Optimal Planning for Smart Energy Management of Residential Batteries

Tom Alard[1], Yvan Saeys[1,2], and Arne Gevaert[1,2]

[1] Department of Mathematics, Computer Science and Statistics, Ghent University
[2] Data Mining and Modelling for Biomedicine, VIB Center for Inflammation Research, VIB Center for AI & Computational Biology, Ghent, Belgium

Abstract. In most residential batteries, the charge/discharge policy consists of a simple rule: charge whenever energy is abundant and discharge whenever it is scarce. However, this simple algorithm is far from optimal for households with dynamic energy contracts, where the price of energy can change every hour. In this work, we use time series forecasting on past energy production and consumption data to optimize the charge/discharge policy of residential batteries in households with solar panels and dynamic, day-ahead pricing energy contracts. The resulting method requires no external inputs, relying solely on past observations of energy production and consumption. We demonstrate that our method produces 80% optimal results on a historical dataset of Belgian households with residential batteries, which is more than twice as efficient as the baseline policy.

Keywords: Linear Programming · Time Series · Energy · Optimization

1 Introduction

Solar panels play a crucial role in the transition towards renewable energy. In Flanders, many households have invested in solar panels as a way to profit financially and to reduce their ecological footprint [1]. Recently however, the Flemish government has abolished the rollback meter [4], which treated an energy surplus from solar panels as negative energy usage, allowing households to "sell" surplus energy at the same price as the energy they buy from the grid. This rollback meter has been replaced with a digital meter, which keeps track of the amount of energy taken from and put on the grid separately. As the compensation for energy that has been put on the grid is significantly lower than the price of energy taken from it, selling surplus energy has become significantly less profitable for households. This development has sparked a growing interest in ways to use the energy generated by solar panels as efficiently as possible.

Residential batteries are increasingly viewed as an essential addition to solar panels, as they allow households to store surplus energy for later use. Not only does this lead to lower energy costs, but it also contributes to a more stable power grid by flattening the so-called *duck curve* [5]. Typically, residential batteries use a simple charge/discharge policy: the battery charges whenever there is excess solar energy and discharges when solar energy is insufficient to meet demand. We will refer to this strategy as the *baseline policy.* The baseline policy is simple to implement, but it is not necessarily optimal for smart residential batteries that can consider additional contextual factors. Especially for users with dynamic energy contracts, where the price of energy is determined daily for each hour of the following day, a more sophisticated charge/discharge policy can be beneficial. For example, a user might choose to purchase extra energy during low-cost nighttime hours if a shortage of energy is anticipated the next day, when prices are higher. In this way, the development of smart charge/discharge policies can help reduce energy prices and further flatten the duck curve.

Research has already been conducted into controlling batteries with a scheduler based on predictions of future energy production and consumption [7, 13]. Recently, several new models for predicting time series data have also been introduced [12, 10]. However, there is to the best of our knowledge no existing work that combines these two domains in the context of optimizing charge/discharge policies for households with a dynamic energy contract.

In this work, we introduce a general approach to optimize charge/discharge policies for residential batteries based on linear programming and time series forecasting. The method works by forecasting the difference between energy production and consumption or *energy balance* (EB) at different timepoints and feeding these forecasts to a linear programming (LP) model. This model produces a charge/discharge policy designed to minimize energy usage. In Section 2, we provide a brief overview of the dataset and describe the methodology in more detail. Next, Section 3 contains experimental results on unseen data, showing that our method provides a significant improvement in energy usage over the naive baseline. Finally, we explore avenues for future research in Section 4.

2 Dataset and Method

In this section, we first provide an overview of the data that was used to train the EB model and the specifications of the battery used in our simulations. Next, we introduce the LP model, which computes the optimal charge/discharge policy to minimize the user's energy cost while maximizing battery lifespan based on EB forecasts. We then proceed by introducing the EB forecast model and discussing different ways in which the EB model can be integrated with the LP model in practice.

2.1 Data and Battery Specification

The dataset used to train the EB model consists of historical data on energy consumption and production for 138 Belgian households. For each household, measurements of energy consumption and production are available in intervals of 15 minutes between January 1st and November 9th of 2022. We split the data using a 70/10/20 train-validation-test split, respecting the chronological order of the data. To simulate price fluctuations in the dynamic energy contracts, historical day-ahead energy prices were obtained for the same period using the European Transparency Platform [2]. To these prices we added a grid tariff and VAT of 6%, reflecting the current situation in the Flanders region.

The specifications of the battery in the simulations were based on those of the Pylontech Force H2 residential battery [3]. This battery has a depth of discharge (DoD) of 90%, meaning that the charge should always be between 5% and 95%. We assume an efficiency of 98%, *i.e.* around 2% of energy is lost when charging or discharging the battery. The battery can charge or discharge by at most 3.55 kWh per hour and has a total capacity of 7.1 kWh. These parameters will be used to construct the LP model in Section 2.2.

2.2 Linear Programming Model

The granularity of the scheduler is limited by that of the energy production and consumption forecast. In our case, this granularity is 15 minutes, meaning that the scheduler can opt to charge, discharge, or buy a certain amount of energy from the grid every 15 minutes. For every time step t, we define the following continuous variables:

$$\begin{aligned}
c_t &: \text{Energy used to charge the battery in time step } t \\
d_t &: \text{Energy taken from the battery in time step } t \\
b_t &: \text{Energy bought from the grid in time step } t \\
a_t &: \text{Energy available in the battery at time } t \\
u_t &: \text{Energy demand from the user in time step } t \\
g_t &: \text{Energy generated by solar panels in time step } t
\end{aligned}$$

All variables are expressed in kWh. For each time step t, we can identify the following general constraints:

$$\begin{aligned}
0 \leq c_t &\leq c_{\max} \\
0 \leq d_t &\leq d_{\max} \\
0 &\leq b_t \\
a_{\min} \leq a_t &\leq a_{\max}
\end{aligned}$$

where $c_{\max}$ and $d_{\max}$ indicate the maximal amount of energy in kWh that can respectively be charged on or discharged from the battery in any 15-minute interval, and $a_{\min}$ and $a_{\max}$ indicate the minimal and maximal allowed charge of the battery in kWh at any given point in time.

The following constraints respectively describe the requirement that enough energy must be available to the user at each time step t, the basic behaviour of the battery power level and the fact that the battery cannot charge and discharge during a single time step:

$$b_t + d_t - c_t \geq u_t - g_t \tag{1}$$
$$a_t + c_t e_c - d_t e_d = a_{t+1} \tag{2}$$
$$c_t = 0 \vee d_t = 0 \tag{3}$$

where e_c and e_d represent the charging and discharging efficiency, respectively. These are values between 0 and 1 that depend on the battery model. We now proceed to the cost function that the scheduler will minimize. In order to do this, we introduce the following two variables:

$$p_{s,t} : \text{Selling price for 1 kWh of energy at time point } t$$
$$p_{b,t} : \text{Buying price for 1 kWh of energy at time point } t$$

Finally, the constant α represents the cost of charging the battery by 1 kWh. Typically, residential batteries have a guaranteed lifespan expressed in number of charging cycles. Dividing the total cost of the battery by this number of charging cycles allows us to compute the cost of a single charging cycle, and by extension the cost of charging the battery by 1 kWh. The cost function can now be expressed as follows:

$$\mathcal{L} := \sum_{t=1}^{n} b_t p_{b,t} - (g_t - u_t + b_t - c_t + d_t) p_{s,t} + \alpha c_t$$

Note that the terms $g_t p_{s,t}$ and $u_t p_{s,t}$ do not contain any variables the model can control. Therefore, these terms can be dropped from the cost function without influencing the resulting solution.

To minimize $\mathcal{L}$ using linear programming, all constraints need to be linear and all variables continuous. These requirements are met in our case, except for constraint 3, which is non-linear. However, it can easily be shown that this constraint can be dropped without loss of generality, as it can never be violated in an optimal solution. Indeed, assume $c_t, d_t > 0$ for some t. Then the corresponding terms in $\mathcal{L}$ are $(d_t - c_t)p_{s,t} + \alpha c_t$. We can then define alternative values:

$$c'_t := 0$$
$$d'_t := d_t - c_t$$

i.e. instead of charging by c_t and discharging by d_t in the same time step, we simply discharge the battery by $d_t - c_t$. The corresponding terms in $\mathcal{L}$ then become:

$$(d'_t - c'_t)p_{s,t} + \alpha c'_t = (d_t - c_t)p_{s,t}$$

which is clearly lower, as both α and c_t are greater than 0 and all other terms are unchanged. Therefore, this alternative solution is more optimal.

Because we can drop constraint 3, the problem becomes solvable using linear programming. To this end, we use the GLOP solver in the Google OR-Tools library [11]. As the problem can be solved in polynomial time, the solver can efficiently compute optimal policies given forecasts for the energy production, consumption, and buying and selling prices. We assume that perfect forecasts for the buying and selling prices for the next 24 hours are available at all times[1]. The energy production and consumption variables, in contrast, still need to be estimated.

2.3 Energy Balance Model

In this section, we introduce the forecasting model for the production and consumption variables and the dataset it was trained on. Note that the terms g_t and u_t only appear in constraint (1), and only the difference between the two variables is required to compute the constraint. For this reason, we can train a single model to predict $g_t - u_t$ rather than training two separate models for g_t and u_t. We will call this difference the *energy balance* (EB).

To train the EB model, we used the Darts library [9]. We train a TiDE [8] model, which is an encoder-decoder architecture using MLP layers. A single model is trained to forecast the energy usage of all customers, allowing for efficient application to new customers without requiring retraining. Hyperparameters were optimized using Optuna [6]. Results of the hyperparameter tuning process can be seen in Table 1.

Parameter	**Range**	**Chosen value**
hiddenSize	[256, 512, 1024]	512
numEncoderLayers	[1, 2, 3]	3
numDecoderLayers	[1, 2, 3]	3
decoderOutputDim	[4, 8, 16, 32]	4
temporalDecoderHidden	[32, 64, 128]	64
dropoutLevel	[0.0, 0.1, 0.2, 0.3, 0.5]	0.5
layerNorm	[True, False]	True
learningRate	Log-scale in [1e-5, 1e-2]	4.7e−4
revIn	[True, False]	True

Table 1. Search space for hyperparameters and the corresponding values chosen by Optuna. Based on [8].

The model was trained for 5 epochs. In each epoch, the model goes over all of the time points (except the first 192 and last 96) of each client. For each of these time points, a forecast is made for the next 96 time points (24 hours) based on the previous 192 time points (48 hours). The mean squared error (MSE) loss between this forecast and the ground truth is then used to train the model.

[1] Note that in practice, day-ahead prices are typically made public once per day. The impact of this reality on our method can be studied in future work.

2.4 Integrating EB and LP Models

In order to generate smart charge/discharge policies for residential batteries, the LP model must use the forecasts made by the EB model as input. This can be done in several ways. A naive approach would be to compute daily energy balance predictions. Using these predictions, the LP solver can compute a charge/discharge policy for the next 24 hours. However, this approach is highly inflexible and can lead to short-sighted policies. For example, as this policy only looks ahead for 96 time steps, there are no consequences for completely discharging the battery in the final step(s). Additionally, this approach would be unable to adapt to unforeseen circumstances, *e.g.* a large unexpected spike in energy usage.

To address these issues, we will use a *sliding window* approach, inspired by [7]. Instead of producing a single forecast and policy per day, this approach produces a new forecast and policy for the next 24 hours every 15 minutes. In this way, the model always has access to the most recent historical data when making decisions, and the policy always takes into account a full 24 hour window into the future, effectively resolving the issue of short-sightedness. After each time step of 15 minutes, the true energy balance for that time step becomes available. This value is then compared to the predicted energy balance and the policy is adjusted accordingly. If more energy was available than anticipated in a given time step t, the policy will be adapted as follows: if the plan was to sell energy in the next time step, then the unanticipated surplus is also sold. Otherwise, the surplus is used to either reduce d_{t+1} or increase c_{t+1}, depending on whether the original plan was to discharge or charge the battery in the next time step, respectively. If there is still a surplus of energy after d_{t+1} and c_{t+1} have reached their limits, then the surplus is used to reduce b_{t+1}. Finally, if there is still a surplus after b_{t+1} is reduced to 0, this surplus is sold to the grid. An analogous approach is used when there is an unanticipated energy deficit, first attempting to charge the battery less or discharge it more, then buying any additional required energy from the grid.

Additionally, we introduce two theoretical policies: the *locally* and *globally perfect policies*. These policies are also computed using the LP model, but are based on ground truth energy balance data which is typically not available at prediction time. However, they allow us to quantify the quality of the policy produced by our proposed method. The locally perfect policy simply replaces the 24-hour forecast with ground truth energy balance data. This is the theoretically optimal policy with a 24-hour window, which can be used to quantify the potential performance that could still be gained by improving the energy balance forecasts. On the other hand, the globally perfect policy is computed at once for the entire test period, again using ground truth energy balance data. This policy is the theoretically optimal policy, and can be used as a "gold standard."

3 Results

In this section, we compare the results of our proposed algorithm with the baseline and the locally and globally perfect theoretical models.

3.1 Model vs Baseline

Figure 1 compares the performance of our proposed method with the baseline policy and the locally perfect policy. The performance of each policy is measured as energy bill savings relative to the globally optimal policy. We see that our proposed method significantly improves on the baseline policy for nearly all customers: 134 out of 138 customers achieved a measurable improvement using the proposed method over the baseline. For the other four customers, the difference in performance was only a few cents over a two-month period.

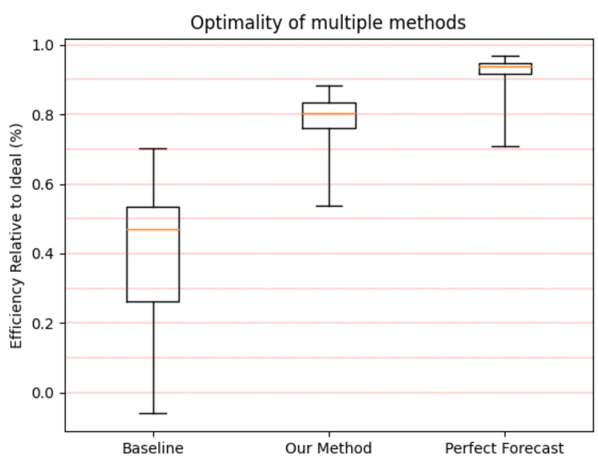

Fig. 1. Performance of the proposed method versus the baseline and locally optimal policies.

3.2 Out-of-Dataset Customers

Figure 1 shows a significant improvement in performance for in-dataset customers. In this section, we assess the generalization capability of our proposed method by measuring the performance on customers outside of the training dataset. To this end, we retrained the energy balance forecast model in a cross-validation loop, each time selecting five customers as a test set and training on the other 133. We then measure the savings achieved by the policy for those five customers relative to the savings achieved when all 138 customers are in the train set.

Figure 2 shows the results of this experiment. We can see that the policy still achieves over 98% of savings for new, entirely unseen customers. This shows a remarkable generalization ability of our proposed approach.

Fig. 2. Performance on out-of-dataset customers relative to the performance when trained on all customers.

3.3 Return on Investment

Finally, we investigate the return on investment for the residential battery under the different charge/discharge policies. When interpreting these results, it is important to note that they are likely not representative for a full calendar year. This is caused by the 70/10/20 train-validation-test split: as the chronological order of the data is respected when performing the split, the test period overlaps mostly with the autumn and winter seasons. As solar panels will generate significantly less energy during this period, it seems reasonable to assume that the resulting savings can be viewed as an approximate lower bound on the average savings for one year. Once more historical data is made available, this hypothesis can be validated. As the absolute size of the test set is not identical for all customers, we will express savings as an average amount saved in euros per day.

The results are shown in Figure 3. We see that our method saves on average around 1.1 euros per day, whereas the baseline policy only saves around 0.6. Using these estimates, we can compute a rough estimate of the return of investment of the residential battery using each method. At the time of writing, the Force H2 battery [3] has a price of 4000 euros, including installation. Assuming the household saves 1.1 euros per day leads to a yearly estimated savings of 401.5 euros per year. Note again that this is likely to be an underestimate of the true yearly savings.

Dividing the total cost of the battery by the estimated yearly savings, we arrive at an estimated payback period of 9.96 years. Performing the same computation using the baseline policy leads to an estimated payback period of 18.26 years. Obviously, this is an extremely naive computation and further research is required to validate these estimates.

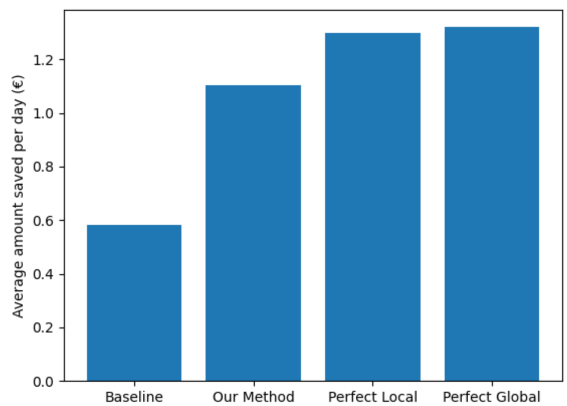

Fig. 3. Average savings per day attributed to the residential battery for each charge/discharge policy.

At the time of writing, the Force H2 battery has a warranty of seven years. We find that, using the baseline policy, none of the customers in the dataset would have recovered their investment by the end of this period. Using our proposed method, 14 customers would have done so. For the locally and globally perfect policies, this number becomes 40 and 41, respectively.

4 Conclusion

In this work, we propose a general approach for optimizing the charge/discharge policy of residential batteries to maximize financial and energy savings. Our method combines a deep learning-based energy balance forecasting model with a linear programming-based model to compute an optimal charge/discharge policy. We find that this approach achieves around 80% optimal policies on average, which is a significant improvement over the baseline performance of under 50% average optimality. Additionally, we show that the approach generalizes well, achieving around 98% of the original performance on average when applied to customers outside of the training dataset.

Although these results are promising, we identify some interesting areas of further research. First of all, because of a lack of available data, we were unable to test the performance of our method on a full calendar year. As the test period in this work did not cover the summer months, when solar energy is much more abundant, we suspect that the absolute savings estimates in this work are underestimating the true average savings for a full year. On the other hand, as the dataset only spanned part of a single calendar year (2022), results on this data may not generalize to other years, for example due to unique energy price fluctuations in the year 2022. Additionally, our results show that there is still some performance to be gained by improving the energy balance forecast model. A study of alternative architectures for this model might therefore lead to increased savings. Finally, our results were computed for a single battery model. An interesting direction for further research would therefore be to generalize the study to different models, investigating the influence of battery parameters such as charging efficiency, maximal charge and average lifespan.

References

1. Energiekaart Vlaanderen - VEKA, https://apps.energiesparen.be/energiekaart/vlaanderen/zonnepanelen
2. ENTSO-E Transparency Platform. https://transparency.entsoe.eu/
3. Pylontech Website, Pylon Technologies Co., Ltd. https://en.pylontech.com.cn/products/forceh2
4. Uitspraak (arrest) Grondwettelijk Hof over het vernietigingsberoep tegen het decreet digitale meters. https://www.vlaanderen.be/bouwen-wonen-en-energie/elektriciteit-en-aardgas/digitale-meter/uitspraak-arrest-grondwettelijk-hof-over-het-vernietigingsberoep-tegen-het-decreet-digitale-meters
5. What the duck curve tells us about managing a green grid, https://www.caiso.com/documents/flexibleresourceshelprenewables_fastfacts.pdf
6. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
7. Beaudin, M., Zareipour, H., Schellenberg, A.: Residential energy management using a moving window algorithm. In: 2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe). pp. 1–8 (2012). https://doi.org/10.1109/ISGTEurope.2012.6465896
8. Das, A., Kong, W., Leach, A., Mathur, S.K., Sen, R., Yu, R.: Long-term forecasting with tiDE: Time-series dense encoder. Transactions on Machine Learning Research (2023), https://openreview.net/forum?id=pCbC3aQB5W
9. Herzen, J., Lässig, F., Piazzetta, S.G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T.V., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., KoÅ›cisz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., Grosch, G.: Darts: User-friendly modern machine learning for time series. Journal of Machine Learning Research **23**(124), 1–6 (2022), http://jmlr.org/papers/v23/21-1177.html

10. Ma, P., Cui, S., Chen, M., Zhou, S., Wang, K.: Review of family-level short-term load forecasting and its application in household energy management system. Energies **16**(15) (2023). https://doi.org/10.3390/en16155809, https://www.mdpi.com/1996-1073/16/15/5809
11. Perron, L., Furnon, V.: Or-tools, https://developers.google.com/optimization/
12. Sabadus, A., Blaga, R., Hategan, S.M., Calinoiu, D., Paulescu, E., Mares, O., Boata, R., Stefu, N., Paulescu, M., Badescu, V.: A cross-sectional survey of deterministic pv power forecasting: Progress and limitations in current approaches. Renewable Energy (2024). https://doi.org/https://doi.org/10.1016/j.renene.2024.120385, https://www.sciencedirect.com/science/article/pii/S0960148124004506
13. Sahoo, S., Sharma, A., Chakrabarti, S., Sahoo, S.R.: Optimal residential battery scheduling using short-term forecasts. In: 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe). pp. 1–5 (2019). https://doi.org/10.1109/ISGTEurope.2019.8905741