Data-driven power systems (operations)

31st European Conference on Operational Research, Athens, Greece

Juan M. Morales (juan.morales@uma.es) and others

OASYS group, University of Málaga (Spain)

July 13, 2021

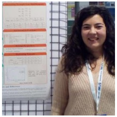
European Research Council
Established by the European Commission

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN

The others

Salvador Pineda

Yury Dvorkin



Asunción Jiménez

Ricardo Fernández-Blanco

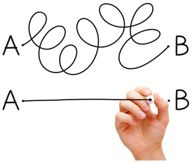
Álvaro Porras

Martine Labbé

Víctor Bucarey

Takeaway messages

- In many cases the simplicity of less is more.

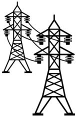


- A huge amount of opportunities and challenges at the intersection of Operations Research and Statistics.

Power Systems: A place to look for challenges

Producers

Lines



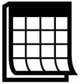
Consumers

Renewables

Storage

Decisions

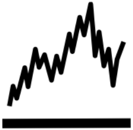
Long-term



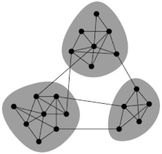
Short-term

Dimension

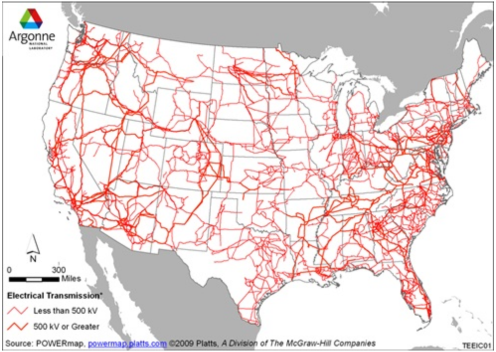
Temporal



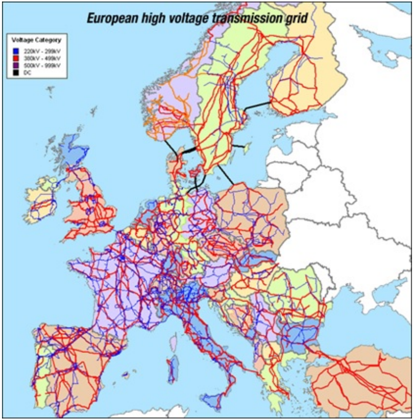
Spatial



Power Systems: A super grid of smart grids



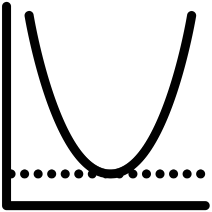
Likely, the most colossal infrastructure ever built by Homo sapiens



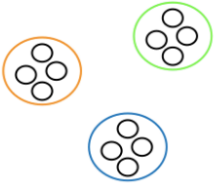
What we don't see is way much bigger!

Math Tools

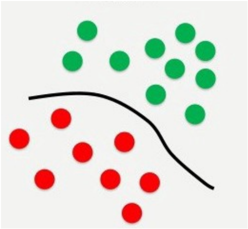
Optimization



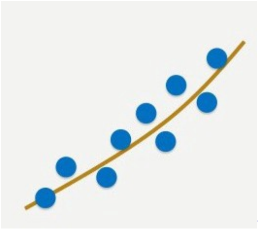
Unsupervised learning (clustering)



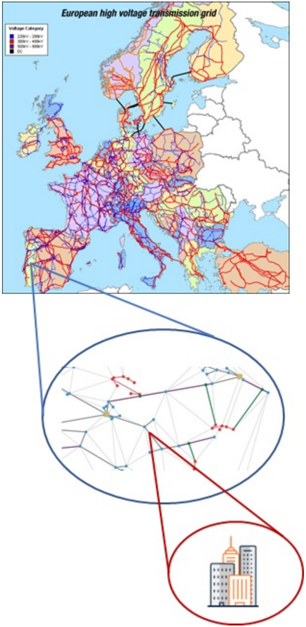
Supervised learning (classification)



Supervised learning (regression)



Challenges

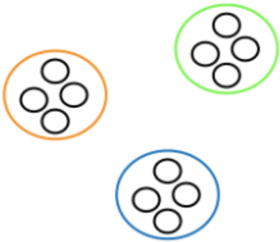


- Challenge 1: **Unit commitment problem**
- Challenge 2: TSO-DSO coordination
- Challenge 3: Power demand forecasting and bidding

Challenge 1: Time-adaptive unit commitment[1]

Unit Commitment Problem	
Horizon	24 hours
Obj	Min production cost
Var	On/off status (binary) Generating dispatches
Con	Generation = Demand Unit technical limits Line technical limits

Clustering



AIM
Reduce the cost of unit commitment without increasing the computational effort

[1] S. Pineda, R. Fernández-Blanco and J.M. Morales (2019), "Time-Adaptive Unit Commitment," IEEE Transactions on Power Systems 34(5): 3869-3878

Challenge 1: Unit commitment problem

- For given:
    - Set of generating units $g = 1, \ldots, N_G$
    - Set of time periods $t = 1, \ldots, N_T$
    - Known electricity demand d_t

Challenge 1: Unit commitment problem

- For given:
  - Set of generating units $g = 1, \ldots, N_G$
  - Set of time periods $t = 1, \ldots, N_T$
  - Known electricity demand d_t
- The unit commitment (UC) problem determines:
  - On/off status of generating units u_{gt}
  - Power dispatch of generating units p_{gt}

Challenge 1: Unit commitment problem

- For given:
  - Set of generating units $g = 1, \ldots, N_G$
  - Set of time periods $t = 1, \ldots, N_T$
  - Known electricity demand d_t
- The unit commitment (UC) problem determines:
  - On/off status of generating units u_{gt}
  - Power dispatch of generating units p_{gt}
- Mathematically, UC is formulated as a mixed-integer program (MIP):

$$\begin{aligned} \min_{u_{gt}, p_{gt}} \quad & f(u_{gt}, p_{gt}) \\ \text{s.t.} \quad & g_j(u_{gt}, p_{gt}, d_t) \leqslant 0, \quad \forall j \\ & u_{gt} \in \{0, 1\} \end{aligned}$$

Challenge 1: Unit commitment problem

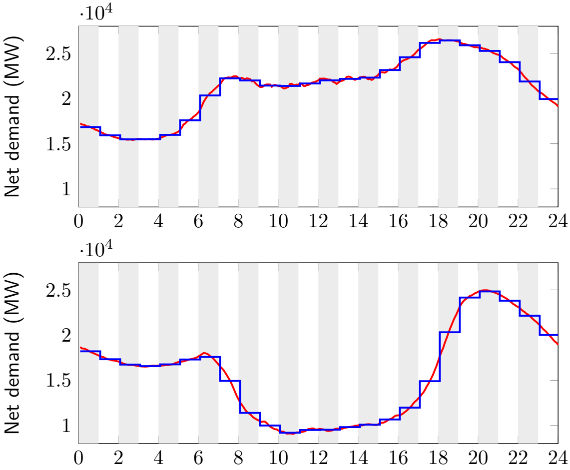
- For given:
  - Set of generating units $g = 1, \ldots, N_G$
  - Set of time periods $t = 1, \ldots, N_T$
  - Known electricity demand d_t
- The unit commitment (UC) problem determines:
  - On/off status of generating units u_{gt}
  - Power dispatch of generating units p_{gt}
- Mathematically, UC is formulated as a mixed-integer program (MIP):

$$
\begin{aligned}
\min_{u_{gt}, p_{gt}} \quad & f(u_{gt}, p_{gt}) \\
\text{s.t.} \quad & g_j(u_{gt}, p_{gt}, d_t) \leqslant 0, \quad \forall j \\
& u_{gt} \in \{0, 1\}
\end{aligned}
$$

- UC problem is computationally expensive
- Increasing N_G or N_T may turn UC intractable

Challenge 1: Current approaches

The conventional hourly approach worsens as the renewable share grows

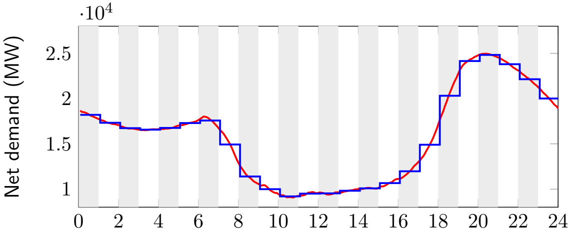


CAISO (14/01/19)
14% renewable

CAISO (22/04/19)
26% renewable

Challenge 1: Current approaches

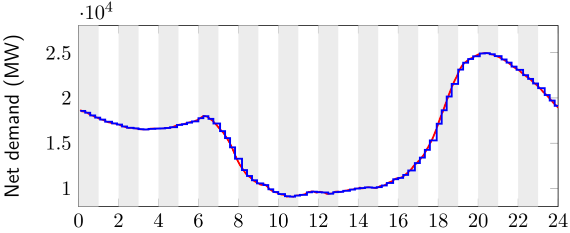
Higher time resolutions reduce cost but increase computational burden



CAISO (22/04/19)

26% renewable

hourly resolution



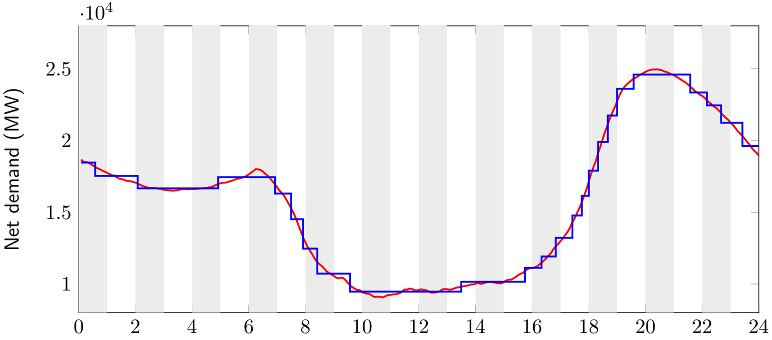
CAISO (22/04/19)

26% renewable

15-min resolution

Challenge 1: Proposed approach

- What about using 24 time periods of different duration?
- Time-adaptive unit-commitment (TA-UC)



Proposed time-period aggregation

1. Original data (5-min resolution)

Proposed time-period aggregation

1. Original data (5-min resolution)

2. Compute distance between each pair of adjacent clusters

9 7 4 2 5 8 3 2 4 5 4 6 7 7 1 9 6 2 5 2 9 3 2 6 2 9 4 4 3 2 8 3 7 5 9

Proposed time-period aggregation

1. Original data (5-min resolution)

2. Compute distance between each pair of adjacent clusters

9 7 4 2 5 8 3 2 4 5 4 6 7 7 1 9 6 2 5 2 9 3 2 6 2 9 4 4 3 2 8 3 7 5 9

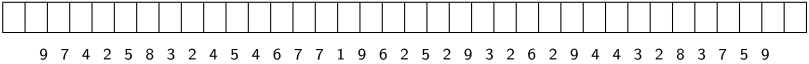
3. Merge the two closest adjacent clusters and update distances

9 7 4 2 5 8 3 2 4 5 4 6 7 6 8 6 2 5 2 9 3 2 6 2 9 4 4 3 2 8 3 7 5 9

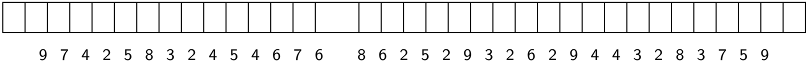
Proposed time-period aggregation

1. Original data (5-min resolution)

2. Compute distance between each pair of adjacent clusters

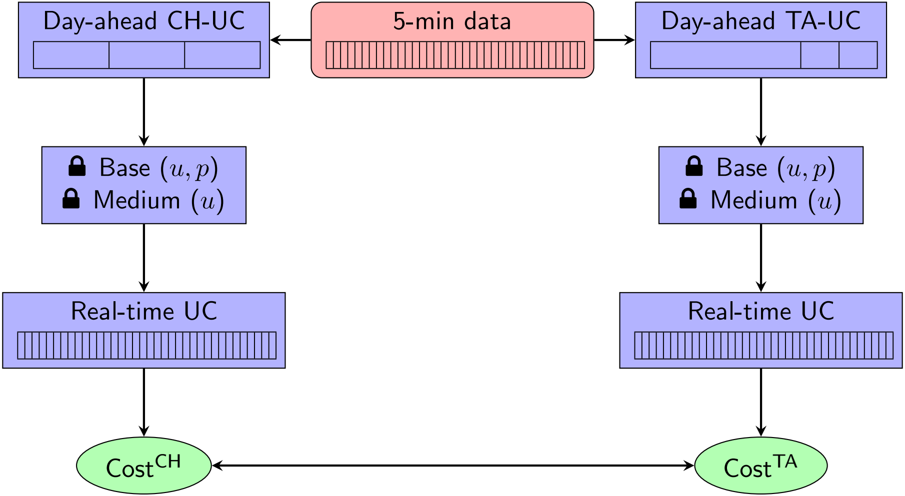


3. Merge the two closest adjacent clusters and update distances



4. Repeat 2 and 3 until the final number of clusters is obtained

Comparison: CH-UC vs. TA-UC



CH-UC vs. TA-UC: Results

- Demand is 10% of that in Spain in 2017 (3.8GW peak demand)
- Wind and solar capacity factors in Spain in 2017
- Renewable penetrations from 20% to 60%
- Start-up costs, ramp limits and minimum times of thermal units
- Three generation portfolios:

	Base (🔒u, p)	Medium (🔒u)	Peak
Normal-flex	1.2GW	1.2GW	1.5GW
High-flex	-	2.4GW	1.5GW
Low-flex	2.4GW	-	1.5GW

CH-UC vs. TA-UC: Results

		Relative cost savings		
Wind	Solar	Normal-flex	High-flex	Low-flex case
10%	10%	0.01%	0.00%	0.27%
20%	0%	0.01%	0.01%	0.30%
0%	20%	0.12%	0.07%	0.53%
30%	30%	2.35%	1.04%	3.49%
60%	0%	0.56%	0.08%	1.02%
0%	60%	2.56%	1.43%	4.76%

CH-UC vs. TA-UC: Results

		Relative cost savings		
Wind	Solar	Normal-flex	High-flex	Low-flex case
10%	10%	0.01%	0.00%	0.27%
20%	0%	0.01%	0.01%	0.30%
0%	20%	0.12%	0.07%	0.53%
30%	30%	2.35%	1.04%	3.49%
60%	0%	0.56%	0.08%	1.02%
0%	60%	2.56%	1.43%	4.76%

- More renewables ➔ Higher savings

CH-UC vs. TA-UC: Results

		Relative cost savings		
Wind	Solar	Normal-flex	High-flex	Low-flex case
10%	10%	0.01%	0.00%	0.27%
20%	0%	0.01%	0.01%	0.30%
0%	20%	0.12%	0.07%	0.53%
30%	30%	2.35%	1.04%	3.49%
60%	0%	0.56%	0.08%	1.02%
0%	60%	2.56%	1.43%	4.76%

- More renewables ➔ Higher savings
- More solar ➔ Higher savings

CH-UC vs. TA-UC: Results

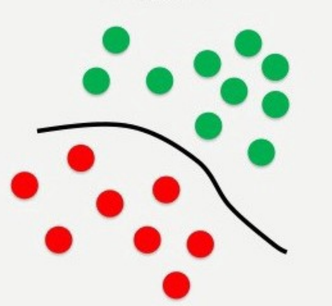
		Relative cost savings		
Wind	Solar	Normal-flex	High-flex	Low-flex case
10%	10%	0.01%	0.00%	0.27%
20%	0%	0.01%	0.01%	0.30%
0%	20%	0.12%	0.07%	0.53%
30%	30%	2.35%	1.04%	3.49%
60%	0%	0.56%	0.08%	1.02%
0%	60%	2.56%	1.43%	4.76%

- More renewables ➔ Higher savings
- More solar ➔ Higher savings
- Low flexibility ➔ Higher savings

Challenge 1: Constraint screening[2]

Unit Commitment Problem	
Horizon	24 hours
Obj	Min production cost
Var	On/off status (binary) Power dispatch
Con	Generation = Demand Unit technical limits Line technical limits

Classification



AIM

Reduce the computational time of the unit commitment problem by screening out transmission capacity constraints

[2] S. Pineda, J.M. Morales, A. Jiménez-Cordero (2020), "Data-Driven Screening of Network Constraints for Unit Commitment," IEEE Transactions on Power Systems 35(5):3695-3705.

How is the unit commitment problem formulated?

- Single-period
- DC power flow
- Thermal units
- Renewable units
- Known demand
- No failures

$$\min_{p_g, u_g, q_n, \epsilon_n} \quad \sum_g c_g p_g + L \sum_n |\epsilon_n| \tag{1a}$$

$$\text{s.t.} \quad q_n + \epsilon_n = \sum_{g : b_g = n} p_g - d_n, \forall n \tag{1b}$$

$$\sum_n q_n = 0 \tag{1c}$$

$$u_g \underline{p}_g \leqslant p_g \leqslant u_g \rho_g \overline{p}_g, \forall g \tag{1d}$$

$$-\overline{f}_l \leqslant \sum_n a_{ln} q_n \leqslant \overline{f}_l, \forall l \tag{1e}$$

$$u_g \in \{0, 1\}, \forall g \tag{1f}$$

We compare six different methods to remove constraints (1e)

Challenge 1: Constraint screening

Benchmark

- No network constraints are removed (Extremely high time)

Single-bus

- All network constraints are removed (Very fast)

Naive

- It removes line constraints that have not been congested in the past

Constraint generation

- It solves the single-bus UC and iteratively adds violated constraints

Bounding (optimization-based)[3]

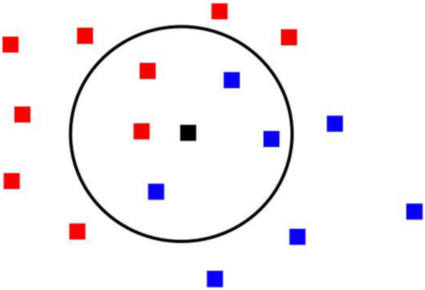
- It computes the maximum feasible flow through all lines
- It only removes redundant constraints

[3] A. Porras, S. Pineda, J. M. Morales, A. Jiménez-Cordero, "Cost-driven Screening of Network Constraints for the Unit Commitment Problem," https://arxiv.org/abs/2104.05746

Challenge 1: Proposed approach

Data-driven method (DD)

- Line congestion is inferred via statistical learning
- No need for solving additional optimization problems
- It removes not only redundant but also inactive constraints
- K-nearest neighbors is used for its simplicity and interpretability
- It can be combined with constraint generation to ensure feasibility

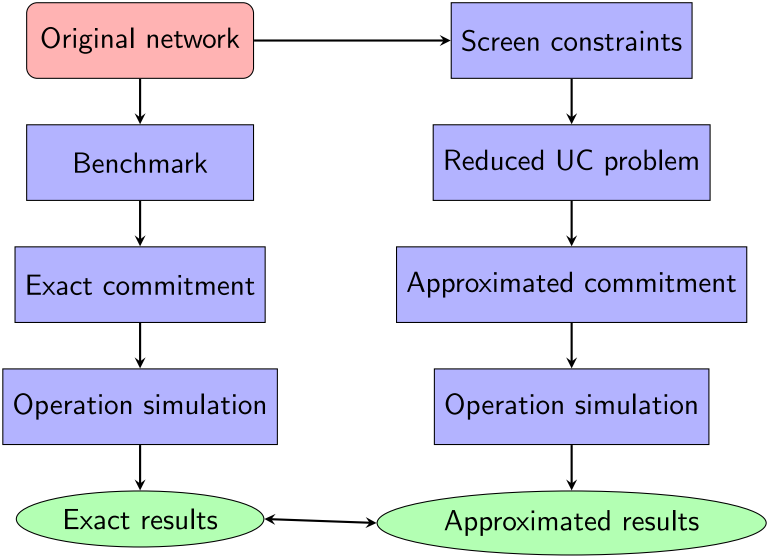


■ → Line is congested

■ → Line is not congested

■ → Line is assumed to be congested

What about the results?



Challenge 1: Results

- Power system in Texas with 2000 buses and 3206 lines
- Electricity demand at each bus is randomly sampled from a uniform distribution between 0 and twice the nominal demand
- 10% of the lines become congested during the year, and the line that most often gets congested reaches its capacity limit during 4000 hours
- 300 instances for training and 60 for testing

Challenge 1: Results

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0.0	0.00	0.00	100.0
Single-bus	100.0	-2.17	0.26	0.4
Naive	92.3	0.00	0.00	10.6
ConGen	98.8	0.00	0.00	8.9
Bounding	54.3	0.00	0.00	64.7
Data-Driven	98.6	0.04	0.03	2.3
DD+CG	98.5	0.00	0.00	5.3

- Single-bus approach is fast but provides catastrophic results
- Naive removes 92% of constraints and achieves the optimal solution
- ConGen removes a lot of constraints but requires high time
- Bounding only removes 54% of constraints and limits time reduction
- Data-Driven removes most constraints but involves small infeasibilities
- DD+CG recovers the original solution at lowest time

Challenge 1: Results

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0.0	0.00	0.00	100.0
Single-bus	100.0	-2.17	0.26	0.4
Naive	92.3	0.00	0.00	10.6
ConGen	98.8	0.00	0.00	8.9
Bounding	54.3	0.00	0.00	64.7
Data-Driven	98.6	0.04	0.03	2.3
DD+CG	98.5	0.00	0.00	5.3

- Single-bus approach is fast but provides catastrophic results
- Naive removes 92% of constraints and achieves the optimal solution
- ConGen removes a lot of constraints but requires high time
- Bounding only removes 54% of constraints and limits time reduction
- Data-Driven removes most constraints but involves small infeasibilities
- DD+CG recovers the original solution at lowest time

Challenge 1: Results

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0.0	0.00	0.00	100.0
Single-bus	100.0	-2.17	0.26	0.4
Naive	92.3	0.00	0.00	10.6
ConGen	98.8	0.00	0.00	8.9
Bounding	54.3	0.00	0.00	64.7
Data-Driven	98.6	0.04	0.03	2.3
DD+CG	98.5	0.00	0.00	5.3

- Single-bus approach is fast but provides catastrophic results
- Naive removes 92% of constraints and achieves the optimal solution
- ConGen removes a lot of constraints but requires high time
- Bounding only removes 54% of constraints and limits time reduction
- Data-Driven removes most constraints but involves small infeasibilities
- DD+CG recovers the original solution at lowest time

Challenge 1: Results

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0.0	0.00	0.00	100.0
Single-bus	100.0	-2.17	0.26	0.4
Naive	92.3	0.00	0.00	10.6
ConGen	98.8	0.00	0.00	8.9
Bounding	54.3	0.00	0.00	64.7
Data-Driven	98.6	0.04	0.03	2.3
DD+CG	98.5	0.00	0.00	5.3

- Single-bus approach is fast but provides catastrophic results
- Naive removes 92% of constraints and achieves the optimal solution
- ConGen removes a lot of constraints but requires high time
- Bounding only removes 54% of constraints and limits time reduction
- Data-Driven removes most constraints but involves small infeasibilities
- DD+CG recovers the original solution at lowest time

Challenge 1: Results

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0.0	0.00	0.00	100.0
Single-bus	100.0	-2.17	0.26	0.4
Naive	92.3	0.00	0.00	10.6
ConGen	98.8	0.00	0.00	8.9
Bounding	54.3	0.00	0.00	64.7
Data-Driven	98.6	0.04	0.03	2.3
DD+CG	98.5	0.00	0.00	5.3

- Single-bus approach is fast but provides catastrophic results
- Naive removes 92% of constraints and achieves the optimal solution
- ConGen removes a lot of constraints but requires high time
- Bounding only removes 54% of constraints and limits time reduction
- Data-Driven removes most constraints but involves small infeasibilities
- DD+CG recovers the original solution at lowest time

Challenge 1: Results

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0.0	0.00	0.00	100.0
Single-bus	100.0	-2.17	0.26	0.4
Naive	92.3	0.00	0.00	10.6
ConGen	98.8	0.00	0.00	8.9
Bounding	54.3	0.00	0.00	64.7
Data-Driven	98.6	0.04	0.03	2.3
DD+CG	98.5	0.00	0.00	5.3

- Single-bus approach is fast but provides catastrophic results
- Naive removes 92% of constraints and achieves the optimal solution
- ConGen removes a lot of constraints but requires high time
- Bounding only removes 54% of constraints and limits time reduction
- Data-Driven removes most constraints but involves small infeasibilities
- DD+CG recovers the original solution at lowest time

Challenge 1: Results

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0.0	0.00	0.00	100.0
Single-bus	100.0	-2.17	0.26	0.4
Naive	92.3	0.00	0.00	10.6
ConGen	98.8	0.00	0.00	8.9
Bounding	54.3	0.00	0.00	64.7
Data-Driven	98.6	0.04	0.03	2.3
DD+CG	98.5	0.00	0.00	5.3

- Single-bus approach is fast but provides catastrophic results
- Naive removes 92% of constraints and achieves the optimal solution
- ConGen removes a lot of constraints but requires high time
- Bounding only removes 54% of constraints and limits time reduction
- Data-Driven removes most constraints but involves small infeasibilities
- DD+CG recovers the original solution at lowest time

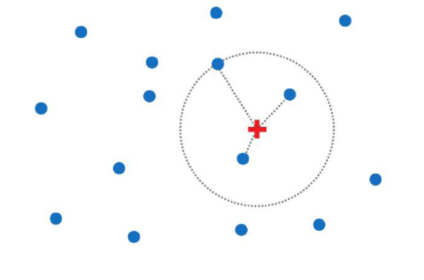
In short,

- Disregarding the capacity constraints of some transmission lines reduces the computational burden of the unit-commitment problem.
- We propose a data-driven approach that takes advantage of historical information to disregard both *redundant* and *inactive* constraints.
- Our method achieves computational savings of 70%-98% depending on the congestion level of the power system and its size.
- Combining our method with a constraint generation procedure guarantees that the solution of the original problem is recovered.

Challenge 1: Exploiting past UC solutions[4]

Unit Commitment Problem	
Horizon	24 hours
Obj	Min production cost
Var	On/off status (binary) Generating dispatches
Con	Generation = Demand Unit technical limits Line technical limits

Nearest Neighbors



AIM
Leverage past unit commitment solutions to solve new instances of the problem

[4] S. Pineda, J. M. Morales, "Is learning for the unit commitment problem a low-hanging fruit?" https://arxiv.org/abs/2106.11687

Challenge 1: Proposed approach

- We propose a naive K-nearest neighbor procedure:

  1. Find the K past UC instances that are the closest to the new one in terms of net demand profile.
  2. Take the solutions of those K instances and evaluate each of them on the new one (this implies solving K LPs).
  3. From the K candidate solutions, keep the one with the lowest cost.

Challenge 1: Results

System	Av. error (%)	Max error (%)	$< 0.01\%$	# Infes	Speedup
1888rte	0.0174	0.2394	230	1	116.5x
1951rte	0.0382	0.3759	47	8	150.4x
2848rte	0.0186	0.1332	179	2	132.6x
3012wp	0.0485	0.4864	37	5	188.8x
3375wp	0.1256	0.8073	9	13	215.9x
6468rte	-0.0001	0.0175	498	0	41.2x
6470rte	-0.0016	0.0187	496	0	171.9x
6495rte	-0.0001	0.0481	496	0	41.0x
6515rte	-0.0009	0.0133	497	0	101.7x

- 500 instances, leave-one-out
- $K = 50$
- MIP gap $= 0.01\%$

Challenge 1: Results

System	Av. error (%)	Max error (%)	$< 0.01\%$	# Infes	Speedup
1888rte	0.0174	0.2394	230	1	116.5x
1951rte	0.0382	0.3759	47	8	150.4x
2848rte	0.0186	0.1332	179	2	132.6x
3012wp	0.0485	0.4864	37	5	188.8x
3375wp	0.1256	0.8073	9	13	215.9x
6468rte	-0.0001	0.0175	498	0	41.2x
6470rte	-0.0016	0.0187	496	0	171.9x
6495rte	-0.0001	0.0481	496	0	41.0x
6515rte	-0.0009	0.0133	497	0	101.7x

- No need for complicated learning techniques for these systems, as naive strategies involve time reductions with negligible errors.
- For these systems, the naive learning strategy involves errors slightly higher than the set MIP gap, but with substantial time reductions.
- For these systems, the naive approach involves higher errors and some infeasible cases. Thus, other learning approaches may be required.

Challenge 1: Results

System	Av. error (%)	Max error (%)	$< 0.01\%$	# Infes	Speedup
1888rte	0.0174	0.2394	230	1	116.5x
1951rte	0.0382	0.3759	47	8	150.4x
2848rte	0.0186	0.1332	179	2	132.6x
3012wp	0.0485	0.4864	37	5	188.8x
3375wp	0.1256	0.8073	9	13	215.9x
6468rte	-0.0001	0.0175	498	0	41.2x
6470rte	-0.0016	0.0187	496	0	171.9x
6495rte	-0.0001	0.0481	496	0	41.0x
6515rte	-0.0009	0.0133	497	0	101.7x

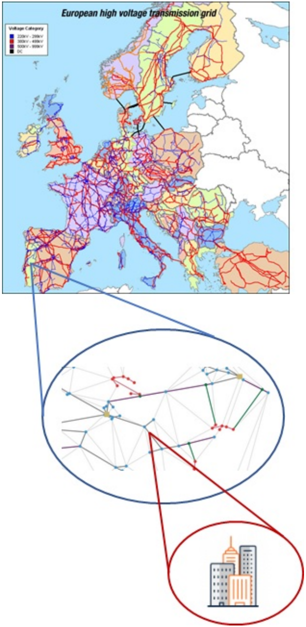
- No need for complicated learning techniques for these systems, as naive strategies involve time reductions with negligible errors.
- For these systems, the naive learning strategy involves errors slightly higher than the set MIP gap, but with substantial time reductions.
- For these systems, the naive approach involves higher errors and some infeasible cases. Thus, other learning approaches may be required.

Challenge 1: Results

System	Av. error (%)	Max error (%)	$< 0.01\%$	# Infes	Speedup
1888rte	0.0174	0.2394	230	1	116.5x
1951rte	0.0382	0.3759	47	8	150.4x
2848rte	0.0186	0.1332	179	2	132.6x
3012wp	0.0485	0.4864	37	5	188.8x
3375wp	0.1256	0.8073	9	13	215.9x
6468rte	-0.0001	0.0175	498	0	41.2x
6470rte	-0.0016	0.0187	496	0	171.9x
6495rte	-0.0001	0.0481	496	0	41.0x
6515rte	-0.0009	0.0133	497	0	101.7x

- No need for complicated learning techniques for these systems, as naive strategies involve time reductions with negligible errors.
- For these systems, the naive learning strategy involves errors slightly higher than the set MIP gap, but with substantial time reductions.
- For these systems, the naive approach involves higher errors and some infeasible cases. Thus, other learning approaches may be required.

Challenges

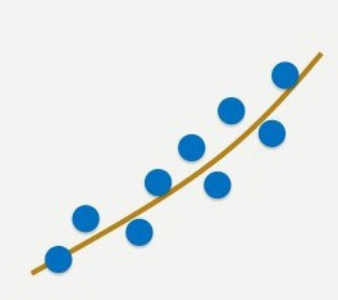


- Challenge 1: Unit commitment problem
- Challenge 2: **TSO-DSO coordination**
- Challenge 3: Power demand forecasting and bidding

Challenge 2: Overview[5]

TSO-DSO coordination	
Horizon	1 hour
Obj	Min production cost
Var	Power dispatch Substation operation
Con	Generation = Demand Unit technical limits Line technical limits

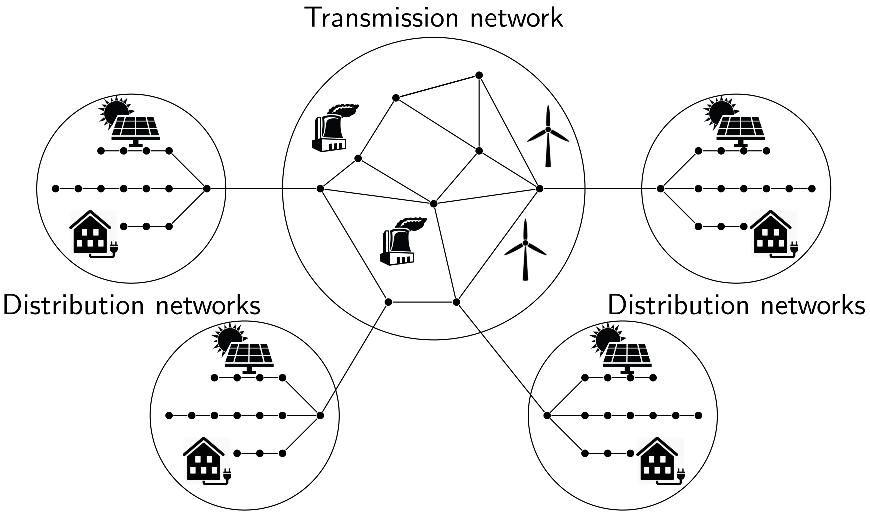
Regression



AIM
Facilitating TSO-DSO coordination by learning the response of distribution networks using contextual information

[5] J. M. Morales, S. Pineda, Y. Dvorkin, "Learning-based Coordination of Transmission and Distribution Operations," https://arxiv.org/abs/2104.06100

Challenge 2: TSO-DSOs coordination



Challenge 2: Current approaches

Benchmark

- Full representation of transmission and distribution networks.
- It requires full access to network topology and characterization, and end-users parameters. A single entity treasures all the info.
- Very high computational time.

Single-bus approach (SB)

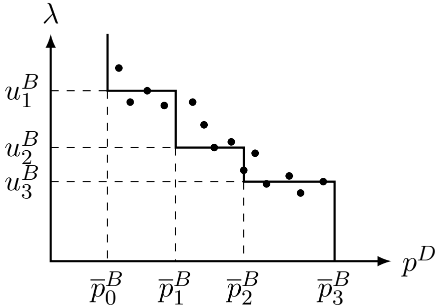
- All network constraints of DNs are ignored.
- Very low computational time.
- Good results only for uncongested DNs.

Price-agnostic approach (PAG)

- It assumes that the response of DNs is independent of price.
- It forecasts the expected response of DNs under the current context.
- Good results only for inflexible distributed generators and consumers.

Challenge 2: Contextual price-aware approach (PAW)

- It determines the step-wise non-increasing function that best approximates the response of DNs to the electricity price.[6]

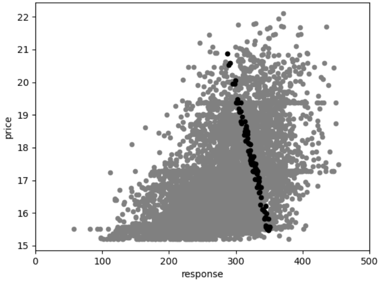


- It does not require topological info and preserves end-users' privacy.
- The curve can be directly processed by market-clearing algorithms.

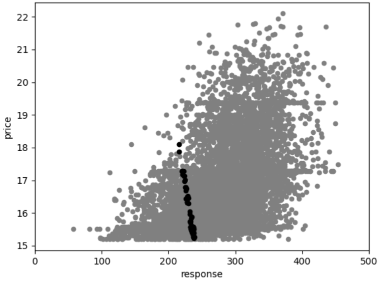
[6] V. Bucarey, M. Labbé, J. M. Morales, S. Pineda, "An exact dynamic programming approach to segmented isotonic regression," upcoming in OMEGA.

Challenge 2: Contextual price-aware approach (PAW)

Afternoon (high demand, high solar)



Night (low demand, no solar)

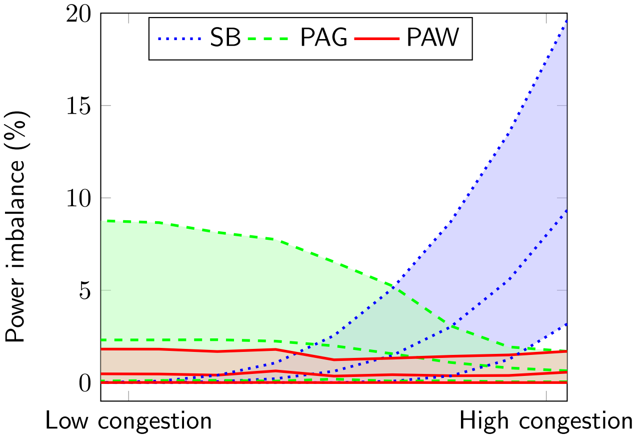


The curve changes with the context.

Challenge 2: Results

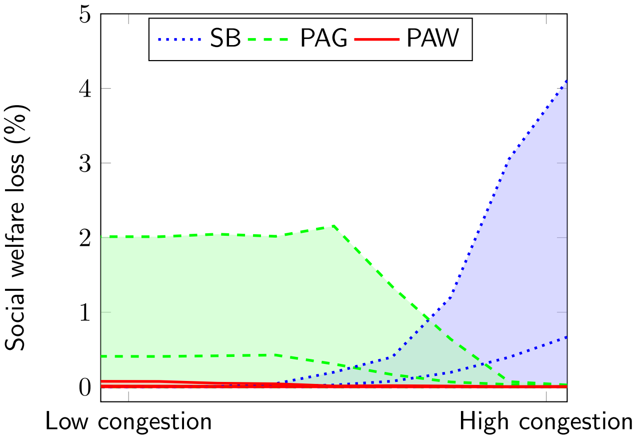
- Transmission network with 118 buses and 186 lines.
- 91 distribution networks with 32 buses and 32 lines each.
- DN include flexible consumers and solar power units.
- DN parameters are varied to simulate different congestion levels.
- 8760 hours for training and 100 hours for testing.

Challenge 2: Results



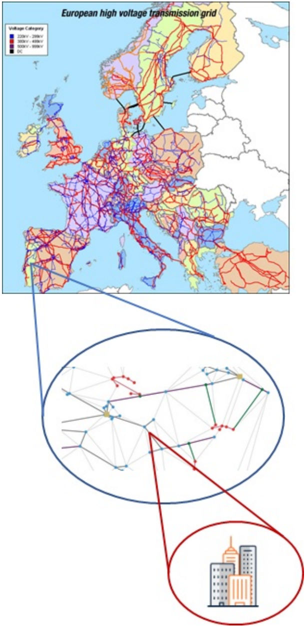
- SB works for low congestion since network can be disregarded
- PAG works for high congestion since response is independent of price
- PAW works for low and high congestion. Average imbalance $< 0.7\%$

Challenge 2: Results



- Maximum loss for SB and PAG are 2% and 4%, respectively
- Maximum loss for PAW is 0.1%

Challenges



- Challenge 1: Unit commitment problem
- Challenge 2: TSO-DSO coordination
- Challenge 3: **Power demand forecasting and bidding**

Challenge 3: Overview[7]

Load forecasting & bidding	
Horizon	minutes to one day
Obj	Min electricity bill
Var	Power consumption Loads' state variables
Con	Load state dynamics Users' comfort Devices' operation limits

Inverse optimization

AIM
Forecasting the demand of a pool of (partially) rational and flexible consumers of electricity and fostering their market participation.

[7] R. Fernández-Blanco, J.M. Morales, S. Pineda, "Forecasting the Price-Response of a Pool of Buildings via Homothetic Inverse Optimization," Applied Energy, vol. 290, pp. 116791, 2021.

Challenge 3: Key ingredients of our approach

- **Building prototype**. Represents the *average* behavior of the buildings in the pool.
- **Homothety**. The feasible operational region of the pool of buildings as a homothet of that of the building prototype.
  - Dilation factor β.
  - Translation vector $\boldsymbol{\tau}$.
- **(Piece-wise linear) utility function.** Maximization of welfare and minimization of discomfort costs. Vector of marginal utilities $\mathbf{u}$.
- **Inverse optimization (IO)**. Inference of β, $\boldsymbol{\tau}$ and $\mathbf{u}$: input parameters of an optimization problem (i.e., the forecasting model).

Challenge 3: Proposed approach

Feasible operational region of the *prototype* building

↓

Homothetic transformation (β, τ)

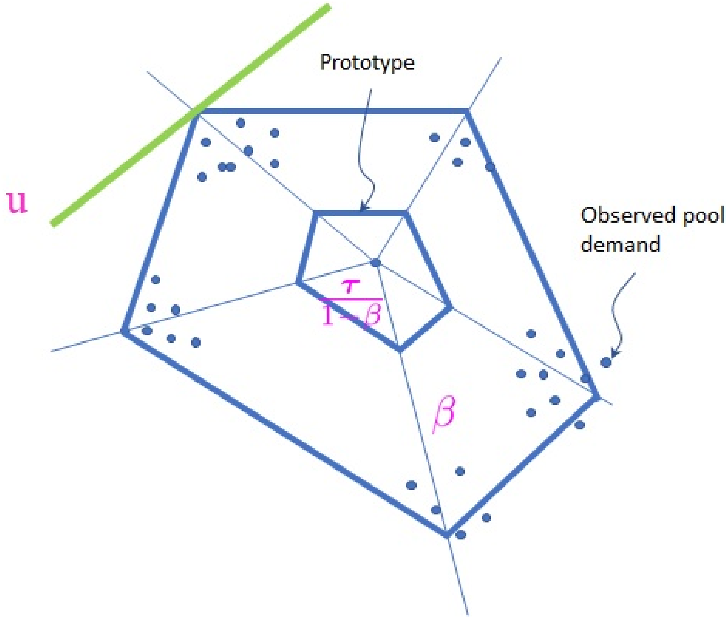
↓

Approximated feasible operational region of the *pool* of buildings

↓

Forecasting model
Maximize piece-wise linear utility function ($\mathbf{u}$, $price$)
s.t. Approximated feasible region of the pool of buildings

IO: Inference of β, τ, and m — Bilevel programming



Challenge 3: Results

- Simulations are run for two groups of 100 buildings each and 1872 hours (78 days).
- One group is composed of highly heterogeneous buildings, while the buildings in the other are quite homogeneous (in terms of thermal specifications).
- Training, validation and test: 35, 35 and 7 days, respectively.

Challenge 3: Comparison Methodologies

Acronym	Methodology
hio	**The proposed homothetic IO approach**
ntd	The two-step IO approach proposed in Saez-Gallego and Morales (2018)[8]. It disregards building thermal dynamics
arimax	AutoRegressive Integrated Moving Average Model with eXogenous variables (Python)
naive	Forecast in day d = observed values in $d-1$

[8] J. Saez-Gallego and J.M. Morales (2018). Short-term forecasting of price-responsive loads using inverse optimization. *IEEE Trans. Power Systems* 9(5):4805-4814

Challenge 3: Results

Table: Error Metric (MAE) – Comparison of Models

Model	Low heterogeneity		High heterogeneity	
	$n_B = 1$	$n_B = 6$	$n_B = 1$	$n_B = 6$
hio	52.7	52.5	22.5	16.9
ntd	87.5	88.9	24.0	26.0
arimax	108.3	108.3	23.0	23.0
naive	90.4	90.4	24.2	24.2

To take away: The aggregate demand of a highly heterogeneous pool of buildings is much easier to predict.

Challenge 3: Results

Table: Error Metric (MAE) – Comparison of Models

Model	Low heterogeneity		High heterogeneity	
	$n_B = 1$	$n_B = 6$	$n_B = 1$	$n_B = 6$
hio	52.7	52.5	22.5	16.9
ntd	87.5	88.9	24.0	26.0
arimax	108.3	108.3	23.0	23.0
naive	90.4	90.4	24.2	24.2

To take away: *hio* systematically outperforms the rest of the methods (*arimax* is competitive in the highly heterogeneous case only).

Challenge 3: Results

Table: Error Metric (MAE) – Comparison of Models

Model	Low heterogeneity		High heterogeneity	
	$n_B = 1$	$n_B = 6$	$n_B = 1$	$n_B = 6$
hio	52.7	52.5	22.5	16.9
ntd	87.5	88.9	24.0	26.0
arimax	108.3	108.3	23.0	23.0
naive	90.4	90.4	24.2	24.2

To take away: Accounting for buildings' thermal dynamics is significantly advantageous (compare *hio* with *ntd*).

Challenge 3: Results

Table: Error Metric (MAE) – Comparison of Models

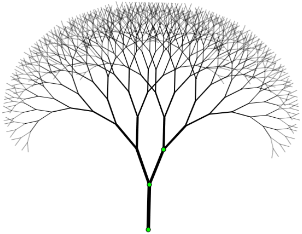
Model	Low heterogeneity		High heterogeneity	
	$n_B = 1$	$n_B = 6$	$n_B = 1$	$n_B = 6$
hio	52.7	52.5	22.5	16.9
ntd	87.5	88.9	24.0	26.0
arimax	108.3	108.3	23.0	23.0
naive	90.4	90.4	24.2	24.2

To take away: The number of marginal utility blocks has a substantial impact in the highly heterogeneous case.

To wrap up...

- Before considering the vast and frenetic world of black-box models, deep learning, big data... to solve a problem, STOP and THINK whether it is really worth the pain (recall challenge 1).

- But don't get me wrong... new problems may indeed call for new tools. Plenty of room for progress (challenges 2 and 3).



Thanks for the attention!

Questions?

More info: oasys.uma.es
Email: juan.morales@uma.es

Building prototype: Feasible region

$$\theta_h^p = a_1 \theta_{h-1}^p + (1 - a_1)\left[\theta_h^{amb} - a_2 p_h^p\right], \quad \forall h \in \mathcal{H} \tag{2a}$$

$$\underline{\theta}_h^p \leqslant \theta_h^p \leqslant \overline{\theta}_h^p, \quad \forall h \in \mathcal{H} \tag{2b}$$

$$0 \leqslant p_h^p \leqslant P, \quad \forall h \in \mathcal{H} \tag{2c}$$

- θ_h: Indoor temperature in time period h, comfort bounds $\underline{\theta}_h$, $\overline{\theta}_h$.
- p_h: Cooling power, rated power P.
- θ_h^{amb}: Ambient temperature in time period h.
- a_1, a_2: Building parameters (depending on thermal resistance and capacitance, COP and discretization time step).

Building prototype: Feasible region in matrix form

$$\mathbf{0} \leqslant \boldsymbol{p}^p \leqslant \mathbf{P} \;\; (= P \cdot \mathbf{1}_{|\mathcal{H}|}) \tag{3a}$$

$$\underline{\boldsymbol{\theta}}^p \leqslant \boldsymbol{\Lambda} \boldsymbol{B} \boldsymbol{p}^p + \boldsymbol{\Lambda}(\boldsymbol{c}^p + \boldsymbol{t}^p) \leqslant \overline{\boldsymbol{\theta}}^p \tag{3b}$$

- $\boldsymbol{c}^p = [a_1 \theta_0^p, 0, \ldots, 0]^T$: Vector of initial conditions.
- $\boldsymbol{t}^p = \boldsymbol{\theta}^{amb}(1 - a_1)$.

Aggregate building model: Feasible region in matrix form

$$\mathbf{0} \leqslant \boldsymbol{p}^p \leqslant \mathbf{P} \quad (= P \cdot \mathbf{1}_{|\mathcal{H}|}) \tag{2a}$$

$$\underline{\boldsymbol{\theta}}^p \leqslant \boldsymbol{\Lambda} \boldsymbol{B} \boldsymbol{p}^p + \boldsymbol{\Lambda}(\boldsymbol{c}^p + \boldsymbol{t}^p) \leqslant \overline{\boldsymbol{\theta}}^p \tag{2b}$$

Homothety: $\boldsymbol{p}^a = \beta \boldsymbol{p}^p + \boldsymbol{\tau}$

$$\boldsymbol{\tau} \leqslant \boldsymbol{p}^a \leqslant \beta \mathbf{P} + \boldsymbol{\tau} \tag{3a}$$

$$\beta \underline{\boldsymbol{\theta}}^p + \boldsymbol{\Lambda} \boldsymbol{B} \boldsymbol{\tau} \leqslant \boldsymbol{\Lambda} \boldsymbol{B} \boldsymbol{p}^a + \boldsymbol{\Lambda} \beta \left(\boldsymbol{c}^p + \boldsymbol{t}^p \right) \leqslant \beta \overline{\boldsymbol{\theta}}^p + \boldsymbol{\Lambda} \boldsymbol{B} \boldsymbol{\tau}. \tag{3b}$$

Forecasting model

Forecast for day d, given the price vector $\boldsymbol{\lambda}_d$ and discomfort costs $\mathbf{c}^s$:

$$\max_{\boldsymbol{p}^a_{b,d}, \boldsymbol{s}^a_d} \quad \sum_{b \in \mathcal{B}} \left(\boldsymbol{m}_{b,d} - \boldsymbol{\lambda}_d\right)^T \boldsymbol{p}^a_{b,d} - (\boldsymbol{c}^s)^T \boldsymbol{s}^a_d \tag{4a}$$

subject to:

$$\boldsymbol{\tau} \leqslant \sum_{b \in \mathcal{B}} \boldsymbol{p}^a_{b,d} \leqslant \beta \mathbf{P} + \boldsymbol{\tau} \tag{4b}$$

$$\beta \underline{\boldsymbol{\theta}}^p_d + \boldsymbol{\Lambda B \tau} - \boldsymbol{s}^a_d \leqslant \sum_{b \in \mathcal{B}} \boldsymbol{\Lambda B p}^a_{b,d} + \boldsymbol{\Lambda} \beta \left(\boldsymbol{c}^p_d + \boldsymbol{t}^p_d\right) \tag{4c}$$

$$\sum_{b \in \mathcal{B}} \boldsymbol{\Lambda B p}^a_{b,d} + \boldsymbol{\Lambda} \beta \left(\boldsymbol{c}^p_d + \boldsymbol{t}^p_d\right) \leqslant \beta \overline{\boldsymbol{\theta}}^p_d + \boldsymbol{\Lambda B \tau} + \boldsymbol{s}^a_d \tag{4d}$$

$$0 \leqslant \boldsymbol{p}^a_{b,d} \leqslant \overline{\boldsymbol{e}}_{b,d}, \quad \forall b \in \mathcal{B} \tag{4e}$$

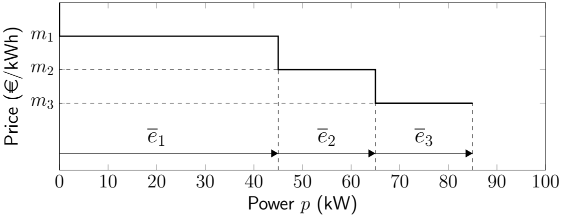
$$\boldsymbol{s}^a_d \geqslant 0, \tag{4f}$$

$\boldsymbol{m}_{b,d}$: Step-wise marginal utilities, $\boldsymbol{p}^a_d = \sum_{b \in \mathcal{B}} \boldsymbol{p}^a_{b,d}$

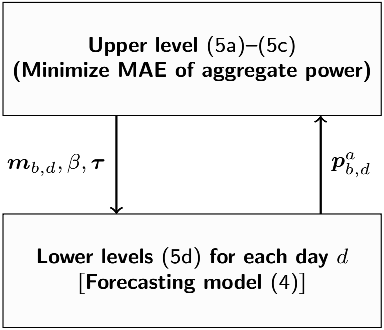
Forecasting model

$\boldsymbol{m}_{b,d}$: Step-wise marginal utilities, $\boldsymbol{p}^a_d = \sum_{b \in \mathcal{B}} \boldsymbol{p}^a_{b,d}$

Example: Three step-wise marginal utility function:



$\boldsymbol{m}_{b,d} = \boldsymbol{\nu}_b + \mathbf{Z}_d \boldsymbol{\rho}$, $\quad \mathbf{Z}_d$: Covariates, regressors or features



$$\min_{\Xi} \sum_{d\in\mathcal{D}} \left\| \sum_{b\in\mathcal{B}} \boldsymbol{p}^{a}_{b,d} - \boldsymbol{p}^{a'}_{d} \right\|_1 \tag{5a}$$

subject to:

$$\boldsymbol{m}_{b,d} = \boldsymbol{\nu}_b + \boldsymbol{Z}_d \boldsymbol{\rho}, \quad \forall b \in \mathcal{B}, d \in \mathcal{D} \tag{5b}$$

$$\boldsymbol{\nu}_b \geqslant \boldsymbol{\nu}_{b+1}, \quad \forall b < n_B \tag{5c}$$

$$\text{Forecast. model}(\boldsymbol{m}_{b,d}, \beta, \boldsymbol{\tau}), \quad d \in \mathcal{D}. \tag{5d}$$

$$\Xi = \{\boldsymbol{m}_{b,d}, \boldsymbol{p}^{a}_{b,d}, \boldsymbol{s}^{a}_{d}, \beta, \boldsymbol{\tau}, \boldsymbol{\nu}_b, \boldsymbol{\rho}\}$$

$\boldsymbol{p}^{a'}_{d}$: Vector of observed aggregate power in day d

Solution approach: Regularization + Initialization[9]

[9] R. Fernández-Blanco, J.M. Morales and S. Pineda, "Forecasting the price-response of a pool of buildings via homothetic inverse optimization." arXiv:2004.09819v2.

Data generation

We assume that the consumption of each building i for each day d is given by:

$$\min_{p_h, s_h, \theta_h} \quad \sum_{h \in \mathcal{H}} (p_h \lambda_h + \varrho s_h) \tag{6a}$$

$$\theta_h = a_1 \theta_{h-1} + (1 - a_1) \left[\theta_h^{amb} - a_2 p_h \right], \forall h \in \mathcal{H} \tag{6b}$$

$$- s_h + \underline{\theta}_h \leqslant \theta_h \leqslant \overline{\theta}_h + s_h, \quad \forall h \in \mathcal{H} \tag{6c}$$

$$0 \leqslant p_h \leqslant \overline{p}_h, \quad \forall h \in \mathcal{H} \tag{6d}$$

$$s_h \geqslant 0, \quad \forall h \in \mathcal{H} \tag{6e}$$

The pool demand is driven by the heterogeneity factor $\hbar$. For instance:

$$C^i \rightsquigarrow U\big[(1 - \hbar)C^p, (1 + \hbar)C^p\big]$$

with C^p being the thermal capacitance of the prototype building.

Data generation

Table: Statistics on the aggregate power

	$\hbar = 0.1$	$\hbar = 0.75$
Maximum (kW)	541.0	218.4
Mean (kW)	64.0	42.0
# hours without consumption (%)	61.8	0.0

- Simulations are run for 100 buildings and 1872 hours (78 days) using model (6), for $\hbar = 0.1$ (low heterogeneity) and $\hbar = 0.75$ (high heterogeneity).
- Training, validation and test: 35, 35 and 7 days, respectively.
- Covariates ($\boldsymbol{Z}_d$): Ambient temperature at hours $h-2$, $h-1$, h, $h+1$ and $h+2$.
- CONOPT under Pyomo 3.7.3 for solving the regularized nonlinear programs; CPLEX for the linear programs.