{R_t}{D_t}$	RTE	Charge Rate
1	Full	1	33	20	1	10	0.1	2.5	0.81	C/10
2	Full	1	33	20	1	10	0.1	2.5	0.81	C/1
3	Full	1	33	20	1	10	0.1	2.5	0.70	C/10
4	Full	1	33	20	1	10	0.1	2.5	0.70	C/1
5	Full	1	33	20	1	10	0.2	2.5	0.81	C/10
6	Full	1	33	20	1	10	0.2	2.5	0.81	C/1
7	Full	1	33	20	1	10	0.2	2.5	0.70	C/10
8	Full	1	33	20	1	10	0.2	2.5	0.70	C/1
9	Full	1	33	20	1	10	0.1	5.0	0.81	C/10
10	Full	1	33	20	1	10	0.1	5.0	0.81	C/1
11	Full	1	33	20	1	10	0.1	5.0	0.70	C/10
12	Full	1	33	20	1	10	0.1	5.0	0.70	C/1
13	Full	1	33	20	1	10	0.2	5.0	0.81	C/10
14	Full	1	33	20	1	10	0.2	5.0	0.81	C/1
15	Full	1	33	20	1	10	0.2	5.0	0.70	C/10
16	Full	1	33	20	1	1	0.2	5.0	0.70	C/1
17	BA	96	33	20	1	1	–	–	0.81	C/10
18	BA	96	33	20	1	1	–	–	0.81	C/1
19	BA	96	33	20	1	1	–	–	0.70	C/10
20	BA	96	33	20	1	1	–	–	0.70	C/1

5. Numerical experiments

Our main objective in this section is to assess the performance of the two variants of the least-squares API, and direct policy search. Throughout, the time step is $\Delta t = 15$ min and the discount factor is $\gamma = 99.90\%$. We found that discount factors of $\gamma = 99\%$ or smaller produce policies that are relatively myopic, and do not store energy for extended periods.

5.1. Benchmark problems with discrete state spaces

We first consider finite, discretized state and action spaces with a fixed probability transition matrix. An exact solution for an infinite horizon problem can be found using the value iteration method, see, e.g. Puterman (1994). Computing these optimal policies typically requires approximately two weeks of CPU time. In this method, $V^0(s)$ is initialized to a constant for all states s in the state space, and at each iteration n, the algorithm updates the value function at each state using

$$V^n(s) = \max_x \left\{ C(s,x) + \gamma \sum_{s' \in \mathcal{S}} V^{n-1}(s') \Pr\left(s'|s,x\right) \right\}, \quad \text{for every } s \in \mathcal{S}.$$

To establish a set of benchmark problems, we consider 20 instances of the energy storage operation optimization problem explained in Section 4. Table 1 summarizes these problems. Here, problem type "Full" refers to the problem in Figure 2 with energy from the wind source and the grid serving an electricity demand. In the absence of a wind source and demand, the storage device is used solely to buy/sell the electricity from/to the grid. This model considers only trading between the

storage and the grid to take advantage of price variations and is referred to as the battery arbitrage problem. In Table 1, the problem type "BA" refers to a battery arbitrage problem. We discretized the state space in the benchmark problems and then created fixed probability transition matrices for the exogenous information process in order to create a true discrete process. Table 1 also reports 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 average maximum 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 maximum charge and discharge rate of the storage device. For example, $C/10$ indicates that the storage device can be fully charged or discharged within 10 hours. The transition matrix of the electricity prices was fitted using the PJM Western Hub real time prices (with and without time of day). The transition matrix of the load was fitted using the load of the PJM Mid-Atlantic Region (with time of day). The transition matrix for the wind was fitted 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, i.e., $S_t = (R_t, E_t, P_t)$. For these experiments, time and demand are held fixed in order to keep the benchmark problems computationally tractable, as the exact value iteration, even for these simplified problems, requires approximately two weeks on a 3 Ghz processor. For Problems 17−20, the state variable is given by $S_t = (\tau_t, R_t, P_t)$, where τ_t is the time-of-day (96 corresponding to 15-minute intervals in a day), R_t is the resource level, and P_t is the electricity price. To implement the least-squares API methods, quadratic basis functions ($\phi_k(S) = S_i S_j$ for $i, j = 1, \ldots, |S|$) are used.

To specify reasonable values for the maximum number of policy evaluation and policy improvement iterations, M and N in Figure 1, we implement the LSAPI method using instrumental variables Bellman error minimization (IVBEM) on the 17$^{\text{th}}$ benchmark problem several times. For this test problem, as illustrated in Figure 3, most of the improvement has occurred before $N = 5000$ iterations of policy evaluations and $M = 30$ iterations of policy improvement step. Thus, in the rest of this section, we fix $N = 5000$ and $M = 30$.

5.2. Least-squares API and direct policy search for benchmark problems

This section compares the least-squares API methods, the myopic policy, and the direct policy search based on KGCP. Subsequently, the policy computed by the algorithm 1 with the policy evaluation $\hat{\theta}_{\text{LSBEM}}$, in Equation (10), is referred to as LSAPI. We also refer to the policy computed by the algorithm 1 with the policy evaluation $\hat{\theta}_{\text{IVBEM}}$ in Equation (11) by IVAPI. The myopic policy discharges the storage device as quickly as possible and keeps the charge level at its minimum allowed level since then. The value of the myopic policy is still positive due to the wind source.

We run each algorithm 100 times. For each run of the algorithms, the final policies computed by each algorithm are evaluated on the same set of sample paths, $\omega \in \Omega$, where ω is generated from the discretized exogenous information process. We then record the average percentage of optimality across the 100 runs. For a policy π, the average percentage of optimal is computed by

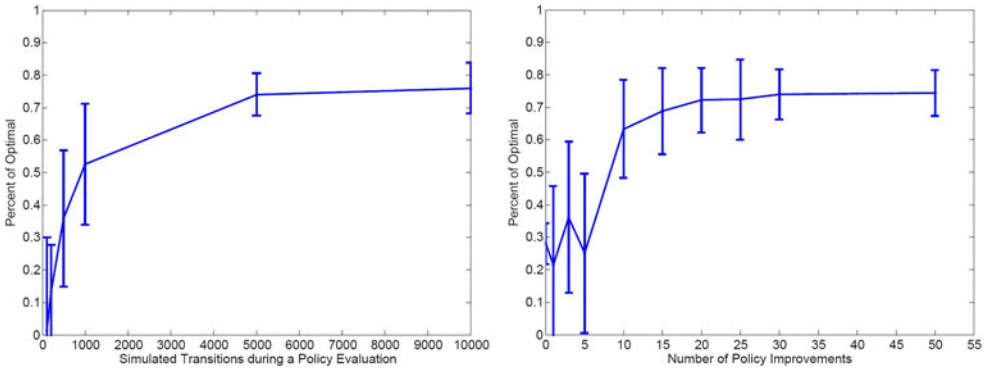

Figure 3. Progress of the least-squares API method using IVBEM policy evaluation for benchmark problem 17, as a function of N (left plot) for $M = 30$, and as a function of M (right plot) for $N = 5000$.

$$\% \text{ of optimality} = \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\frac{\hat{V}^{\pi}(S_0(\omega))}{V^{*}(S_0(\omega))}, \tag{28}$$

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. Here, $\hat{V}^{\pi}(S_0(\omega))$ is the value of the policy π run on the sample path ω, starting at the state $S_0(\omega)$. In Equation (28), $V^{*}(S_0(\omega))$ is the true value of the optimal policy for state $S_0(\omega)$ computed using the exact value iteration method.

Similarly, we implement the direct policy search using KGCP 100 times, and compute the average percent of optimal and its standard deviation. To implement 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.

Figure 4 illustrates the percentage of optimal corresponding to each of these policies for the benchmark problems. This figure shows that IVAPI significantly outperforms LSAPI for all benchmark problems, but still underperforms the optimal policy by a wide margin. Direct policy search produces solutions that are on average 91.80% of optimal, and are always at least 70% of optimal for Problems 1−16. This suggests that for the benchmark problems on the application of interest, direct policy search is more robust relative to the least-squares API methods.

In order to reduce the number of basis functions in the algorithms, one may consider smaller dimensions for the postdecision state when constructing the value function approximation. Figure 5 shows the results using three value function approximations: (1) all three state variables R_t, E_t, and P_t, (2) R_t and P_t, (3) R_t. It is observed that using R_t as the only domain of the postdecision value function results in quite poor performances for most benchmark problems. Using both R_t and P_t appears to do fairly well overall, although using all of the state variable dimensions yields the best results.

5.3. Benchmark problems with continuous state spaces

In this section, we consider a set of 10 problems with continuous state spaces, continuous actions, and the state transitions for the energy storage optimization problem

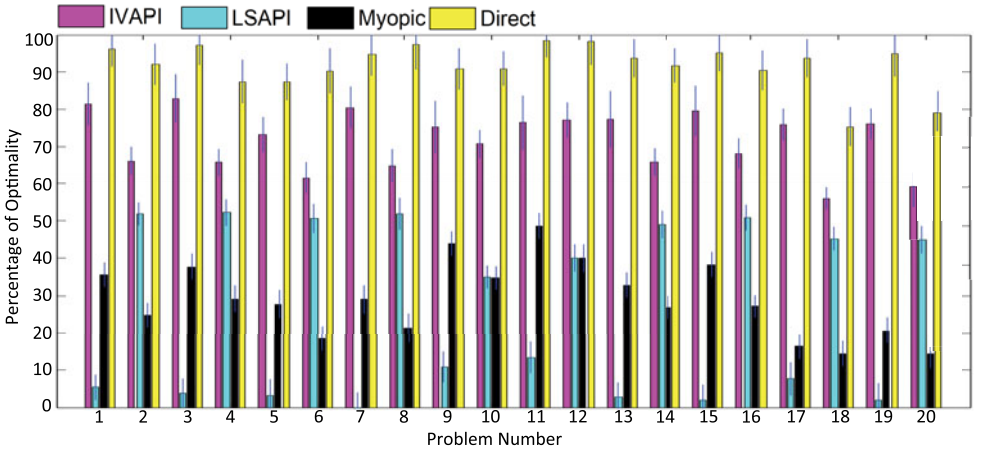

Figure 4. Performance (percentage of optimality) and 95% confidence intervals of different policies for 20 benchmark problems in Table 1. Policies: least-squares API with instrumental variables (IVAPI), least-squares API with least-squares Bellman error minimization (LSAPI), myopic policy (myopic), direct policy search (direct).

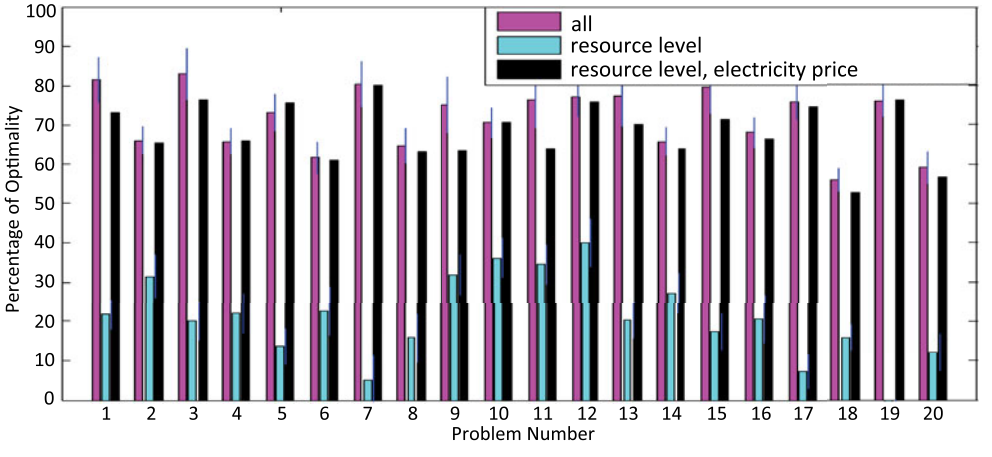

Figure 5. Performance (percentage of optimality) and 95% confidence intervals of the IVAPI policy for the 20 benchmark problems in Table 1, when only certain dimensions of the poststate are included in the poststate value function approximation.

in Section 4. Table 2 summarizes these problems. For Problems 1–3, the electricity prices and demands are time-dependent and stochastic. Problems 4–10 are continuous steady-state problems. For these problems an optimal policy is not available. However, we compare both least-squares API policies and the myopic policy.

Figure 6 illustrates the average objective value for the 10 problems described in Table 2. This figure shows that IVAPI consistently outperforms LSAPI, suggesting that again the use of instrumental variables for the policy evaluation phase of the least-squares API method brings value. Even for some of these problems, the myopic policy outperforms the LSAPI policy.

Table 2. Benchmark problems with continuous states.

Problem		Number of discretization levels					Parameters			
Number	Type	Time τ_t	Resource R_t	Price P_t	Demand D_t	Wind E_t	Wind $\frac{E_t}{D_t}$	Storage $\frac{R_t}{D_t}$	RTE	Charge Rate
1	Full	96	Cont.	Cont.	Cont.	Cont.	0.1	2.5	0.81	C/10
2	Full	96	Cont.	Cont.	Cont.	Cont.	0.1	5.0	0.81	C/10
3	BA	96	Cont.	Cont.	1	1	–	–	0.81	C/10
4	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	5.0	0.81	C/10
5	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	2.5	0.81	C/1
6	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	2.5	0.70	C/1
7	BA	1	Cont.	Cont.	1	1	–	–	0.81	C/10
8	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	5.0	0.81	C/1
9	Full	1	Cont.	Cont.	Cont.	Cont.	0.1	5.0	0.70	C/1
10	Full	1	Cont.	Cont.	Cont.	Cont.	0.2	2.5	0.81	C/1

Problems 1–3 have time-dependent stochastic demands and prices. Problems 4–10 are steady-state.

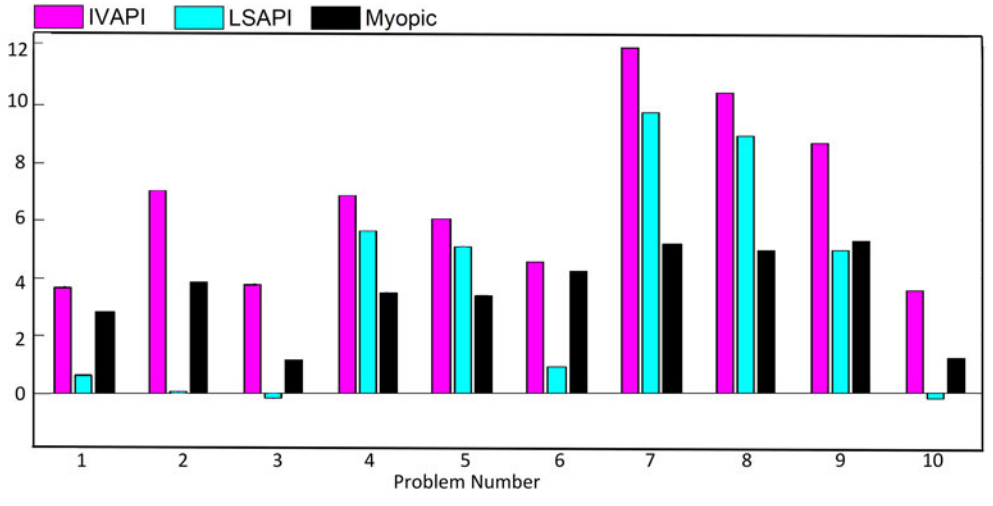

Figure 6. Average objective (in millions) corresponding to the IVAPI, LSAPI, and myopic policies for the benchmark problems with continuous states described in Table 2.

Given that the IVAPI outperforms the alternative least-squares API approach, Figure 7(a) depicts a sample path of the IVAPI policy for Problem 1 in Table 1. The storage device is charged when electricity prices are low and discharged when electricity prices are high. We also note that the storage device fully discharges (below 20%) relatively infrequently. Figure 7(b) illustrates the electricity price and R_t from implementing the IVAPI policy for Problem 5 in Table 2 on one random sample path. Similar to the previous case, this plot shows that the policy tends to start charging the battery at night when electricity prices are low and then discharges the storage device throughout the day when electricity prices are higher.

6. Discussion and limitations

The findings in the previous section suggest that for our benchmark problems and implementation choices, IVAPI is a promising and scalable algorithm. The direct policy search approach, which directly seeks the parameters of a policy function

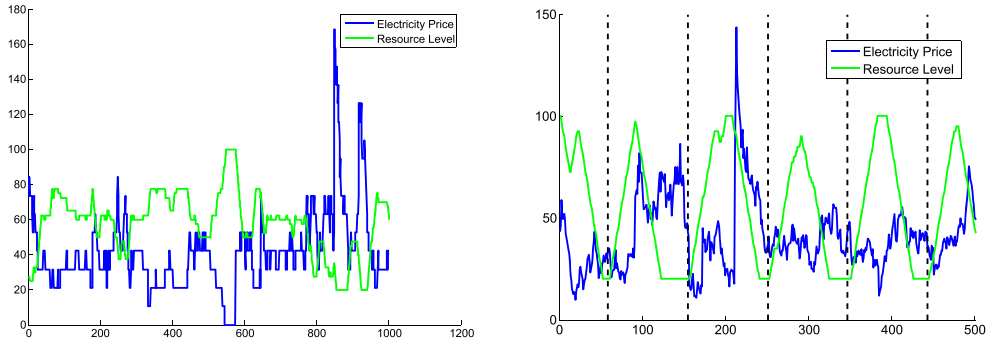

Figure 7. Electricity price and storage level R_t corresponding to the IVAPI policy for one benchmark problem from Table 1 (left plot) and one problem from Table 2 (right plot).

approximation, is capable to come closer to optimality compared to the other ADP algorithms considered in this article. However, direct policy search quickly becomes intractable as the number of parameters in the policy function approximation increases. This challenge arises for example for non-stationary problems or as the number of basis functions in the value function approximation grows. In addition, choosing an appropriate search domain for direct policy search is another significant complication as the number of basis functions increases. A computationally tractable approach for the direct policy search is proposed Moazeni et al. (2017), where a sufficient condition for the optimality of the direct search policy and a bound on the sub-optimality are theoretically established.

There are several limitations associated with our findings. The computational comparison in this article is limited to a subset of approximate dynamic programming methods, namely, two least-squares API methods, direct policy search, and myopic policy. There are alternative dynamic programming approaches based on various varieties of value iteration (Puterman 1994; de Farias and Van Roy 2000; Bertsekas and Tsitsiklis 1996) with approximate value functions, as well as the approximate linear programming approach, made to handle approximations based on basis functions (de Farias and Van Roy 2003, 2004).

Another limitation of our study is that our findings are based on some choices in the implementation of these algorithms. A least-squares API method uses as parameters the number of policy improvements and simulation sample size, M and N. While we chose conservatively large values for these parameters, and used them consistently for all benchmark problems, further sensitivity analysis study could be done to assess the robustness of the findings with respect to these choices. Furthermore, as with any simulation-based approach, the outcomes and standard errors estimated from replications remain sensitive to the underlying stochastic processes and set of drawn samples.

Similarly to most other approximate dynamic programming techniques, the API methods and direct policy search rely on a parametrized value function approximation architecture. As Lagoudakis and Parr (2003) states, "the choice of basis functions is a fundamental problem in itself." The present study was carried out using a linear

architecture with quadratic basis functions, which is a frequent choice in the literature Bradtke and Barto (1996), Lagoudakis and Parr (2003), Wang et al. (2015), but many other options exist for the basis functions. While we conducted sensitivity analysis on the degree and number of basis functions before opting for quadratic basis functions, our comparisons and findings are based on the value function approximation architecture that was adopted.

The energy storage problem can be cast as a type of inventory management model with multiple sources of uncertainty. The findings in this article are limited to benchmark problems developed for the energy storage model. Additional computational studies and comparisons against other optimal benchmark problems could also be insightful and broaden the applicability of the findings.

We fixed the inputs of the algorithms (such as basis functions, discount factor, number of simulations, etc) across the benchmark problems and across the algorithms. Therefore differences in performances observed between these 3 algorithms are merely attributed to the choice of the algorithm, and not the implementation details (such as basis functions).

7. Conclusions

This article studies four variants of LSAPI methods, based on Bellman error minimization policy evaluation. We consider least-squares Bellman error minimization, Bellman error minimization using instrumental variables, least-squares projected Bellman error minimization, and projected Bellman error minimization using instrumental variables. Policy evaluations using Bellman error minimization with instrumental variables is equivalent to projected Bellman error minimization policy evaluations.

The LSAPI methods were then evaluated numerically using a stochastic dynamic optimization problem arising in energy storage control. We create a library of benchmark problems to compare the different algorithmic strategies including least-squares API with two variants of the policy evaluation phase, as well as a Knowledge Gradient based direct policy search method.

Several interesting conclusions can be drawn from our numerical work. Bellman error minimization using instrumental variables appears to improve significantly over the least-squares API method with basic Bellman error minimization, but otherwise did not work well when compared to the optimal benchmark, producing results that ranged between 60% and 80% of optimal. Direct policy search performed much better, with results averaging over 90% of optimal. Given that this is a problem that is ideally suited to least-squares API, it calls into question whether this is a method that can be counted on to produce good results.

This research suggests that there are clear advantages to using direct policy search, possibly in conjunction with approximate policy iteration. We suggest using least-squares API to find good values of the regression parameters, and then apply direct policy search to improve the policy in the region of the fitted regression parameters. For certain problems, it may actually be advantageous to leave variables out of the value function approximation to simplify the policy search process. The challenge is

that in its derivative-free form, policy search does not scale easily with the dimension of the parameter space. 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.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Somayeh Moazeni http://orcid.org/0000-0003-3631-563X

References

Antos A, Szepesvari C, Munos R. 2008. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. Mach Learn. 71(1): 89–129.

Barton JP, Infield DG. 2004. Energy storage and its use with intermittent renewable energy. IEEE Trans Energy Convers. 19(2):441–448.

Bertsekas DP. 2012. Dynamic programming and optimal control, Vol. II, 4th ed. Approximate dynamic programming. Belmont (MA): Athena Scientific.

Bertsekas DP, Tsitsiklis JN. 1996. Neuro-dynamic programming. 1st ed. Belmont (MA): Athena Scientific.

Bowden R, Turkington D. 1984. Instrumental variables. Cambridge (UK): Cambridge University Press.

Bradtke S, Barto A. 1996. Linear least-squares algorithms for temporal difference learning. Mach Learn. 22(1–3):33–57.

Brown BG, Katz RW, Murphy AH. 1984. Time series models to simulate and forecast wind speed and power. J Climate Appl Meteorol. 23(8):1184–1195.

Brunet Y. 2011. Energy storage. Hoboken, NJ: Wiley.

Busoniu L, Lazaric A, Ghavamzadeh M, Munos R, Babuska R, Schutter BD. 2012. Least-squares methods for policy iteration. In: Wiering M, van Otterlo M, editors. Reinforcement learning: state-of-the-art. Berlin, Heidelberg: Springer Berlin Heidelberg; p. 75–109.

Cameron AC, Trivedi PK. 2005. Microeconometrics: methods and applications. Cambridge (UK): Cambridge University Press.

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

Carmona R, Ludkovski M. 2010. Valuation of Energy Storage: An Optimal Switching Approach. *Quantitative Finance.* 10(4):359–374.

Cartea A, Figueroa M. 2005. Pricing in electricity markets: a mean reverting jump diffusion model with seasonality. Appl Math Fin. 12(4):313–335.

Dann C, Neumann G, Peters J. 2014. Policy evaluation with temporal differences: A survey and comparison. J Mach Learn Res. 15:809–883.

De Farias D, Van Roy B. 2000. On the existence of fixed points for approximate value iteration and temporal-difference learning. J Optim Theory Appl. 105(3):589–608.

de Farias D, Van Roy B. 2003. The linear programming approach to approximate dynamic programming. Oper Res. 51(6):850–865.

de Farias D, Van Roy B. 2004. On constraint sampling in the linear programming approach to approximate dynamic programming. Math Oper Res. 29(3):462–478.

de Farias DP, Van Roy B. 2000. On the existence of fixed points for approximate value iteration and temporal-difference learning. J Optim Theor Appl. 105:1201–1225.
Durbin J. 1954. Errors in variables. Rev L'Inst Int Stat. 22(1/3):23–32.
Eydeland A, Wolyniec K. 2003. Energy and power risk management: New developments in modeling, pricing, and hedging. Hoboken (NJ): John Wiley & Sons Inc.
Feinberg E, Genethliou D. 2010. Load forecasting. In: Chow JH, Wu FF, Momoh JA, editors. Applied mathematics for restructured electric power systems: optimization, control, and computational intelligence. Berlin, Heidelberg: Springer; p. 269–285.
Gonzalez JG, de la Muela RMR, Santos LM, Gonzalez AM. 2008. Stochastic joint optimization of wind generation and pumped-storage units in an electricity market. IEEE Trans Power Syst. 23(2):460–468.
Greenblatt J, Succar S, Denkenberger D, Williams R, Socolow R. 2007. Baseload wind energy: modeling the competition between gas turbines and compressed air energy storage for supplemental generation. Energy Policy 35(3):1474–1492.
Halman N, Nannicini G, Orlin J. 2018. On the complexity of energy storage problems. Discrete Optim. 28:31–53.
He M, Zhao L, Powell WB. 2012. Approximate dynamic programming algorithms for optimal dosage decisions in controlled ovarian hyper stimulation. Eur J Oper Res. 222(2):328–340.
Heuberger PSC, den Hov P, Wahlberg B. 2005. Modeling and identification with rational orthogonal basis functions. New York (NY): Springer.
Hu Y, Defourny B. 2017. Optimal price-threshold control for battery operation with aging phenomenon: a quasiconvex optimization approach. Ann Oper Res. 1–28. doi:10.1007/s10479-017-2505-4. Available from: https://doi.org/10.1007/s10479-017-2505-4.
Judd K. 1998. Numerical methods in economics. Cambridge (MA): MIT Press.
Kendall M, Stuart A. 1961. The advanced theory of statistics: Inference and relationship. Vol. 2. New York: Hafner Publishing Company.
Kolter JZ. 2011. The fixed points of off-policy TD. Presented at the Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS); p. 2169–2177.
Konidaris G, Osentoski S, Thomas P. 2011. Value function approximation in reinforcement learning using the Fourier basis. In: Proceedings of the Twenty-Fifth Conference on Artificial Intelligence, August; p. 380–385. Available from: http://lis.csail.mit.edu/pubs/konidaris-aaai11a.pdf.
Lagoudakis M, Parr R. 2003. Least-squares policy iteration. J Mach Learn Res. 4(6):1107–1149.
Lai G, Margot F, Secomandi N. 2010. An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. Oper Res. 58(3):564–582.
MacKay DJ. 2009. Sustainable energy-without the hot air. 1st ed. Cambridge (UK): UIT Cambridge Ltd.
Maei HH, Szepesvari C, Bhatnagar S, Silver D, Precup D, Sutton RS. 2009. Convergent temporal-difference learning with arbitrary smooth function approximation. In: Advances in neural information processing systems. Vol. 22. Vancouver, BC: MIT Press. p. 1–9.
Menache I, Mannor S, Shimkin N. 2005. Basis function adaptation in temporal difference reinforcement learning. Ann Oper Res. 134(1):215–238.
Moazeni S, Coleman TF, Li Y. 2016. Smoothing and parametric rules for stochastic mean-CVaR optimal execution strategy. Ann Oper Res. 237(1–2):99–120.
Moazeni S, Defourny B. 2018. Optimal control of energy storage under random operation permissions. IISE Trans. 50(8):668–682.
Moazeni S, Powell WB, Defourny B, Bouzaiene-Ayari B. 2017. Parallel nonstationary direct policy search for risk-averse stochastic optimization. INFORMS J Comput. 29(2):332–349.
Moazeni S, Powell WB, Hajimiragha A. 2015. Mean-conditional value-at-risk optimal energy storage operation in the presence of transaction costs. IEEE Trans Power Syst. 30(3):1222–1232.
Nadarajah S, Margot F, Secomandi N. 2015. Relaxations of approximate linear programs for the real option management of commodity storage. Manage Sci. 61(12):3054–3076.

Nascimento JM, Powell WB. 2013. An optimal approximate dynamic programming algorithm for concave, scalar storage problems with vector-valued controls. IEEE Trans Autom Contr. 58:2995–3010.
Pirrong C, Jermakyan M. 2008. The price of power: The valuation of power and weather derivatives. J Bank Fin. 32:2520–2529.
Porteus E. 2002. Foundations of stochastic inventory theory. 1st ed. Redwood City (CA): Stanford Business Books.
Powell WB. 2011. Approximate dynamic programming: Solving the curses of dimensionality. 2nd ed. New York (NY): Wiley.
Powell WB, Ma J. 2011. A review of stochastic algorithms with continuous value function approximation and some new approximate policy iteration algorithms for multidimensional continuous applications. J Control Theory Appl. 9(3):336–352.
Powell WB, Ryzhov IO. 2012. Optimal learning. Hoboken (NJ): Wiley.
Puterman ML. 1994. Markov decision processes. New York: Wiley.
Scott WR, Frazier P, Powell WB. 2011. The correlated knowledge gradient for simulation optimization of continuous parameters using Gaussian process regression. SIAM J Optim. 21(3): 996–1026.
Simao HP, Day J, George A, Gifford T, Powell WB, Nienow J. 2009. An approximate dynamic programming algorithm for large-scale fleet management: a case application. Transport Sci. 43(2):178–197.
Söderström T, Stoica P. 1983. Instrumental variable methods for system identification. Vol. 161. Berlin, Germany: Springer.
Sutton R, Barto A. 1998. Reinforcement learning. Cambridge (MA): The MIT Press.
Sutton RS, Maei HH, Precup D, Bhatnagar S, Silver D, Szepesvari C, Wiewiora E. 2009a. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In: Proceedings of the 26th Annual International Conference on Machine Learning – ICML'09, New York (NY): ACM Press; p. 993–1000.
Sutton RS, Szepesvari C, Maei HH. 2009b. A convergent o (n) algorithm for off-policy temporal-difference learning with linear function approximation. In: Advances in neural information processing systems. Vol. 21. Princeton (NJ): Citeseer; p. 1609–1616.
Swider D. 2007. Compressed air energy storage in an electricity system with significant wind power generation. IEEE Trans Energy Convers. 22(1):95.
Szepesvari C. 2010. Algorithms for reinforcement learning (synthesis lectures on artificial intelligence and machine learning). Williston (VT): Morgan and Claypool.
Topaloglu H, Powell WB. 2006. Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems. INFORMS J Comput. 18(1):31–42.
Tsitsiklis J, Van Roy B. 1997. An analysis of temporal-difference learning with function approximation. IEEE Trans Automativ Control 42(5):674–690.
Tsitsiklis JN. 1994. Asynchronous stochastic approximation and Q-learning. Mach Learn. 16(3):185–202.
Wang Y, O'Donoghue B, Boyd S. 2015. Approximate dynamic programming via iterated Bellman inequalities. Int J Robust Nonlinear Control 25(10):1472–1496.
Young PC. 2011. Recursive estimation and time-series analysis. 2nd ed. Berlin, Germany: Springer-Verlag.
Yu H. 2012. Least squares temporal difference methods: an analysis under general condition. SIAM J Control Optim. 50(6):3310–3343.

Appendix A: The instrumental variable method

The instrumental variable method (IVM) is a well-known technique for dealing with errors in the explanatory variables of a regression problem, and provides a way to obtain consistent parameter estimates, see, e.g. Cameron and Trivedi (2005), Young (2011). Consider the linear model in the matrix form $Y = X\theta$, where Y is a $N \times 1$ vector of response variables, X is a

$N \times K$ matrix of explanatory variables, and θ is a $K \times 1$ vector of weights. Let X' and Y' be observable values of the true values X and Y. Denote the errors in the observed values of X and Y by X'' and Y'', respectively. Hence we have $X' = X + X''$ and $Y' = Y + Y''$. When the observation of X and the error in X are correlated, the least squares estimator can be biased and inconsistent, see, e.g. Chapter 4 of Cameron and Trivedi (2005). A properly chosen instrumental variable can yield a consistent estimator for θ. Suppose that an instrumental variable, Z_j, exists such that it is correlated with the true X_l, the l^{th} column of X, but uncorrelated with the errors in the observations of X and Y.

Denote $\Sigma_{jl} := \text{Cov}[Z_j, X_l]$, for $j, l = 1, \ldots, K$. Assume that the matrix Σ has full rank K. For the instrument Z, the instrumental variables (IV) estimator is defined as

$$\hat{\theta}_{IV} = (Z^\top X')^{-1} Z^\top Y'. \tag{A1}$$

Note that $\hat{\theta}_{IV}$ is uniquely defined when $Z^\top X'$ has full rank K.

Proposition A.1. *Consider the model $Y = X\theta$ with observable values X' and Y', and error terms X'' and Y''. Suppose that the noise in X and Y satisfy $\mathbb{E}[Y''] = 0$, and $\mathbb{E}[X''_{ij}] = 0$, for every $i = 1, \ldots, N$ and $j = 1, \ldots, K$. Suppose that $\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} Z_{ij} X_{il} = \Sigma_{jl}$, for $j, l = 1, \ldots, K$, and $\text{Cov}[Z_{ij}, Y''_i] = \text{Cov}[Z_{ij}, X''_{il}] = 0$, for every $i = 1, \ldots, N$, and $j = 1, \ldots, K$. In addition assume that $\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} Z_{ij} Y''_i = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} Z_{ij} X''_{il} = 0$, for every $j = 1, \ldots, K$. Then the IV estimator $\hat{\theta}_{IV}$ is a consistent estimator of θ, i.e., $\hat{\theta}_{IV} \to \theta$ with probability one, as $N \to \infty$.*

For a proof of Proposition A.1 and further discussion on IV estimators, see Cameron and Trivedi (2005).

Appendix B: Consistency of IVPBEM policy evaluation

This appendix shows that $\hat{\theta}_{\text{IVPBEM}}$ in Equation (15) is consistent (converges in probability to the true weights). The following discussion remains valid even when the state space is continuous or the discount factor is $\gamma = 1$. More formally, we aim to show that $\hat{\theta}_{\text{IVPBEM}}$ is a consistent estimator for projected Bellman equation. Using the notations in Appendix A, define

$$X \overset{\text{def}}{=} \Pi_{t-1} \left(\Phi_{t-1} - \gamma \mathbb{E}\left[\gamma \Phi_t \middle| \left\{ S^x_{t-1} \right\} \right] \right), \tag{B1}$$

$$X' \overset{\text{def}}{=} \Pi_{t-1} (\Phi_{t-1} - \gamma \Phi_t), \tag{B2}$$

$$Y \overset{\text{def}}{=} \Pi_{t-1} \bar{C}_t, \tag{B3}$$

$$Y' \overset{\text{def}}{=} \Pi_{t-1} C_t, \tag{B4}$$

where $\bar{C}_t$ is the vector with entries $\bar{C}_{t,n} \overset{\text{def}}{=} \mathbb{E}[C(S_{t,n}, X^{\hat{\pi}}(S_{t,n}|\theta))|S^x_{t-1,n}]$. Recall that Φ_{t-1}, Φ_t, and C_t are as in Equation (7) corresponding to the policy $\hat{\pi}$.

Theorem B.1. *Let assumption 2.1 hold and the covariance matrix Σ, with elements $\Sigma_{jl} = \text{Cov}[(\Phi_{t-1})_j, X_l]$, have full rank K. Suppose that the rows of Φ_{t-1} are i.i.d., $\mathbb{E}[|(\Phi_{t-1})_{ij}(Y' - Y)_i|] < \infty$, and $\mathbb{E}[|(\Phi_{t-1})_{ij}(X' - X)_{il}|] < \infty$, for $j, l = 1, \ldots, K$. Then the estimator $\hat{\theta}_{\text{IVPBEM}}$ in equation (15) is a consistent estimator.*

Proof. *Define the instrumental variable $Z \overset{\text{def}}{=} \Phi_{t-1}$. It thus follows the notations X' and Y' in equations (B2) and (B4) that $\hat{\theta}_{\text{IVPBEM}}$ in equation (15) has the form $(Z^\top X\prime)^{-1} Z^\top Y'$. According to Proposition A.1 in Appendix A the IV estimators are consistent estimators of θ for the model*

$Y = X\theta$. Therefore, to complete the proof of Theorem B.1, it is sufficient to show that the assumptions for Proposition A.1 hold.

From the description of Y and Y' in (B3) and (B4), we have $Y'' = Y' - Y = \Pi_{t-1}(C_t - \bar{C}_t)$. Therefore,

$$\begin{aligned}\mathbb{E}[Y''] &= \mathbb{E}[\Pi_{t-1}(C_t - \bar{C}_t)] \\ &= \mathbb{E}\left[\mathbb{E}\left[\Pi_{t-1}(C_t - \bar{C}_t)|\{S^x_{t-1}\}\right]\right] \\ &= \mathbb{E}\left[\Pi_{t-1}\underbrace{\mathbb{E}\left[(C_t - \bar{C}_t)|\{S^x_{t-1}\}\right]}_{=0}\right] = 0.\end{aligned} \tag{B5}$$

Next, it follows from Equations (B1) and (B2) that the mean of the noise in the observation of the explanatory variables, $X'' = X' - X$, equals zero:

$$\begin{aligned}\mathbb{E}[X''] &= \mathbb{E}[X' - X] \\ &= \mathbb{E}\left[\Pi_{t-1}(\Phi_{t-1} - \gamma\Phi_t) - \Pi_{t-1}\left(\Phi_{t-1} - \mathbb{E}\left[\gamma\Phi_t|\{S^x_{t-1}\}\right]\right)\right] \\ &= \gamma\mathbb{E}\left[\Pi_{t-1}\left(\mathbb{E}\left[\Phi_t|\{S^x_{t-1}\}\right] - \Phi_t\right)\right].\end{aligned}$$

Therefore, using $\mathbb{E}[\Pi_{t-1}(\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \Phi_t)] = \mathbb{E}[\mathbb{E}[\Pi_{t-1}(\mathbb{E}[\Phi_t|\{S^x_{t-1}\}] - \Phi_t)|\{S^x_{t-1}\}]]$, we have

$$\begin{aligned}\mathbb{E}[X''] &= \gamma\mathbb{E}\left[\Pi_{t-1}\mathbb{E}\left[\mathbb{E}\left[\Phi_t|\{S^x_{t-1}\}\right] - \Phi_t|\{S^x_{t-1}\}\right]\right] \\ &= \gamma\mathbb{E}\left[\Pi_{t-1}\left(\underbrace{\mathbb{E}\left[\Phi_t|\{S^x_{t-1}\}\right] - \mathbb{E}\left[\Phi_t|\{S^x_{t-1}\}\right]}_{=0}\right)\right] = 0.\end{aligned} \tag{B6}$$

We next show that $\mathrm{Cov}[Z_{ij}, Y''_i] = 0$, for every i, j:

$$\begin{aligned}\mathrm{Cov}\left[Z_{ij}, Y''_i\right] &= \mathbb{E}\left[Z_{ij}Y''_i\right] - \mathbb{E}\left[Z_{ij}\right]\underbrace{\mathbb{E}\left[Y''_i\right]}_{=0} \\ &= \mathbb{E}\left[(\Phi_{t-1})_{ij}(\Pi_{t-1}(C_t - \bar{C}_t))_i\right] \\ &= \mathbb{E}\left[(\Phi_{t-1})_{ij}e_i^\top\ \Pi_{t-1}(C_t - \bar{C}_t)\right],\end{aligned}$$

where e_i denotes the column vector of all zeros except at the i^{th} element which equals 1. Therefore,

$$\begin{aligned}\mathrm{Cov}\left[Z_{ij}, Y''_i\right] &= \mathbb{E}\left[\mathbb{E}\left[(\Phi_{t-1})_{ij}e_i^\top\ \Pi_{t-1}(C_t - \bar{C}_t)|\{S^x_{t-1}\}\right]\right] \\ &= \mathbb{E}\left[(\Phi_{t-1})_{ij}e_i^\top\ \Pi_{t-1}\underbrace{\mathbb{E}\left[C_t - \bar{C}_t|\{S^x_{t-1}\}\right]}_{=0}\right] = 0.\end{aligned} \tag{B7}$$

Next we show that for every i, j, and l, $\mathrm{Cov}[Z_{ij}, X''_{il}] = 0$:

$$\mathrm{Cov}\left[Z_{ij}, X''_{il}\right] = \mathbb{E}\left[Z_{ij}X''_{il}\right] - \mathbb{E}\left[Z_{ij}\right]\underbrace{\mathbb{E}\left[X''_{il}\right]}_{=0} = \mathbb{E}\left[Z_{ij}\left(X'_{il} - X_{il}\right)\right] = \mathbb{E}\left[Z_{ij}e_i^\top(X' - X)e_l\right].$$

Hence,

$$\begin{aligned}\mathrm{Cov}\left[Z_{ij}, X''_{il}\right] &= \gamma\mathbb{E}\left[(\Phi_{t-1})_{ij}e_i^\top \Pi_{t-1}\left(\mathbb{E}\left[\Phi_t|\left\{S^x_{t-1}\right\}\right]-\Phi_t\right)e_l\right]\\ &= \gamma\mathbb{E}\left[\mathbb{E}\left[(\Phi_{t-1})_{ij}e_i^\top \Pi_{t-1}\left(\mathbb{E}\left[\Phi_t|\left\{S^x_{t-1}\right\}\right]-\Phi_t\right)e_l|\left\{S^x_{t-1}\right\}\right]\right]\\ &= \gamma\mathbb{E}\left[(\Phi_{t-1})_{ij}e_i^\top \Pi_{t-1}\mathbb{E}\left[\mathbb{E}\left[\Phi_t|\left\{S^x_{t-1}\right\}\right]-\Phi_t|\left\{S^x_{t-1}\right\}\right]e_l\right]\\ &= \gamma\mathbb{E}\left[(\Phi_{t-1})_{ij}e_i^\top \Pi_{t-1}\underbrace{\left(\mathbb{E}\left[\Phi_t|\left\{S^x_{t-1}\right\}\right]-\mathbb{E}\left[\Phi_t|\left\{S^x_{t-1}\right\}\right]\right)}_{=0}e_l\right] = 0.\end{aligned} \tag{B8}$$

The assumptions stated in the theorem and the law of large numbers imply that $\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^N Z_{ij}X_{il} = \mathrm{Cov}[Z_j, X_l]$, *and for every* $j = 1, \ldots, K, \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^N Z_{ij}Y''_i = 0$ *and* $\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^N Z_{ij}X''_{il} = 0$. *Thus, Proposition A.1 can be applied.*

Appendix C: Proof of Proposition 2.1

Proof: We first show that IVBEM and IVPBEM estimators in equations (11) and (15) are equal. Recall that $\Pi_{t-1} = \Phi_{t-1}(\Phi_{t-1}^\top\Phi_{t-1})^{-1}\Phi_{t-1}^\top$. Starting with equation (15), we have

$$\begin{aligned}\hat\theta_{\mathrm{IVPBEM}} &= \left((\Phi_{t-1})^\top\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)\right)^{-1}(\Phi_{t-1})^\top\Pi_{t-1}C_t\\ &= \left((\Phi_{t-1})^\top\left(\underbrace{\Pi_{t-1}\Phi_{t-1}}_{\Phi_{t-1}}-\gamma\Pi_{t-1}\Phi_t\right)\right)^{-1}\underbrace{(\Phi_{t-1})^\top\Phi_{t-1}\left((\Phi_{t-1})^\top\Phi_{t-1}\right)^{-1}}_{I_{K\times K}}\Phi_{t-1}^\top C_t\\ &= \left[\Phi_{t-1}^\top\Phi_{t-1}-\gamma\underbrace{(\Phi_{t-1})^\top\Phi_{t-1}\left((\Phi_{t-1})^\top\Phi_{t-1}\right)^{-1}}_{I_{K\times K}}(\Phi_{t-1})^\top\Phi_t\right]^{-1}\Phi_{t-1}^\top C_t\\ &= \left(\Phi_{t-1}^\top(\Phi_{t-1}-\gamma\Phi_t)\right)^{-1}\Phi_{t-1}^\top C_t = \hat\theta_{\mathrm{IVBEM}}.\end{aligned}$$

Next, we show Equations (11) and (14) are equivalent. We have

$$\begin{aligned}\hat\theta_{\mathrm{LSPBEM}} &= \left((\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))^\top(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))\right)^{-1}(\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t))^\top\Pi_{t-1}C_t\\ &= \left((\Phi_{t-1}-\gamma\Phi_t)^\top\Pi_{t-1}^\top\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)\right)^{-1}(\Phi_{t-1}-\gamma\Phi_t)^\top(\Pi_{t-1})^\top\Pi_{t-1}C_t.\end{aligned}$$

It follows from $\Pi_{t-1} = \Phi_{t-1}(\Phi_{t-1}^\top\Phi_{t-1})^{-1}\Phi_{t-1}^\top$ that $(\Pi_{t-1})^\top\Pi_{t-1} = \Pi_{t-1} = \Pi_{t-1}^\top$. Hence, $\hat\theta_{\mathrm{LSPBEM}}$ equals

$$\begin{aligned}&\left((\Phi_{t-1}-\gamma\Phi_t)^\top\Pi_{t-1}(\Phi_{t-1}-\gamma\Phi_t)\right)^{-1}(\Phi_{t-1}-\gamma\Phi_t)^\top\Pi_{t-1}C_t\\ &= \left((\Phi_{t-1}-\gamma\Phi_t)^\top\Phi_{t-1}(\Phi_{t-1}^\top\Phi_{t-1})^{-1}\Phi_{t-1}^\top(\Phi_{t-1}-\gamma\Phi_t)\right)^{-1}(\Phi_{t-1}-\gamma\Phi_t)^\top\Phi_{t-1}(\Phi_{t-1}^\top\Phi_{t-1})^{-1}\Phi_{t-1}^\top C_t\\ &= \left(\Phi_{t-1}^\top(\Phi_{t-1}-\gamma\Phi_t)\right)^{-1}\left((\Phi_{t-1}^\top\Phi_{t-1})^{-1}\right)^{-1}\left((\Phi_{t-1}-\gamma\Phi_t)^\top\Phi_{t-1}\right)^{-1}(\Phi_{t-1}-\gamma\Phi_t)^\top\Phi_{t-1}(\Phi_{t-1}^\top\Phi_{t-1})^{-1}\Phi_{t-1}^\top C_t\\ &= \left(\Phi_{t-1}^\top(\Phi_{t-1}-\gamma\Phi_t)\right)^{-1}(\Phi_{t-1}^\top\Phi_{t-1})\underbrace{\left((\Phi_{t-1}-\gamma\Phi_t)^\top\Phi_{t-1}\right)^{-1}(\Phi_{t-1}-\gamma\Phi_t)^\top\Phi_{t-1}}_{I_K}(\Phi_{t-1}^\top\Phi_{t-1})^{-1}\Phi_{t-1}^\top C_t\\ &= \left(\Phi_{t-1}^\top(\Phi_{t-1}-\gamma\Phi_t)\right)^{-1}\underbrace{(\Phi_{t-1}^\top\Phi_{t-1})(\Phi_{t-1}^\top\Phi_{t-1})^{-1}}_{I_K}\Phi_{t-1}^\top C_t = \hat\theta_{\mathrm{IVBEM}}.\end{aligned}$$

This completes the proof of $\hat\theta_{\mathrm{LSPBEM}} = \hat\theta_{\mathrm{IVBEM}}$. □