# PyPESOL: The Python P2P Energy Sharing Optimization Library



Citation for the original published paper (version of record):
Duvignau, R., Gulisano, V., Papatriantafilou, M. (2025). PyPESOL: The Python P2P Energy Sharing Optimization Library. E-Energy '25: Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems. http://dx.doi.org/10.1145/3679240.3734691

# PyPESOL: The Python P2P Energy Sharing Optimization Library

Romaric Duvignau
Chalmers University of Technology
and University of Gothenburg
Gothenburg, Sweden
duvignau@chalmers.se

Vincenzo Gulisano
Chalmers University of Technology
and University of Gothenburg
Gothenburg, Sweden
vinmas@chalmers.se

Marina Papatriantafilou
Chalmers University of Technology
and University of Gothenburg
Gothenburg, Sweden
ptrianta@chalmers.se

## Abstract

We present the first public release of PyPESOL: the Python P2P Energy Sharing Optimization Library. PyPESOL is a lightweight, modular, and flexible framework designed for cost optimization in peer-to-peer energy sharing scenarios. It supports both single-user optimization and multi-user coordination, including automatic group formation. Users provide input data such as (forecasted or historical) electricity consumption, time-varying prices, solar generation profiles, and battery or PV capacities. Based on configurable system models (including different tariff schemes and loss assumptions), PyPESOL computes optimal battery usage decisions for individuals or groups. To enable scalable group formation, the library implements efficient greedy algorithms to partition users into energy communities. To the best of our knowledge, PyPESOL is the first publicly available framework that combines flexibility, efficiency, and scalability for P2P energy sharing optimization. Importantly, PyPESOL is also open-source: https://github.com/dcs-chalmers/pypesol.

## CCS Concepts

• **Networks** → **Peer-to-peer networks**; • **Hardware** → *Energy generation and storage*; *Power and energy*; • **Applied computing**; • **Computing methodologies** → *Optimization algorithms*;

## Keywords

Peer-to-peer energy sharing, Energy optimization, Battery scheduling, Group formation, Solar energy, Smart grid, Python library

**ACM Reference Format:**
Romaric Duvignau, Vincenzo Gulisano, and Marina Papatriantafilou. 2025. PyPESOL: The Python P2P Energy Sharing Optimization Library. In *The 16th ACM International Conference on Future and Sustainable Energy Systems (E-ENERGY '25), June 17–20, 2025, Rotterdam, Netherlands.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3679240.3734691

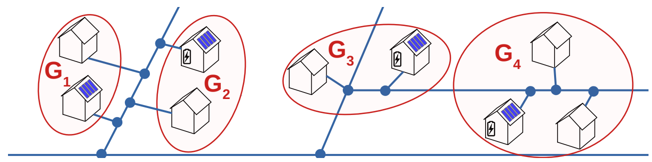


**Figure 1: Illustration of P2P energy sharing: prosumers are equipped with PV panels on roof top and possibly a battery system; energy and data are exchanged within each group.**

## 1 Introduction

Renewable electricity generation, especially from solar photovoltaic (PV) systems, is becoming increasingly affordable for residential users. As a result, households are evolving from traditional consumers into *prosumers*, users who both consume and produce electricity. In this context, *Peer-to-Peer (P2P) energy sharing* (cf. Figure 1, refer to [1, 5, 9, 10] and references therein) has emerged as a promising approach for optimizing the use of distributed renewable resources through local coordination among users. By sharing solar generation and battery storage capacity in small groups or *energy communities*, P2P setups aim to reduce individual electricity bills, enhance local self-consumption, and relieve pressure on the centralized grid. Energy can be exchanged freely or traded at regular intervals, depending on the chosen incentive and tariff models. These decentralized schemes have drawn increasing attention from the research community, motivated by their cost-saving potential and their role in supporting renewable integration. Despite the promise, many challenges remain in making P2P energy sharing practical at scale. In particular, calculating at scale local energy usage for individuals or groups as well as best *peer matching* (i.e., partitioning users into groups) are computationally intensive tasks.

## 2 A Flexible Optimization Library

To support research and development, we present **PyPESOL** — the *Python P2P Energy Sharing Optimization Library*. PyPESOL provides a modular and extensible framework for modeling and solving optimization problems related to energy consumption, storage, and sharing. It supports both individual and group-based cost optimization, and includes functionality for forming self-contained user groups using scalable heuristic algorithms. **Input**: time series data for each user: (forecasted or historical) consumption, electricity prices, solar generation profiles, capacities for the battery and PV systems. **Output**: battery usage (i.e., relative battery level) per time slot to minimize electricity costs either individually or collectively; peer partition into communities. **Models**: single-user cost-optimization [3, 5, 6] (optimal decisions using an LP-solver), including battery and transmission losses [7] or alternative grid pricing [4]. **Peer Matching**: recent configurable greedy matching algorithms [3, 5] with proven theoretical guarantees [2, 5, 8]. PyPESOL is based on the Python's `pyomo` and `coin-or/Cbc`[1] open-source LP solver. The library serves as the code-base of several ACM, IEEE and interdisciplinary publications [3–7]. A brief "getting started" Jupyter notebook illustrates its functionalities[2].

[1] **Pyomo:** http://www.pyomo.org/; **Coin-or/Cbc:** https://github.com/coin-or/Cbc.
[2] https://github.com/dcs-chalmers/pypesol/blob/main/getting_started.ipynb.


*E-ENERGY '25, Rotterdam, Netherlands*

ACM ISBN 979-8-4007-1125-1/25/06
https://doi.org/10.1145/3679240.3734691

| # P | # C | # Edges | Problem size | Time pb (s) | Matching |
|---|---|---|---|---|---|
| 0.5k | 1.8k | 67k | 1 year | 2.1 ± 0.17 | 32min |
| 0.9k | 3.6k | 270k | 6 months | 1.1 ± 0.12 | 35min |
| 2.7k | 10.7k | 2.4M | 2 months | 0.44 ± 0.06 | 44min |
| 5.8k | 23.1k | 11M | 1 month | 0.27 ± 0.04 | 1.2h |
| 11.6k | 46.2k | 45M | 2 weeks | 0.17 ± 0.03 | 2.4h |
| 23.2k | 92.4k | 182M | 1 week | 0.13 ± 0.02 | 7h |

**Table 1: Performance of PyPESOL: no. of prosumers $|P|$ (# P), no. of consumers $|C|$ (# C), no. of pairs $(p, c) \in P \times C$ (# Edges), length of data per problem, execution time per optimization problem (± standard deviation) and for the full matching.**

# 3 Optimization Models

*Single-user Cost-Optimization (LP-solver).* In this simplified model, the goal is to minimize the yearly electricity bill for a particular user following the provided inputs using the following LP-formulation. The electricity cost, $\mathrm{cost}(h, t)$, of user $h$ at hour $t$ is calculated as:

$$\mathrm{cost}(h,t) = \mathbf{el}_{in}(h,t) \cdot (\mathrm{p}(t) \cdot T + \mathrm{el}_{tax}) - \mathbf{el}_{out}(h,t) \cdot (\mathrm{p}(t) + \mathrm{el}_{net}),$$

where $\mathbf{el}_{in}(h, t)$ is the electricity bought, $\mathbf{el}_{out}(h, t)$ electricity sold, $\mathrm{p}(t)$ the market price, $T$ a taxation level, $\mathrm{el}_{tax}$ the added grid tax and $\mathrm{el}_{net}$ a small benefit for selling electricity to the grid. The grid interactions are trivially found for consumers and prosumers without a battery. For prosumers with a battery, their cost is optimized over a period of time from $t_0$ to $t_r$ via solving the following LP:

- **Objective function:** minimize $\mathrm{bill}(h, [t_0, t_r]) = \sum_{t=t_0}^{t_r} \mathrm{cost}(h, t)$;
- **Constraints (for all $t_0 \le t \le t_r$):** battery level $\mathbf{bat}(h, t)$ between 0 and capacity $\mathrm{B}_h$, and $\mathbf{bat}(h, t) = \mathbf{bat}(h, t-1) + \mathrm{el}_{gen}(h, t) - \mathrm{el}_{cons}(h, t) + \mathbf{el}_{in}(h, t) - \mathbf{el}_{out}(h, t)$, where $\mathrm{el}_{gen}(h, t)$ is $h$'s electricity generation and $\mathrm{el}_{cons}(h, t)$ its consumption, for hour $t$;
- **Opt. variables:** $\{\mathbf{bat}(h, t), \mathbf{el}_{in}(h, t), \mathbf{el}_{out}(h, t) \mid t_0 \le t \le t_r\}$, with $\mathbf{bat}(h, t_0 - 1)$ indicating the initial battery level.

*Group Cost-Optimizations.* Under the "aggregate model", groups are optimized by considering them as a single user, neglecting transmission losses. When the latter are considered, a larger LP needs to be solved (by a factor $n$ for groups of size $n$). Also, when peak-based tariffs [4] are used (charging for e.g. the max interaction with the grid in past month), slowler non-linear solvers are used.

# 4 Performance Evaluation

We briefly summarize the performance of PyPESOL, as reported in the evaluation of [5]. The library efficiently solves year-long optimization problems for individual users or aggregated groups (over 26k variables) in ca. 2s on aging consumer hardware (Intel i7-7500U, 16GB RAM). For comparison, solving a one-week problem takes only 130ms. Forming groups among 115k users (23k prosumers and 92k consumers) with group sizes up to 5 is completed in 7h, corresponding to a processing rate of roughly 7k candidate edges (i.e., allowed prosumer–consumer pairs) per second. This is made possible by PyPESOL's use of recently introduced greedy algorithms, which yield an order-of-magnitude speedup over baseline methods. Despite its scalability, PyPESOL maintains high solution quality. As illustrated in Figure 2, the library achieves around 75% of the theoretical maximum cost benefits (obtained via an unrealistic fully aggregated community) on large-scale instances. These cost savings are theoretically guaranteed [2, 5, 8] following the processing order of the nodes. In terms of execution time, PyPESOL is up to 400× faster than baseline approaches that require full edge enumeration—already several orders of magnitude faster than solving (even greedily) the full mathematical optimization problem directly [5].

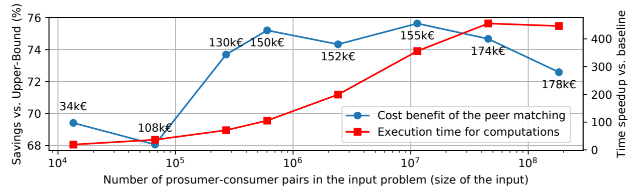


**Figure 2: Comparison of cost and processing time for matching calculations vs. upper-bound or baseline matching (calculating all pairs benefits then doing a greedy matching).**

# 5 Conclusion and Future Extensions

We introduced PyPESOL, the first public release of a comprehensive library for solving P2P energy sharing optimization problems. The library efficiently computes optimal battery usage for individuals, coordinated groups, and user groupings through scalable partitioning algorithms. Future releases will extend PyPESOL with support for more forecasting models, a broader range of tariff structures (including hourly-varying grid prices), and basic Vehicle-to-Grid (V2G) charging. We also aim to extend our optimization techniques to more energy applications and enable interaction with the optimization engine through a web interface and/or mobile app.


# Acknowledgments

This project is financed through the Swedish Electricity Storage and Balancing Centre (SESBC). The center is funded by the *Swedish Energy Agency* together with five academic and twenty-eight non-academic partners. In particular, the authors acknowledge Göteborg Energi for their contribution and support to the TANDEM project.


# References


[1] Sid Chi-Kin Chau, Khaled Elbassioni, and Yue Zhou. 2023. Approximately Socially-Optimal Decentralized Coalition Formation with Application to P2P Energy Sharing. *ACM SIGENERGY Energy Informatics Review* 2, 4 (2023), 3–17.

[2] Romaric Duvignau, Noël Gillet, and Ralf Klasing. 2024. Greediness is not always a vice: Efficient Discovery Algorithms for Assignment Problems. arXiv:2410.09434 [cs.DS] https://arxiv.org/abs/2410.09434

[3] Romaric Duvignau, Vincenzo Gulisano, and Marina Papatriantafilou. 2022. Efficient and scalable geographical peer matching for P2P energy sharing communities. In *Proc. of the 37th ACM/SIGAPP Symp. on Applied Comp. (SAC)*. 187–190.

[4] Romaric Duvignau, Vincenzo Gulisano, and Marina Papatriantafilou. 2023. Cost-optimization for win-win P2P energy systems. In *2023 IEEE Power & Energy Society Innovative Smart Grid Technologies Conf. (ISGT)*. IEEE, 1–5.

[5] Romaric Duvignau, Vincenzo Gulisano, Marina Papatriantafilou, and Ralf Klasing. 2024. Geographical Peer Matching for P2P Energy Sharing. *IEEE Access* 13 (2024), 9718 – 9738.

[6] Romaric Duvignau, Verena Heinisch, Lisa Göransson, Vincenzo Gulisano, and Marina Papatriantafilou. 2020. Small-Scale Communities Are Sufficient for Cost- and Data-Efficient Peer-to-Peer Energy Sharing. In *Proc. of the 11th ACM Int'l Conf. on Future Energy Systems (ACM e-Energy)*. 35–46.

[7] Romaric Duvignau, Verena Heinisch, Lisa Göransson, Vincenzo Gulisano, and Marina Papatriantafilou. 2021. Benefits of small-size communities for continuous cost-optimization in peer-to-peer energy sharing. *Applied Energy* 301 (2021), 117402.

[8] Romaric Duvignau and Ralf Klasing. 2023. Greediness is not always a vice: Efficient Discovery Algorithms for Assignment Problems. In *Proc. of the XII Latin-American Algorithms, Graphs and Optimization Symp. (LAGOS)*. 43–52.

[9] Wayes Tushar, Sohrab Nizami, M Imran Azim, Chau Yuen, David B Smith, Tapan Saha, H Vincent Poor, et al. 2023. Peer-to-peer energy sharing: A comprehensive review. *Foundations and Trends® in Electric Energy Systems* 6, 1 (2023), 1–82.

[10] Wayes Tushar, Chau Yuen, Tapan K Saha, Thomas Morstyn, Archie C Chapman, M Jan E Alam, Sarmad Hanif, and H Vincent Poor. 2021. Peer-to-peer energy systems for connected communities: A review of recent advances and emerging challenges. *Applied Energy* 282 (2021), 116131.