battery-charger: Reinforcement learning techniques for wholesale market participation of grid-scale batteries

Erich Trieschman
Department of Statistics
Stanford University
Stanford, CA
etriesch@stanford.edu

***Abstract*—This paper analyzes the revenue potential of long and short-term batteries. First, I compare the theoretical bounds (optimal and naive) across battery durations. Next, I compare the performance of various reinforcement learning algorithms across the same durations.**

I. Introduction

Grid-scale batteries will play a critical role in our grid's transition to renewable energy. By capturing surplus renewable energy and dispatching it in shortages, they help reduce a grid's reliance on thermal power and lessen the physical demands on transmission lines. In this way, batteries are both load centers and generators of power and a battery schedule in one time period affects the available options at subsequent time periods [1].

Prices for electricity at each node in the grid are settled in real time every 15 and 5 minutes [2]. The optimal operating schedule can only be learned in hindsight, yet effective battery charge can still lead to meaningful revenue. The main challenge with battery management is price uncertainty. A battery operator must decide how to charge and discharge his/her battery given the available information so as to maximize the expected cummulative returns.

A further decision includes which type of battery to use: a long duration battery that can charge and discharge over the course of several days, or a short duration battery that charges and discharges over the course of several hours. While long duration batteries can use strategic reseves to dispatch more at optimal times, they are also less efficient, so gains from improved arbitrage may be offset by roundtrip charging losses.

This paper analyzes the revenue potential of long and short-term batteries. First, I compare the theoretical bounds (optimal and naive) across battery durations. Next, I compare the performance of various reinforcement learning algorithms across the same durations.

II. Methods: Data and parameters

A. Data

My analysis studies battery performance within the California wholesale electricity market. This wholesale market is operated by the California Independent System Operator (CAISO), which clears real-time electricity prices at every node on the grid every 5 and 15 minutes. CAISO data is publicly available and I use the 'gridstatus' package to access the 15-minute realtime market and the 1-hour day ahead market data. I consider data at these nodes from 2020 through 2022 [3].

There are over 1594 nodes in the CAISO market with available price data. To capture potential heterogeneity across the state, my analysis extends to four representative nodes. Two actual nodes, one in the northern zone (NP15) and one in the Southern zone (SP15), and two trading hubs. The two actual nodes I select correspond to two of PG&E's major battery projects within CAISO [4] [1], one identified as proximate to a large solar installation, and the other proximate to a large generator with existing transmission. The trading hubs are virtual hubs, developed by CAISO, to represent the average price paid to generation resources within Existing Zones to support hedging [5]. Table I provides more details on these hubs.

TABLE I
CAISO NODES

Node	Location	Capacity
SANDLOT_2_N022	Mojave, Kern Cty.	169MW676MWh
MOSSLDB_2_B1	Moss Landing, Monterey Ct.	350MW1400MWh
TH_NP15_GENAPND	N. Trading Zone	-
TH_SP15_GENAPND	S. Trading Zone	-

I reserve 2022 as my analysis year to evaluate the performance of various charging strategies. I use 2020 and 2021 data for cross validation and hyper-parameter tuning.

B. Battery parameters

My analysis uses battery parameters described in Table II. I base power capacity on the average of upcoming PG&E grid-scale battery projects [4] [1]. Duration and round-trip efficiency parameters were set in conversations with experts in grid-scale battery storage.

TABLE II
BATTERY PARAMETERS

Parameter	Value
Power capacity (MW)	200
Duration (hr)	{4, 24, 100}
Energy capacity (MWh)	Power capacity * Duration
Efficiency (%)	92.5 (12hr); 86 (24hr); 70 (100hr)

III. METHODS: BASELINE PERFORMANCE

I calculate two baselines to bound the expected performance of my reinforcement learning approaches. As an upper bound, I use linear optimization to calculate the maximum-revenue strategy; as a lower bound, I implement a naive rule of charging during non-peak hours and dispatching during peak hours.

A. Optimal baseline

With known prices, energy arbitrage can be framed as a convex optimization problem; this has been well studied and can be formulated with an objective and constraints as follows [6]:

$$\begin{aligned} \max_{c,d} \ & p^T(\nu d - \frac{1}{\nu}c) \\ s.t \ & E_{[1:T]} = E_{[0:T-1]} + c_{[0:T-1]} - d_{[0:T-1]} \\ & 0 \preceq E \preceq E_{max} \\ & C_{min} \preceq c \preceq C_{max} \\ & D_{min} \preceq d \preceq D_{max} \end{aligned}$$

Where $c, d \in \mathbb{R}^T$ are charge and discharge quantities in each period, t. $p \in \mathbb{R}^T$ is realtime prices in those corresponding periods. $E_{max}, C_{min}, C_{max}, D_{min}, D_{max} \in \mathbb{R}$ are physical constraints on total energy capacity and charge/discharge power capacity respectively. $\nu \in \mathbb{R}$ is the efficiency of charging and discharging. Note that physical constraints and efficiency vary by battery duration.

B. Optimal baseline under 10-day foresight

As a slightly tighter upper bound, I also calculate the optimal battery performance under 10-day forecast windows. Given best weather and price forecasts extend out only 10 days, this theoretical upper bound may represent performance closer to (but still likely above), the performance of reinforcement learning models.

C. Naive baseline

Given trends in energy use and reports of the net load curve in California, I develop a naive baseline set through simple charge and discharge rules throughout the day [2]. Figure 1 shows mean price at every 15 minute interval throughout the day; I use this figure to choose the specific thresholds for my charge and discharge rules, which are the same across battery durations. I provide the naive baseline charging schedule in Table III.

Fig. 1. Mean hourly price across nodes, 2020-2022

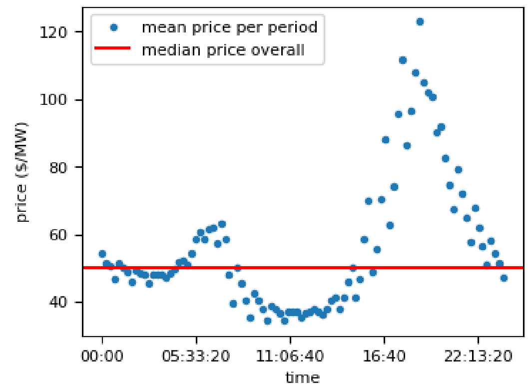


TABLE III
NAIVE BASELINE SCHEDULE

Time	Action	Storage
00:00 - 02:00	-	0
02:00 - 04:00	Charge	2 * Capacity
04:00 - 08:00	-	2 * Capacity
08:00 - 10:00	Discharge	0
10:00 - 11:00	-	0
11:00 - 15:00	Charge	4 * Capacity
15:00 - 17:00	-	4 * Capacity
17:00 - 21:00	Discharge	0
21:00 - 00:00	-	0

IV. METHODS: REINFORCEMENT LEARNING

I use reinforcement learning algorithms from AA228: Decision Making under Uncertainty to develop revenue-maximizing battery schedules [7]. I implement a model-based and a model-free Markov decision process problem (MDPs). In the sections below, I describe the structure of these problems, as well as my approach for model selection and hyperparameter tuning.

A. Action space and reward function

Consistent across both MDPs are my action space and reward function. The actions I consider are Charge, Discharge, and No Action. In any period, t, I take only one action $\in (c_t, d_t, 0)$, where

$$\begin{aligned} \text{Charge: } c_t &= \min(C_{max}, E_{max} - E_{t-1}) \\ &= \min(200MW, 200MW * Duration - E_{t-1}) \\ \text{Discharge: } d_t &= \min(D_{max}, E_{t-1} - E_{min}) \\ &= \min(200MW, E_{t-1}) \\ \text{No action: } &= 0 \end{aligned}$$

I find that a reward function based solely on revenue in each time period is not sufficient to incentivize exploration of my state space. Intuitively, this is because greedy actions always prefer revenue-generating discharge or revenue-maintaining no-action over revenue-losing charge. Wang et al. also finds that this method leads to under-exploring of the state-space and makes significant by incorporating a moving average into the reward function [8]. From the optimal baseline analysis, I also observe that longer-duration batteries' optimal performance involves maintaining partial storage throughout the period. My final reward function includes elements of these three benefits.

The revenue component follows the optimization problem above:

$$REV(s,a) = p_t * \left(\nu * d_t - \frac{1}{\nu} * c_t\right) \tag{1}$$

The moving average component rewards charging and penalizes discharging when the current price drops below the moving average; when current price rises above the moving average, the incentives are reversed:

$$MA(s,a) = \begin{cases} (pma_t - p_t) * 200 & \text{if charge} \\ (p_t - pma_t) * 200 & \text{if discharge} \\ 0 & \text{else} \end{cases} \tag{2}$$

Lastly, the energy component rewards next-period's storage:

$$ES(s,a) = \begin{cases} e_t + 200 & \text{if charge} \\ e_t - 200 & \text{if discharge} \\ e_t & \text{else} \end{cases} \tag{3}$$

My reward function is a weighted average of these three components:

$$\begin{aligned} R(s_t,a) &= \phi_r REV(s_t,a) + \phi_m MA(s_t,a) + \phi_e ES(s_t,a) \\ &\text{where } 0 \preceq \phi \preceq 1 \\ &\qquad \phi^T 1 = 1 \end{aligned}$$

I treat ϕ, the weighting of these three rewards, as a hyperparameter which I tune through crossvalidation.

B. Model-free approach

For a model-free approach, I use Q-learning with ϵ-greedy exploration and a discretized state space. This approach is an offline, model-free approach because I learn the value function offline prior to deploying my strategy, and I make no assumptions about the transition function. I use observations from historical data (from 2020 and 2021) to iteratively learn the value function and an optimal policy. Q-learning is a standard explore-exploit approach where I balance exploring unseen states (i.e., different charges under different price conditions) with exploiting greedy behavior (i.e., discharging to make revenue).

Q-learning requires a discrete state-space. My state space consists of the following discrete quantities

- S_{rt}: Real-time prices, discretized by quantile into 100 bins
- S_m: Month of year
- S_w: Weekday indicator
- S_h: Hour of day
- S_e: State of charge of battery, discretized into 50MW units

The Q-learning algorithm iterates until convergence. In each iteration, I perform batch updates of the Q-function (the value function under every action) using batches of samples, instead of a single sample point at a time. And within each batch, I iterate over every possible energy state of the battery, to improve the estimate of the value at every state of charge of the battery. I find that the Q-function converges in roughly 25 iterations and I use a batch size of 300 sample points.

In this algorithm, I define a few hyperparameters, which I tune through cross validation. Those hyperparameters are

- α: the learning rate
- ϵ: the probability of exploring (instead of taking the optimal action under the current Q-function)
- ψ: a control on the decay of ϵ over time, t, where I decay ϵ by $\exp(-t * \psi)$

C. Model-based approach

For a model-based approach, I approximate a transition function, $T(s_{t+1} \mid s_t, a)$, using a time-series model and a value function, $U^{\pi}(s) = \theta^T B(s)$ through linear regression. With this approach I can work with a continuous state space. My state space consists of the following quantities

- PRT: Real-time prices
- PDA: Day-ahead prices (reported for every hour of the day)
- M: Month of year
- W: Weekday indicator
- H: Hour of day
- E: State of charge of battery, discretized into 50MW units

Note PDA is deterministic because day ahead prices are settled the day prior to trading; hence, next hour's price is known at the beginning of the day. M, W, H have known, deterministic, transitions, as does E, which evolves depending on the decision to charge, discharge, or do nothing. Therefore, under this state space the only uncertain transition state is that of PRT, which I use time-series modeling to estimate; I include lag and square terms to better approximate the series. I present the results of this model in Tables V and VI and Figure 4 in the Appendix (Section VII).

In my model-based approximation of the value function, I approximate a unique value function for every energy state. Hence

$$\begin{aligned} U(s_t) &= U(PRT_t, PDA_t, M_t, W_t, H_t, E_t) \\ &= U_{e=E_t}(PRT_t, PDA_t, M_t, W_t, H_t) \\ &= \theta^T_{e=E_t} B(PRT_t, PDA_t, M_t, W_t, H_t) \end{aligned}$$

And under the transition states above, assuming a normally distributed error for my time-series model, I have

$$\begin{aligned} E[U(s_{t+1})] &= \theta^T_{e=E_t+1} B(E[PRT_{t+1}], PDA_{t+1}, M_{t+1}, W_{t+1}, H_{t+1}) \\ &= \theta^T_{e=E_t+1} B(\hat{PRT}_{t+1}, PDA_{t+1}, M_{t+1}, W_{t+1}, H_{t+1}) \\ &= \theta^T_{e=E_t+1} B(\hat{s}_{[-E]t+1}) \end{aligned}$$

This leads to the following value iteration

$$
\begin{aligned}
U^{(i+1)}(s_t) &\leftarrow R(s,a) + \gamma * E_{T(s_{t+1}|s,a)}[U^{(i)}(s_{t+1})] \\
&\leftarrow R(s,a) + \gamma * \theta^{T}_{e=E_t+1} B(\hat{s}_{[-E]t+1})
\end{aligned}
$$

V. Results

In this section I present my results. Table IV presents the overall performance of baseline and reinforcement learning schedules, across the 4 nodes and 3 durations that I consider. Figure 2 displays the performance of each schedule over time for the 4-hour duration battery at the Moss Landing node and Figure 3 displays the same for the 100-hour duration battery.

I am still tuning the hyperparameters for my reinforcement learning algorithms. The results in this section reflect initial guesses for all hyperparameters.

TABLE IV
Overall performance by schedule, duration, and node

Node	Dur.	naive	10d opt.	optimal	Q-learn	V-it.
TH_NP15	4hr	12.1	25.8	25.7	-30.3	-
(GEN-APND)	100hr	-9.7	15.5	21.3	-17.5	-
TH_SP15_GEN	4hr	16.6	29.6	29.4	-25.8	-
(GEN-APND)	100hr	-3.2	20.4	26.2	-20.4	-
MOSSLDB	4hr	9.9	28.2	28.1	-31.2	-
(2-B1)	100hr	-13.0	16.7	22.5	-18.8	-
SANDLOT	4hr	17.0	30.2	30.1	-25.0	-
(2-N022)	100hr	-3.0	21.1	27.0	-17.5	-

Fig. 2. 4-hr duration performance, by schedule

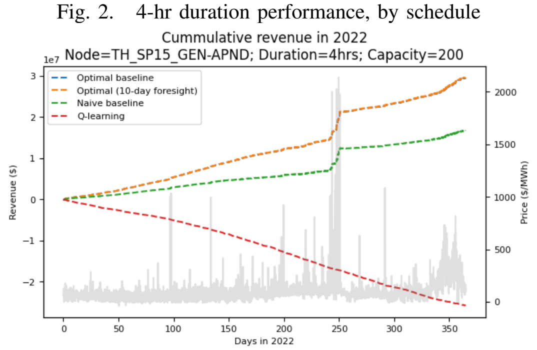
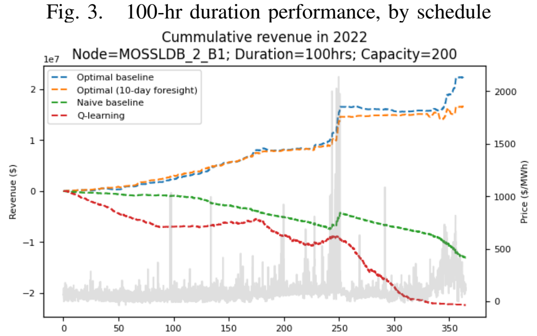


Fig. 3. 100-hr duration performance, by schedule



VI. Discussion and conclusion

From these results I make several conclusions.

- Long-duration batteries generate less revenue over a year than short-duration batteries, both in their theoretical optimal bounds and their naive baselines. While the short-duration batteries can still be profitable under a naive baseline, the long-duration batteries lose money.
- As expected, the optimal baselines under 10-day foresight are exactly the same as the optimal baselines under 1-year foresight for 4-hr duration batteries. This is not the case for long-duration batteries that can optimize behavior beyond a 10-day window.
- Without hyperparameter tuning, the reinforcement learning models underperform even the naive baselines, however with tuning I expect them to perform closer to the optimal bounds.

I identify several opportunities for the extension of this work

- Improved price forecasting: The model-based reinforcement learning approach relies on price forecasts. In this project I use simple time-lagged OLS to make these forecasts, but they could be improved with other machine learnig models.
- Richer state space: I use available price data as well as indicators for time of day and year. Incorporating weather data, gas prices, and other correlated predictors could improve estimation of the value function.
- Improved efficiency and faster runtime: I worked with a large space of hyperparameters, leading to a large set of permutations to evaluate through cross validation. I expect that using higher-performing computing, and improving the efficiency of my algorithms can improve these results.

VII. Appendix tables and figures

TABLE V
Regression results for $PRT_{t+1} \sim PRT_t + X$

Dep. Var:	y	**R-sq:**	0.737
Model:	OLS	**Adj. R-sq:**	0.736
Method:	Least Squares	**F-stat:**	3914.
No. Obs:	70073	**Prob (F-stat):**	0.00
Df Res:	70022	**Log-Li:**	-3.2620e+05
Df Model:	50	**AIC:**	6.525e+05
Cov Type:	nonrobust	**BIC:**	6.530e+05

References

[1] "Grid-scale storage," 2022. [Online]. Available: https://www.iea.org/reports/grid-scale-storage

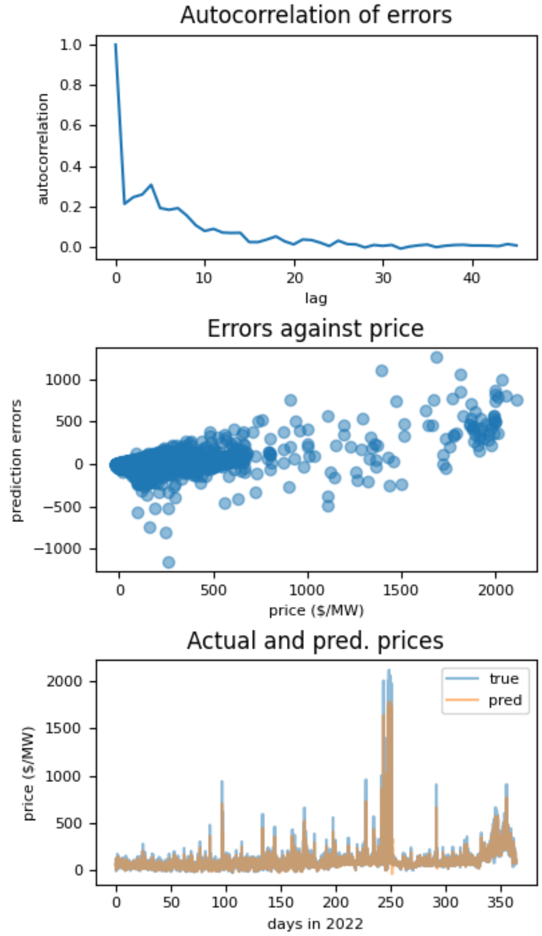
[2] 2022. [Online]. Available: http://www.caiso.com/about/Pages/default.aspx

[3] J. M. Kanter, "gridstatus 0.19.0." [Online]. Available: https://docs.gridstatus.io/en/latest/

[4] A. Colthorpe, "California utility pg&e proposes 1.6gw6.4gwh of new battery storage across nine projects," *Energy Storage News*, 5 2022. [Online]. Available: https://www.energy-storage.news/california-utility-pge-proposes-1-6gw-6-4gwh-of-new-battery-storage-across-nine-proje

[5] "Market redesign & technology upgrade (mrtu) project caiso proposal," 10 2005. [Online]. Available: https://www.caiso.com/Documents/N-5-AppendixA-October26_2005-October19_2005TradingHubWhitePaper.pdf

Fig. 4. Timeseries model performance on validation year



[6] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[7] M. Kochenderfer, T. Wheeler, and K. Wray, *Algorithms for Decision Making*. MIT Press, 2022. [Online]. Available: https://books.google.com/books?id=zKtaEAAAQBAJ

[8] H. Wang and B. Zhang, "Energy storage arbitrage in real-time markets via reinforcement learning," *CoRR*, vol. abs/1711.03127, 2017. [Online]. Available: http://arxiv.org/abs/1711.03127

TABLE VI
REGRESSION RESULTS FOR $PRT_{t+1} \sim PRT_t + X$, CONTINUED

	coef	std err	p-value
lmp_rt_m1	0.6862	0.009	0.000
lmp_rt_m1_sq	-4.6e-06	8.44e-06	0.586
lmp_rt_m2	0.0223	0.009	0.019
lmp_rt_m2_sq	-2.805e-06	9.21e-06	0.761
lmp_rt_m3	0.0573	0.009	0.000
lmp_rt_m3_sq	4.464e-06	9.23e-06	0.629
lmp_rt_m4	0.2925	0.009	0.000
lmp_rt_m4_sq	-0.0002	9.14e-06	0.000
lmp_rt_m5	-0.2004	0.009	0.000
lmp_rt_m5_sq	9.242e-05	9.24e-06	0.000
lmp_rt_m6	-0.0353	0.009	0.000
lmp_rt_m6_sq	1.895e-05	9.21e-06	0.040
lmp_rt_m7	0.0439	0.008	0.000
lmp_rt_m7_sq	3.593e-06	8.41e-06	0.669
lmp_rt_m84	0.0275	0.004	0.000
lmp_rt_m85	-0.0281	0.005	0.000
lmp_rt_m86	-0.0188	0.005	0.000
lmp_rt_m87	-0.0189	0.005	0.000
lmp_rt_m88	0.0437	0.005	0.000
lmp_rt_m89	-0.0175	0.005	0.000
lmp_rt_m90	-0.0178	0.005	0.000
lmp_rt_m91	0.0166	0.005	0.000
lmp_rt_m92	0.1056	0.005	0.000
lmp_rt_m93	-0.0918	0.005	0.000
lmp_rt_m94	0.0236	0.005	0.000
lmp_rt_m95	0.0784	0.004	0.000
lmp_da	0.0465	0.004	0.000
node	-0.3573	0.209	0.088
h_0	-0.2420	0.462	0.600
h_1	-0.4694	0.461	0.309
h_2	-0.2936	0.462	0.525
h_3	0.0505	0.462	0.913
h_4	0.7766	0.462	0.093
h_5	1.2406	0.462	0.007
h_6	-0.7611	0.464	0.101
h_7	-2.3337	0.464	0.000
h_8	-0.6525	0.467	0.162
h_9	-0.3311	0.463	0.475
h_10	-0.1346	0.463	0.771
h_11	0.0772	0.463	0.868
h_12	0.1238	0.463	0.789
h_13	0.8018	0.464	0.084
h_14	1.1200	0.467	0.016
h_15	2.9921	0.472	0.000
h_16	2.6390	0.474	0.000
h_17	4.7740	0.476	0.000
h_18	3.7959	0.479	0.000
h_19	-6.1934	0.479	0.000
h_20	-3.0551	0.478	0.000
h_21	-1.8269	0.472	0.000
h_22	-1.2875	0.467	0.006
h_23	-1.1679	0.464	0.012