Machine-learning aided operation and planning of power systems

FlexAnalytics Symposium

Salvador Pineda and Juan Miguel Morales

OASYS group, University of Málaga (Spain)

September 26, 2022

About Málaga

- Over 300 sundays per year (known as Costa del Sol)
- University of Málaga was stablished in 1972 and currently has 40000 students and 2500 faculty members
- Málaga is becoming the Silicon Valley of the south of Spain
- Andalusia Technology Park includes over 600 companies (Oracle, Ericsson, IBM, TDK, Huawei, Microsoft, Cisco), 20.000 employees and a turnover of 2.000 M€ in 2018

Google To Open A Cybersecurity 'Centre Of Excellence' In Malaga

By **Chris King** - 11 February 2021 @ 21:05

About OASYS

Optimization and **A**nalytics for **S**ustainable energ**Y** **S**ystems (2018)

- 3 professors
- 3 Postdoc
- 4 PhD students
- 2 research assistants
- 1 support assistant

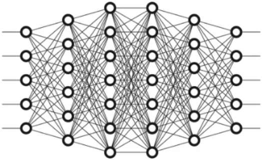
Research topics:

- Mathematical models for decision-making under uncertainty
- Use of large amounts of data for Smart Energy Grids
- Forecasting and optimization for Sustainable Energy Systems
- Algorithms to solve large-scale optimization problems
- Game theory for the analysis of energy markets

More info: oasys.uma.es

Machine learning in power systems

- The blast wave of machine learning has reached power systems
- Most papers propose black-box tools that run as follows:
  - Select a very hard problem to solve (usually NP-hard)
  - Set hyperparameters $\mu, \epsilon_0, \alpha, \beta, \lambda, \ldots$ (without explaining how)
  - Train a deep neural network (using available software)



  - Discuss the computational savings of the proposed learning-based method with that of solving the original problem

Questions we want to address

- Are power system problems so complex that any fortuitous learning-based method may involve significant computational savings?
- Do simpler and interpretable learning-based methods perform similarly to complex black-box methods?
- Should black-box methods be benchmarked against simpler methods in power systems applications?

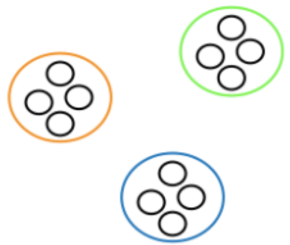
Applications

- Application 1: capacity expansion planning
- Application 2: screening of network constraints
- Application 3: coordination of TSO and DSO
- Application 4: learning unit commitment

Application 1: Capacity expansion planning[1]

Capacity expansion problem	
Horizon	1 year
Obj	Min prod+inv cost
Var	Generation capacities Line capacities
Con	Generation = Demand Unit technical limits Line technical limits

Clustering



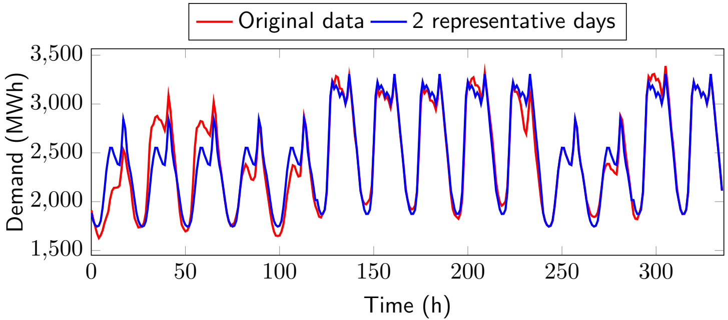
AIM

Reduce the computational burden of capacity planning problems with renewables and storage

[1] S. Pineda and J.M. Morales, "Chronological time-period clustering for optimal capacity expansion planning with storage", in IEEE Transactions on Power Systems, vol. 33, no. 6, pp. 7162 - 7170, 2018

Application 1: Capacity expansion planning

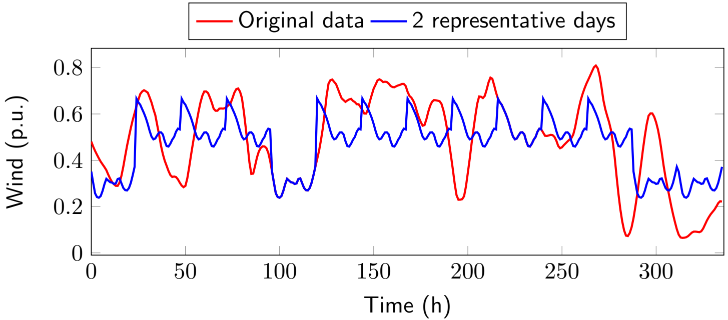
- Electrical demand shows strong daily, weekly and annual patterns and therefore, using representative days works quite well



- Instead of 14 days (336 h), we use 2 representative days (48 h)
- Computational burden is reduced
- Day-to-day chronology information is lost

Application 1: Capacity expansion planning

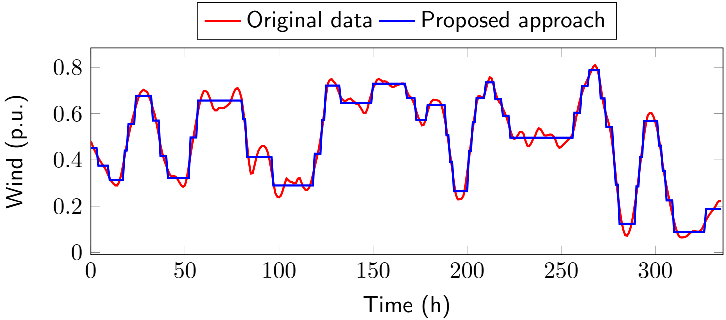
- Some renewables do not present a strong daily pattern



- Energy stored can be used several days later (**seasonal storage**)
- Crucial to overcome the "dark calm" periods in central Europe
- Seasonal storage cannot be modeled using representative days

Application 1: Capacity expansion planning

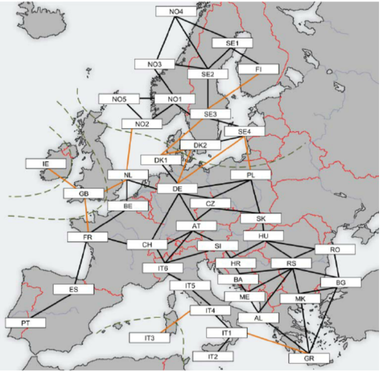
- We propose a new clustering methodology to group **consecutive hours** and maintain chronology
- Wind is approximated more accurately



- Seasonal storage can be properly modeled
- Computational burden is reduced
- Day-to-day chronology information is kept

Application 1: Capacity expansion planning

- Electric power system (28 countries) for 2030 (single target year)
- Investments in conventional and renewable generation, transmission lines and two storage technologies (intraday and interday)
- Greenfield approach (no initial capacities)
- Given renewable penetration target



Application 1: Capacity expansion planning

- Average cost increase over all renewable penetration levels.

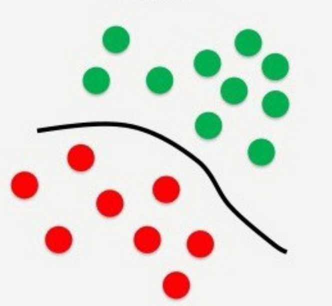
Approach	Number periods	Av. cost increase	Time
F	8760	0 %	~ 10 h
D-28	$28 \times 24 = 672$	13.1 %	~ 100 s
W-4	$4 \times 168 = 672$	48.1 %	~ 100 s
C-672	672	6.1 %	~ 100 s

- The proposed approach is the closest to the full model

Application 2: Screening of network constraints[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.

Application 2: Screening of network constraints

The unit commitment problem can be generally formulated as follows:

$$\min_{p,u} \quad c^\top p \tag{1a}$$

$$\text{s.t.} \quad \mathbf{1}^\top p = \mathbf{1}^\top d \tag{1b}$$

$$u\underline{p} \leqslant p \leqslant u\overline{p} \tag{1c}$$

$$-\overline{f} \leqslant \mathsf{PTDF}(p-d) \leqslant \overline{f}, \tag{1d}$$

$$u \in \{0,1\} \tag{1e}$$

We compare six different methods to remove some constraints (1d)

Application 2: Screening of network constraints

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

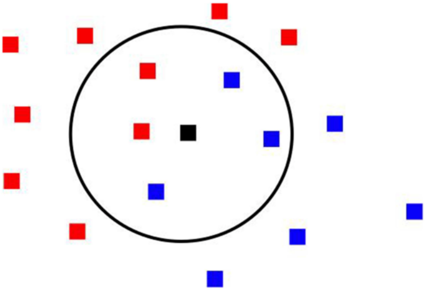
- It solves the single-bus UC and iteratively adds violated constraints

Bounding (Roald and Molzahn 2019)

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

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

Application 2: Screening of network constraints

- 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

Application 2: Screening of network constraints

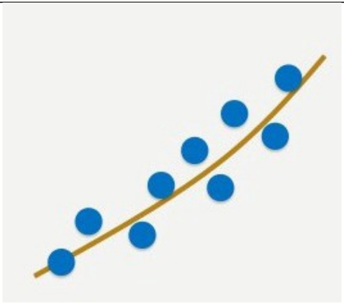
Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0.0	0.00	0.00	100.0
Single-bus	100.0	-2.17	0.26	0.4
Naive	92.3	0.00	0.00	10.6
Constraint Gen (CG)	98.8	0.00	0.00	8.9
Bounding	54.3	0.00	0.00	64.7
Data-Driven (DD)	98.6	0.04	0.03	2.3
DD+CG	98.5	0.00	0.00	5.3

- 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

Application 3: coordination of TSO and DSO[3]

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

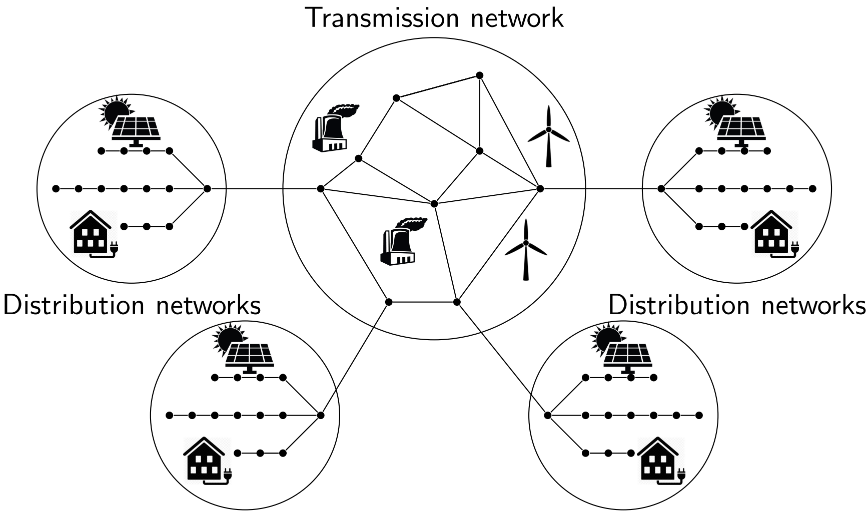
Regression



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

[3] J. M. Morales, S. Pineda, Y. Dvorkin, "Learning-based Coordination of Transmission and Distribution Operations" IEEE Transactions on Power Systems, 2022.

Application 3: coordination of TSO and DSO



Application 3: coordination of TSO and DSO

Benchmark

- Full representation of transmission and distribution networks.
- A single entity treasures all the info.
- Very high computational time.

Single-bus approach (SB)

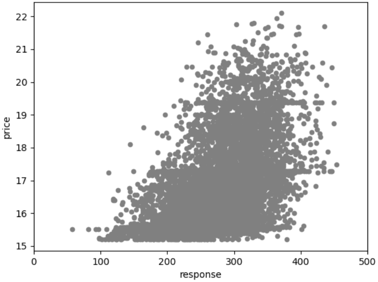
- All network constraints of DNs are ignored.
- Very low computational time.
- Good results only for uncongested DNs.

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.

Contextual price-aware approach (PAW)

1) For each DN, we have access to price-response historical data



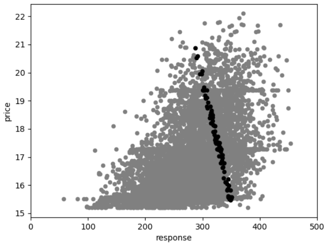
2) For a given context, we select the closest price-response pairs
3) We approximate the data using a step-wise non-increasing function
4) We operate the TN using the step-wise functions

Application 3: coordination of TSO and DSO

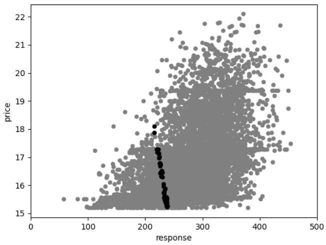
Contextual price-aware approach (PAW)

1) For each DN, we have access to price-response historical data
2) For a given context, we select the closest price-response pairs

Afternoon



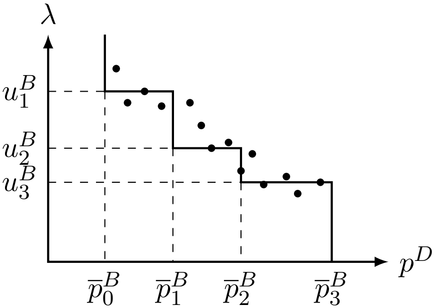
Night



3) We approximate the data using a step-wise non-increasing function
4) We operate the TN using the step-wise functions

Contextual price-aware approach (PAW)

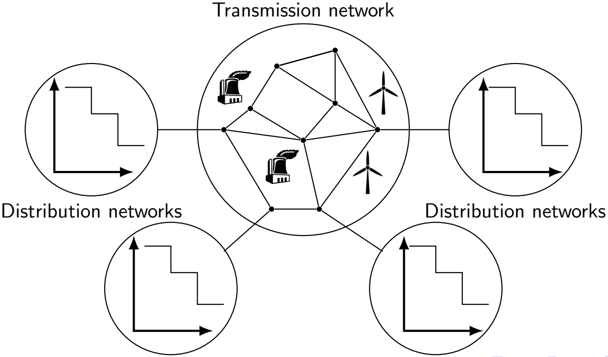
1) For each DN, we have access to price-response historical data
2) For a given context, we select the closest price-response pairs
3) We approximate the data using a step-wise non-increasing function



4) We operate the TN using the step-wise functions

Contextual price-aware approach (PAW)

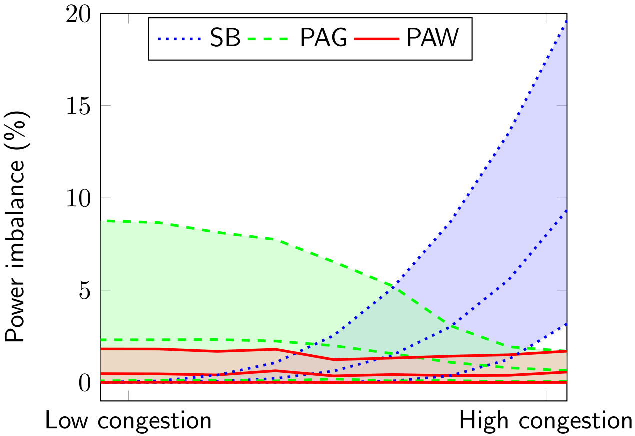
1) For each DN, we have access to price-response historical data
2) For a given context, we select the closest price-response pairs
3) We approximate the data using a step-wise non-increasing function
4) We operate the TN using the step-wise functions



Application 3: coordination of TSO and DSO

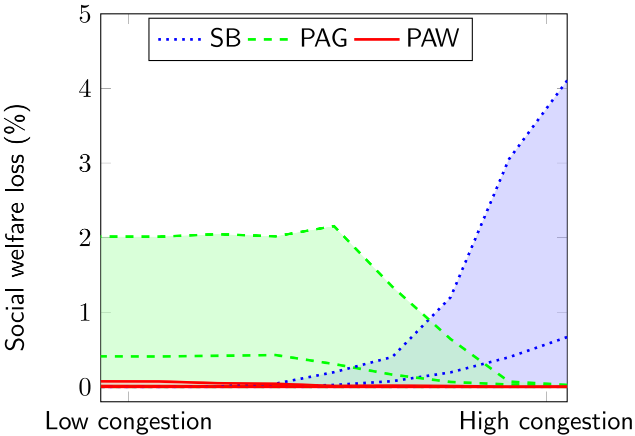
- Transmission network with 118 buses and 186 lines.
- 91 distribution networks with 32 buses and 32 lines each.
- DN include flexible consumers and solar power units.
- DN parameters are varied to simulate different congestion levels.
- 8760 hours for training and 100 hours for testing.

Application 3: coordination of TSO and DSO



- 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 $< 0.7\%$

Application 3: coordination of TSO and DSO

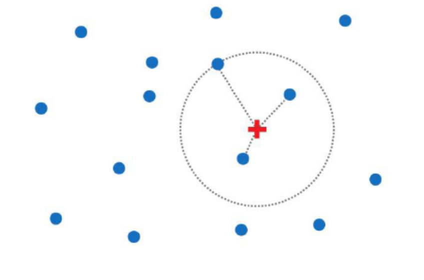


- Maximum loss for SB and PAG are 2% and 4%, respectively
- Maximum loss for PAW is 0.1%

Application 4: learning unit commitment[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?", Electric Power Systems Research, vol. 207, 2022

Application 4: learning unit commitment

$$
\begin{aligned}
\min_{\mathbf{x}\in\mathbb{R}^n,\, \mathbf{y}\in\{0,1\}^m} \quad & f(\mathbf{x},\mathbf{y}) && \text{(2a)}\\
& g_i(\mathbf{x},\mathbf{y}) \leqslant 0, \quad \forall i && \text{(2b)}\\
& h_j(\mathbf{x},\mathbf{d}) \leqslant 0, \quad \forall j && \text{(2c)}
\end{aligned}
$$

- Varying input parameters **d**: demand, renewable power generation
- Continuous variables **x**: power dispatches, power flows through lines
- Binary variables **y**: on/off status of the generating units
- Objective function (2a) minimizes the total generation costs
- Equation (2b): technical constraints of generating units
- Equation (2c): technical constraints of network
- Even if all functions are linear, problem (2a)-(2c) is **NP-hard**

Application 4: learning unit commitment

We have access to a set of historical data including:

- Input parameters $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$
- Optimal continous decisions $\{\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_N^*\}$
- Optimal binary decisions $\{\mathbf{y}_1^*, \mathbf{y}_2^*, \ldots, \mathbf{y}_N^*\}$

Application 4: learning unit commitment

KNN + LP approach. For a new input vector $\tilde{\mathbf{d}}$ do:

- Among $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$, find the K nearest neigborst to $\tilde{\mathbf{d}}$.
- For each neighbor k do
  - Fix binary variables to $\mathbf{y}_k^*$
  - Solve the linear program

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad f(\mathbf{x}, \mathbf{y}_k^*) \tag{3a}$$

$$g_i(\mathbf{x}, \mathbf{y}_k^*) \leqslant 0, \quad \forall i \tag{3b}$$

$$h_j(\mathbf{x}, \tilde{\mathbf{d}}) \leqslant 0, \quad \forall j \tag{3c}$$

  - Denote the optimal solution and optimal value as $\mathbf{x}_k^L$ and z_k^L
- Choose the neighbor with minimum cost $\tilde{k} \in \arg\min_k \{z_k^L\}$
- Provide the optimal solution $\mathbf{x}_{\tilde{k}}^L$ and $\mathbf{y}_{\tilde{k}}^*$

Application 4: learning unit commitment

- KNN + LP results
- 500 instances, leave-one-out
- $K = 50$
- MIP gap = 0.01%

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

Application 4: learning unit commitment

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.

Application 4: learning unit commitment

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.

Application 4: learning unit commitment

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.

Application 4: learning unit commitment

KNN + MIP approach. For a new input vector $\tilde{\mathbf{d}}$ do:

- Among $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$, find the K nearest neigborst to $\tilde{\mathbf{d}}$.
- Set upper bound of $\mathbf{y}$ to $\lceil \frac{1}{K} \sum_k \mathbf{y}_k^* \rceil$.
- Set lower bound of $\mathbf{y}$ to $\lfloor \frac{1}{K} \sum_k \mathbf{y}_k^* \rfloor$.
- Solve the mixed-integer linear program

$$
\begin{aligned}
\min_{\mathbf{x} \in \mathbb{R}^n,\, \mathbf{y} \in \{0,1\}^m} \quad & f(\mathbf{x}, \mathbf{y}) && \text{(4a)} \\
& g_i(\mathbf{x}, \mathbf{y}) \leqslant 0, \quad \forall i && \text{(4b)} \\
& h_j(\mathbf{x}, \tilde{\mathbf{d}}) \leqslant 0, \quad \forall j && \text{(4c)} \\
& \lfloor \frac{1}{K} \sum_k \mathbf{y}_k^* \rfloor \leqslant \mathbf{y} \leqslant \lceil \frac{1}{K} \sum_k \mathbf{y}_k^* \rceil && \text{(4d)}
\end{aligned}
$$

Application 4: learning unit commitment

KNN + MIP results

System	Av. error (%)	Max error (%)	$< 0.01\%$	# Infes	Speedup
1888rte	-0.0004	0.0141	499	0	30.7x
1951rte	0.0002	0.0227	499	0	7.7x
2848rte	0.0004	0.0228	497	0	20.0x
3012wp	-0.0001	0.0199	497	0	15.4x
3375wp	0.0001	0.0198	494	0	14.6x
6468rte	-0.0002	0.0217	499	0	18.9x
6470rte	0.0007	0.0174	499	0	15.9x
6495rte	-0.0001	0.0086	500	0	8.5x
6515rte	0.0003	0.0151	498	0	26.5x

- Most cases below GAP
- No infeasible cases
- Speedup factor between 7.7x and 30.7x

Questions we want to address

- Are power system problems so complex that any fortuitous learning-based method may involve significant computational savings?

  Yes, specially in combinatorial problems like UC

- Do simpler and interpretable learning-based methods perform similarly to complex black-box methods?

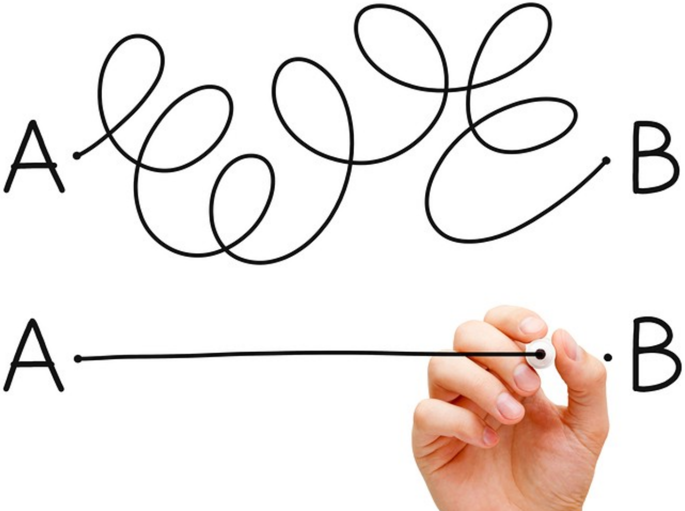
  Yes, depending on problem structure and data

- Should black-box methods be benchmarked against simpler methods in power system applications?

  Yes. Otherwise, irrelevant publications will continue

Takeaway message

In many cases the simplicity of less is more



Thanks for the attention!

Questions?

OASYS
Optimization and Analytics for
Sustainable energY Systems

More info: oasys.uma.es
Email: spineda@uma.es