Data-Driven Screening of Network Constraints for Unit Commitment

INFORMS Annual Meeting 2020

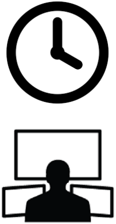
S. Pineda J. M. Morales A. Jiménez-Cordero

OASYS group, University of Málaga (Spain)

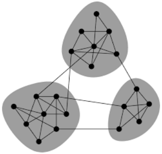
November 11, 2020

What problem are we solving?

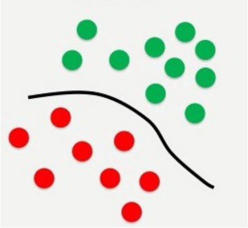
Short-term decisions



Spatial dimension



Classification



What problem are we solving?

Unit Commitment Problem	
Horizon	1 day
Decisions	Generation commitment Generation dispach Power flows
Objective	Min production cost
Constraints	Generation = Demand Unit technical limits Line technical limits
Comput. burden	High

How is that problem formulated?

$$\min_{\mathbf{x}\in\mathbb{R}^n,\, \mathbf{y}\in\{0,1\}^m} \quad f(\mathbf{x},\mathbf{y}) \tag{1a}$$

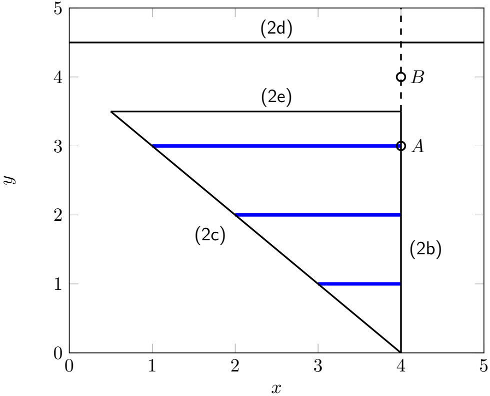
$$g_i(\mathbf{x},\mathbf{y}) \leqslant 0, \quad \forall i \tag{1b}$$

$$h_j(\mathbf{x}) \leqslant 0, \quad \forall j \tag{1c}$$

- Continuous variables $\mathbf{x}$: power dispatches, power flows through lines
- Binary variables $\mathbf{y}$: on/off status of the generating units
- Objective function (1a) minimizes the total generation costs
- Equation (1b): technical constraints of generating units
- Equation (1c): technical constraints of network
- Even if all functions are linear, problem (1a)-(1c) is **NP-hard**

Can we remove constraints to reduce time?

$$
\begin{aligned}
\max_{x\in\mathbb{R},y\in\mathbb{Z}} \quad & x + y && \text{(2a)}\\
\text{s.t.} \quad & x \leqslant 4 && \text{(2b)}\\
& x + y \geqslant 4 && \text{(2c)}\\
& y \leqslant 4.5 && \text{(2d)}\\
& y \leqslant 3.5 && \text{(2e)}
\end{aligned}
$$



- Constraint (2b) is an *active constraint*
- Constraint (2c) is an *inactive constraint*
- Constraint (2d) is a *redundant constraint*
- Constraint (2e) is defined as *quasi-active constraint*

How is the Unit Commitment problem formulated?

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

$$\min_{p_g, u_g, q_n, \epsilon_n} \quad \sum_g c_g p_g + L \sum_n |\epsilon_n| \tag{3a}$$

$$\text{s.t.} \quad q_n + \epsilon_n = \sum_{g:b_g=n} p_g - d_n, \forall n \tag{3b}$$

$$\sum_n q_n = 0 \tag{3c}$$

$$u_g \underline{p}_g \leqslant p_g \leqslant u_g \rho_g \overline{p}_g, \forall g \tag{3d}$$

$$-\overline{f}_l \leqslant \sum_n a_{ln} q_n \leqslant \overline{f}_l, \forall l \tag{3e}$$

$$u_g \in \{0, 1\}, \forall g \tag{3f}$$

We compare 8 different methods to remove constraints (3e)

Which methods can be used to remove constraints?

Benchmark

- No network constraints are removed (Extremely high time)

Single-bus

- All network constraints are removed (Very fast)
- Close-to-optimal solutions in low-congested systems
- Highly suboptimal solutions in general

Perfect information

- Removes all constraints not binding at the optimum
- It cannot be implemented in practice
- It removes quasi-active constraints

Naive

- It removes line constraints that have not been congested in the past
- It requires access to historical data
- Low number of removed constraints

Which methods can be used to remove constraints?

Constraint generation (CG)

- It starts by solving the UC without any network constraint
- Line constraints exceeding their capacity are iteratively added
- It provides the same solution as BN
- High computational burden since the UC is solved at each iteration

Which methods can be used to remove constraints?

Roald method (RO)[1]

- Two optimization problems for each line are solved

$$
\begin{aligned}
\min_{p_g, q_n, d_n} / \max_{p_g, q_n, d_n} \quad & \sum_n a_{l'n} q_n \\
\text{s.t.} \quad & \text{(3b), (3c), (3d), (3e)} \\
& \underline{d}_n \leqslant d_n \leqslant \overline{d}_n, \forall n
\end{aligned}
$$

- If the objective functions reach the line limit, then its capacity constraints are kept. Otherwise, such constraints are removed.

- It only removes redundant constraints

[1] Roald and Molzahn 2019.

Which methods can be used to remove 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

Which methods can be used to remove constraints?

Data-driven method (DD)

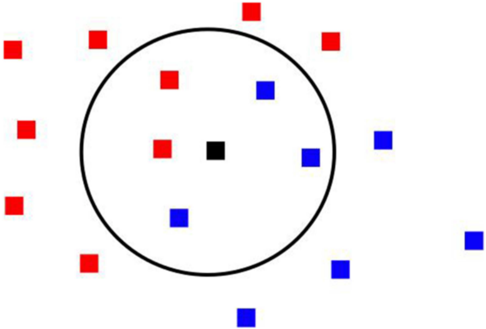
- Data on net demand and congestion status is denoted as $(\tilde{\mathbf{d}}_t, s_{lt}) \forall t$
- For a new time period $\hat{t}$, find the subset of closest K neighbors $(\mathcal{N}_K)$ using the distance function below

$$\text{dist}(\tilde{\mathbf{d}}_t, \tilde{\mathbf{d}}_{\hat{t}}) = \|a_l^T(\tilde{\mathbf{d}}_t - \tilde{\mathbf{d}}_{\hat{t}})\|_2 \tag{5}$$

- Two individuals are close if the net demand of those buses that have a higher impact on the power flow through line l are similar enough
- If $s_{lt} = 0\ \forall t \in \mathcal{N}_K$, line l is assumed uncongested for $\hat{t}$ and its capacity constraints are removed. Otherwise, such constraints are kept.

Which methods can be used to remove constraints?

Data-driven method (DD)



■ → Line is congested

■ → Line is not congested

■ → Line is assumed to be congested and its constraint is kept

Which methods can be used to remove constraints?

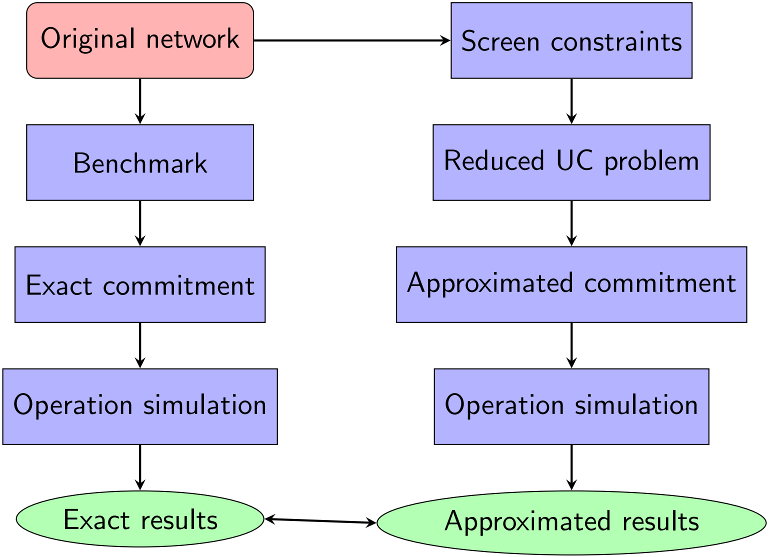
Data-driven + constraint generation (DD+CG)

- Use data to rapidly remove a large number of constraints
- Then iteratively add violated line constraints
- It also provides the same solution as BN
- It requires way less iterations than CG

Have you tried it on a realistic case study?

- IEEE RTS-96 test system modified to accommodate 19 wind farms
- 73 nodes and 120 transmission lines
- 300 training days and 60 test days
- We consider a low- and a high-congested case

What about the results?



What about the results?

Low-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100
Single-bus	100	-0.04	0.044	19.8
Perfect	99.9	0.01	0.001	15.9
Naive	99.2	0.00	0.000	16.5
ConGen	99.9	0.00	0.000	27.0
Roald	85.8	0.00	0.000	24.8
Data-Driven	99.8	0.00	0.000	17.5
DD+CG	99.8	0.00	0.000	22.6

What about the results?

Low-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100
Single-bus	100	-0.04	0.044	19.8
Perfect	99.9	0.01	0.001	15.9
Naive	99.2	0.00	0.000	16.5
ConGen	99.9	0.00	0.000	27.0
Roald	85.8	0.00	0.000	24.8
Data-Driven	99.8	0.00	0.000	17.5
DD+CG	99.8	0.00	0.000	22.6

Single-bus method provides acceptable results

What about the results?

Low-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100
Single-bus	100	-0.04	0.044	19.8
Perfect	99.9	0.01	0.001	15.9
Naive	99.2	0.00	0.000	16.5
ConGen	99.9	0.00	0.000	27.0
Roald	85.8	0.00	0.000	24.8
Data-Driven	99.8	0.00	0.000	17.5
DD+CG	99.8	0.00	0.000	22.6

Five methods provide the same solution than the benchmark

What about the results?

Low-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100
Single-bus	100	-0.04	0.044	19.8
Perfect	99.9	0.01	0.001	15.9
Naive	99.2	0.00	0.000	16.5
ConGen	99.9	0.00	0.000	27.0
Roald	85.8	0.00	0.000	24.8
Data-Driven	99.8	0.00	0.000	17.5
DD+CG	99.8	0.00	0.000	22.6

Roald is conservative and keeps 14% of line constraints

What about the results?

Low-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100
Single-bus	100	-0.04	0.044	19.8
Perfect	99.9	0.01	0.001	15.9
Naive	99.2	0.00	0.000	16.5
ConGen	99.9	0.00	0.000	27.0
Roald	85.8	0.00	0.000	24.8
Data-Driven	99.8	0.00	0.000	17.5
DD+CG	99.8	0.00	0.000	22.6

Naive and Data-Driven achieve the highest time reduction

What about the results?

High-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100.0
Single-bus	100	-20.14	10.557	3.1
Perfect	95.1	1.65	0.435	5.7
Naive	72.5	0.01	0.001	33.3
ConGen	83.2	0.00	0.000	65.4
Roald	21.7	0.00	0.000	80.4
Data-Driven	89.4	0.79	0.107	12.6
DD+CG	88.8	0.00	0.000	32.7

What about the results?

High-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100.0
Single-bus	100	-20.14	10.557	3.1
Perfect	95.1	1.65	0.435	5.7
Naive	72.5	0.01	0.001	33.3
ConGen	83.2	0.00	0.000	65.4
Roald	21.7	0.00	0.000	80.4
Data-Driven	89.4	0.79	0.107	12.6
DD+CG	88.8	0.00	0.000	32.7

Single-bus approach is fast but provides catastrophic results

What about the results?

High-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100.0
Single-bus	100	-20.14	10.557	3.1
Perfect	95.1	1.65	0.435	5.7
Naive	72.5	0.01	0.001	33.3
ConGen	83.2	0.00	0.000	65.4
Roald	21.7	0.00	0.000	80.4
Data-Driven	89.4	0.79	0.107	12.6
DD+CG	88.8	0.00	0.000	32.7

Perfect provides suboptimal results due to quasi-active constraints

What about the results?

High-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100.0
Single-bus	100	-20.14	10.557	3.1
Perfect	95.1	1.65	0.435	5.7
Naive	72.5	0.01	0.001	33.3
ConGen	83.2	0.00	0.000	65.4
Roald	21.7	0.00	0.000	80.4
Data-Driven	89.4	0.79	0.107	12.6
DD+CG	88.8	0.00	0.000	32.7

ConGen removes a lot of constraints but requires high time

What about the results?

High-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100.0
Single-bus	100	-20.14	10.557	3.1
Perfect	95.1	1.65	0.435	5.7
Naive	72.5	0.01	0.001	33.3
ConGen	83.2	0.00	0.000	65.4
Roald	21.7	0.00	0.000	80.4
Data-Driven	89.4	0.79	0.107	12.6
DD+CG	88.8	0.00	0.000	32.7

Roald only removes 22% of constraints and limits time reduction

What about the results?

High-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100.0
Single-bus	100	-20.14	10.557	3.1
Perfect	95.1	1.65	0.435	5.7
Naive	72.5	0.01	0.001	33.3
ConGen	83.2	0.00	0.000	65.4
Roald	21.7	0.00	0.000	80.4
Data-Driven	89.4	0.79	0.107	12.6
DD+CG	88.8	0.00	0.000	32.7

Data-Driven removes most constraints but involves small infeasibilities

What about the results?

High-congested case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0	0.00	0.000	100.0
Single-bus	100	-20.14	10.557	3.1
Perfect	95.1	1.65	0.435	5.7
Naive	72.5	0.01	0.001	33.3
ConGen	83.2	0.00	0.000	65.4
Roald	21.7	0.00	0.000	80.4
Data-Driven	89.4	0.79	0.107	12.6
DD+CG	88.8	0.00	0.000	32.7

DD+CG provides the optimum and significantly reduces time

What about the results?

2000-bus case

Method	Removed(%)	Δcost(%)	Infes(%)	Time(%)
Benchmark	0.0	0.00	0.00	100.0
Single-bus	100.0	-2.17	0.26	0.4
Perfect	99.7	-0.22	0.13	1.0
Naive	92.3	0.00	0.00	10.6
ConGen	98.8	0.00	0.00	8.9
Roald	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

- Time of Data-Driven similar to Single-bus and Perfect
- Solution provided by Data-driven involve tiny inaccuracies
- DD+CG recovers the original solution at lowest time

Conclusions

Method	# Removed	Original solution	Time
Benchmark	🔴	🟢	⌛ ⌛ ⌛
Single-bus	🟢	🔴	⌛
Perfect	🟢	🔴	⌛
Naive	🟠	🟠	⌛
ConGen	🟢	🟢	⌛ ⌛
Roald	🟠	🟢	⌛ ⌛
Data-Driven	🟢	🟠	⌛
DD + CG	🟢	🟢	⌛

Thanks for the attention!

Questions?

OASYS
Optimization and Analytics for Sustainable energY Systems

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