INSTRUCTIONS FOR STORAGE BENCHMARK PROBLEMS

DANIEL R. JIANG* AND WARREN B. POWELL*

Abstract. This document gives instructions for using the MATLAB code for the energy storage optimal benchmarks. As provided, the code allows the user to simulate the optimal allocation strategy on randomly generated data. With minimal additional specification, the user is able to simulate an arbitrary policy (i.e., heuristic or approximately optimal strategies) and compare its performance against optimal. Furthermore, the code provides the framework necessary to train an *approximate dynamic programming* (ADP) or *reinforcement learning* (RL) algorithm. The problem description and parameters are excerpted from the paper Jiang and Powell [2015].

1. Problem Description

The recent surge of interest in renewable energy leads us to present a second example application in area energy storage. The goal of this specific problem is to optimize revenues while satisfying demand, in the presence of 1) a storage device, such as a battery, and 2) a (stochastic) renewable source of energy, such as wind or solar. Our action or decision vector is an *allocation decision*, containing five dimensions, that describe how the energy is transferred within our network, consisting of nodes for the storage device, the spot market, demand, and the source of renewable generation (see Figure 1). Similar problems from the literature that share the common theme of storage include Secomandi [2010], Carmona and Ludkovski [2010], and Kim and Powell [2011], to name a few.

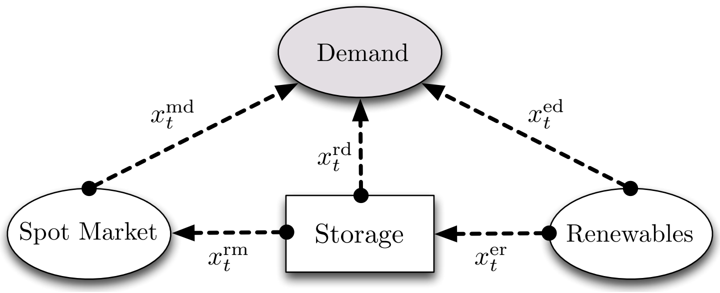

Figure 1. Network Diagram for the Energy Storage Problem

Let the state variable $S_t = (R_t, E_t, P_t, D_t) \in \mathcal{S}$, where R_t is the amount of energy in storage at time t, E_t is the amount of renewable generation available at time t, P_t is the price of energy on the spot market at time t, and D_t is the demand that needs to be satisfied at time t. We define the discretized state space $\mathcal{S}$ by allowing (R_t, E_t, P_t, D_t) to take on all integral values contained within the hyper–rectangle

$$[0, R_{\max}] \times [E_{\min}, E_{\max}] \times [P_{\min}, P_{\max}] \times [D_{\min}, D_{\max}],$$

where $R_{\max} \geq 0$, $E_{\max} \geq E_{\min} \geq 0$, $P_{\max} \geq P_{\min} \geq 0$, and $D_{\max} \geq D_{\min} \geq 0$. Let γ_c and γ_d be the maximum rates of charge and discharge from the storage device, respectively. The decision vector is given by (refer again to Figure 1 for the meanings of the components)

$$x_t = (x_t^{\text{ed}}, x_t^{\text{md}}, x_t^{\text{rd}}, x_t^{\text{er}}, x_t^{\text{rm}})^\mathsf{T} \in \mathcal{X}(S_t),$$

*Department of Operations Research and Financial Engineering, Princeton University where the feasible set $\mathcal{X}(S_t)$ is defined by $x_t \in \mathbb{N}^5$ intersected with the following, mostly intuitive, constraints:

$$\begin{aligned} (x_t^{\text{ed}}, x_t^{\text{md}}, x_t^{\text{rd}}, x_t^{\text{er}}, x_t^{\text{rm}})^\intercal &\geq 0, \\ x_t^{\text{ed}} + x_t^{\text{rd}} + x_t^{\text{md}} &= D_t, \\ x_t^{\text{rd}} + x_t^{\text{rm}} &\leq R_t, \\ x_t^{\text{er}} + x_t^{\text{ed}} &\leq E_t, \\ x_t^{\text{rd}} + x_t^{\text{rm}} &\leq \gamma^d, \\ x_t^{\text{er}} &\leq \min\{R_{\max} - R_t, \gamma_c\}. \end{aligned}$$

Note that whenever the energy in storage combined with the amount of renewable generation is not enough to satisfy demand, the remainder is purchased from the spot market. The contribution function is given by

$$C_t(S_t, x_t) = P_t \left(D_t + x_t^{\text{rm}} - x_t^{\text{md}}\right).$$

To describe the transition of the state variable, first define $\phi = (0, 0, -1, 1, -1)^\intercal$ to be the flow coefficients for a decision x_t with respect to the storage device. We also assume that the dependence on the past for the stochastic processes $\{E_t\}_{t=0}^T$, $\{P_t\}_{t=0}^T$, and $\{D_t\}_{t=0}^T$ is at most Markov of order 1. Thus, we can write the transition function for S_t using the following set of equations:

$$R_{t+1} = R_t + \phi^\intercal x_t \quad \text{and} \quad \begin{aligned} E_{t+1} &= \min\bigl\{\max\{E_t + \hat{E}_{t+1}, E_{\min}\}, E_{\max}\bigr\}, \\ P_{t+1} &= \min\bigl\{\max\{P_t + \hat{P}_{t+1}, P_{\min}\}, P_{\max}\bigr\}, \\ D_{t+1} &= \min\bigl\{\max\{D_t + \hat{D}_{t+1}, D_{\min}\}, D_{\max}\bigr\}, \end{aligned} \tag{1.1}$$

where the exogenous information $W_{t+1} = (\hat{E}_{t+1}, \hat{P}_{t+1}, \hat{D}_{t+1})$ is independent of S_t and x_t (the precise processes used are given in Section 1.1).

1.1. **Parameter Choices.**

In our experiments, a continuous distribution D with density f_D is discretized over a set $\mathcal{X}_D$ by assigning each outcome $x \in \mathcal{X}_D$ the probability $f_D(x)/\sum_{x' \in \mathcal{X}_D} f(x')$. We consider two instances of the energy storage problem for $T = 25$: the first is labeled S1 and has a smaller storage device and relatively low variance in the change in renewable supply $\hat{E}_{t+1}$, while the second, labeled S2, uses a larger storage device and has relatively higher variance in $\hat{E}_{t+1}$. We take $R_{\min} = 0$ with $R_{\max} = 30$ for S1 and $R_{\max} = 50$ for S2, and we set $\gamma_c = \gamma_d = 5$ for S1 and S2. The stochastic renewable supply has characteristics given by $E_{\min} = 1$ and $E_{\max} = 7$, with $\hat{E}_{t+1}$ being i.i.d. discrete uniform random variables over $\{0, \pm 1\}$ for S1 and $\hat{E}_{t+1}$ being i.i.d. $\mathcal{N}(0, 3^2)$ discretized over the set $\{0, \pm 1, \pm 2, \ldots, \pm 5\}$. For both cases, we have $P_{\min} = 30$ and $P_{\max} = 70$ with $\hat{P}_{t+1} = \epsilon_{t+1}^P + \mathbf{1}_{\{U_{t+1} < p\}}\, \epsilon_{t+1}^J$, in order to simulate price spikes (or jumps). ϵ_{t+1}^E is $\mathcal{N}(0, 2.5^2)$ discretized over $\{0, \pm 1, \pm 2, \ldots, \pm 8\}$; ϵ_{t+1}^J is $\mathcal{N}(0, 50^2)$ discretized over $\{0, \pm 1, \pm 2, \ldots, \pm 40\}$; and U_{t+1} is $\mathcal{U}(0, 1)$ with $p = 0.031$. Lastly, for the demand process, we take $D_{\min} = 0$, $D_{\max} = 7$, and $D_t + \hat{D}_{t+1} = \bigl\lfloor 3 - 4\,\sin(2\pi\,(t+1)/T)\bigr\rfloor + \epsilon_{t+1}^D$, where ϵ_{t+1}^D is $\mathcal{N}(0, 2^2)$ discretized over $\{0, \pm 1, \pm 2\}$, in order to roughly model the seasonality that often exists in observed energy demand. For both problems, we use an initial state of an empty storage device and the other dimensions of the state variable set to their minimum values: $S_0 = (R_{\min}, E_{\min}, P_{\min}, D_{\min})$. Table 1 summarizes the sizes of these two problems. Since the size of the action space is not constant over the state space, we report the average, i.e., $|\mathcal{S}|^{-1} \sum_s |\mathcal{X}(s)|$. The maximum size of the feasible set over the state space is also given (and is the same for both S1 and S2). Finally, we remark that the greater than linear increase in computation time for S2 compared to S1 is due to the larger action space and the larger number of exogenous outcomes that need to be evaluated.

Label	State Space	Eff. State Space	Action Space	CPU (Sec.)	Optimal Value
S1	71,176	1,850,576	165, Max: 623	41,675	3,745.58
S2	117,096	3,044,496	178, Max: 623	115,822	4,872.54

TABLE 1. Basic Properties of Energy Storage Problem Instances

2. MATLAB Code Instructions

The two benchmarked instances `S1` and `S2` reside in subfolders, while the core files are in the main folder. The following is a description of the included files.

- **`contribution.m`**: the contribution function takes a state, action, time, and parameters and returns the contribution. This file is referenced through an anonymous function in `parameters.m`.
- **`feasible_set.m`**: returns the feasible set of actions given the current state, time, and parameters. The output is a matrix where each row is a feasible multidimensional decision. This file is referenced through an anonymous function in `parameters.m`.
- **`transition.m`**: returns the next state given the current state, action, time, exogenous information (i.e., E_t, P_t, and D_t), and parameters. This file is referenced through an anonymous function in `parameters.m`.
- **`possible_next_exo.m`**: returns the possible outcomes of the exogenous information and the associated probabilities in the next time period given the current state, action, time, and parameters. Each row of `W_next` is an outcome and the corresponding element in `W_probs` is the probability of that outcome.
- **`simulate_policy.m`**: the simulator that takes as input any policy and a set of parameters. Notice the two ways of generating the next state (the if statement), using either a *transition function* along with possible outcomes of the *exogenous information* or directly attaching the probabilities to possible outcomes of the next state.
- **`optimal_policy.m`**: returns the index (or row) of the optimal action from the matrix of feasible actions (i.e., the output of `feasible_set.m`), given an array `A` that maps from state to action index.
- **`Sx/bdp_A.mat`**: a data matrix that stores the optimal policy (which we have precomputed) in a tabular way (i.e., maps states to actions). There is no need to edit this file.
- **`Sx/parameters.m`**: a script file that loads the problem parameters for the instance `Sx`, as described above. The file is commented and mostly self explanatory; however, note the following:
  - `params.use_next_exo` specifies whether or not to use a *transition function* formulation (set equal to 1) or a *transition probability* formulation (set equal to 0). If this variable is set to 1, then `params.possible_next_exo` and `params.transition` both need to be implemented. On the other hand, if this variable is set to 0, then `params.possible_next_states` needs to be implemented. For this storage problem, we have chosen to specify the distribution of exogenous information along with the transition function.
  - Many of the functions are specified as anonymous functions that point to `.m` files in the main folder.
- **`Sx/simulate_optimal_policy.m`**: a script file that simulates the optimal policy. `repeat` specifies the number of simulations to run and `print=1` tells the simulator to output step by step details (set `print=0` when running a large number of simulations). The results of the simulations are given in `mean_total` and `totals`. The value of `mean_total` should be close to the optimal values given in Table 1.

2.1. **Running the Optimal Policy.** To run simulations of the optimal policy for instance `Sx`, simply run the script `Sx/simulate_optimal_policy.m`. Nothing needs to be changed. The code `rng('default')` can be commented out (as it is by default) to obtain new random samples. The optimal policy can be run from any initial state — simply edit `params.initial_state` in the `parameters.m`.

2.2. **Comparing Another Policy to the Optimal Policy.** To do this, simply copy the file (from the subfolder) `Sx/simulate_optimal_policy.m` and paste it in the same location with a different name. Change the input argument `params.optimal_policy` to an anonymous function that maps the current state to the *index* or *row* of an action from the matrix of feasible actions. This function can be *any mapping*, but see `optimal_policy.m` for an example of how a tabular policy is specified. To make sure that the same random samples are being used, one must simply make sure that the line `rng('default')` is uncommented — as long as both random generators are seeded to the same values initially, the comparisons will be done on the same data.

2.3. **Training an ADP or RL Algorithm.** Typically, the ingredients needed to train an ADP or RL are the contribution function, feasible set, transition function, and a way to sample exogenous information. Since this functionality is provided, it is easy implement an ADP or RL algorithm on top of our problem framework.

Please email `drjiang@princeton.edu` with any comments, suggestions, or questions.

References

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

D. R. Jiang and W. B. Powell. An Approximate Dynamic Programming Algorithm for Monotone Value Functions. *arXiv preprint arXiv:1401.1590*, 2015.

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

N. Secomandi. Optimal Commodity Trading with a Capacitated Storage Asset. *Management Science*, 56(3):449–467, 2010.