Learning-Assisted Optimization for Transmission Switching

EURO 2024 (July 1st 2024)

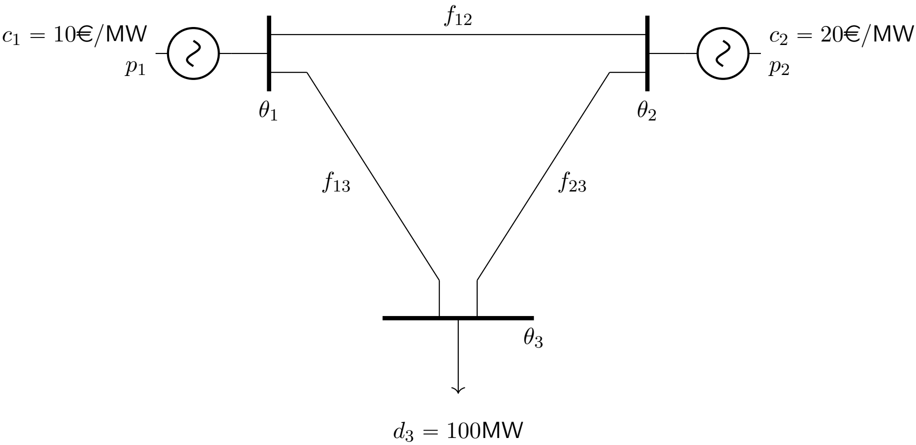
Salvador Pineda

(joint work with J. M. Morales, A. Jiménez-Cordero)

OASYS group, University of Málaga (Spain)

Motivating example

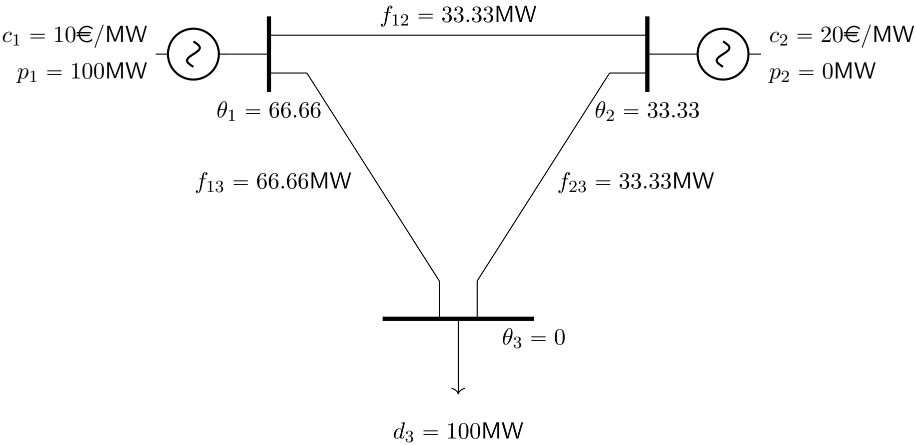
Optimal power flow (OPF): Determine the power generation and power flows to satisfy the demand at the minimum cost



The cheap unit satisfies the demand

Motivating example

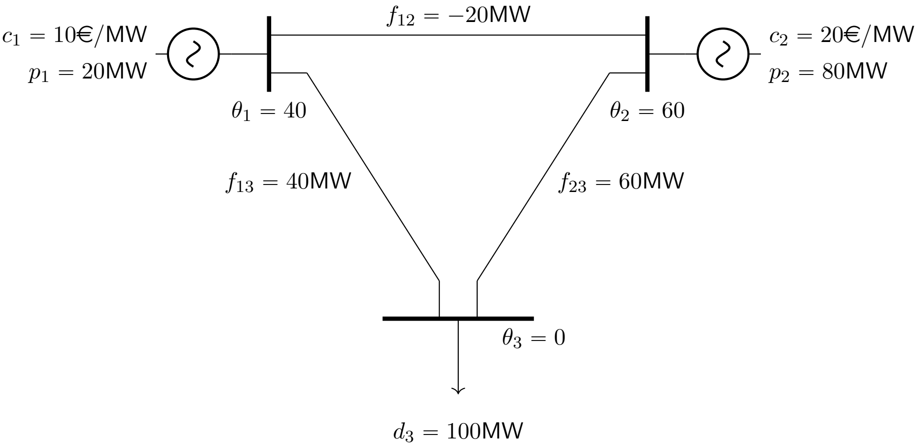
Optimal solution: generate all with cheapest unit (cost = 1000€)



Electrons are not potatoes!!!

Motivating example

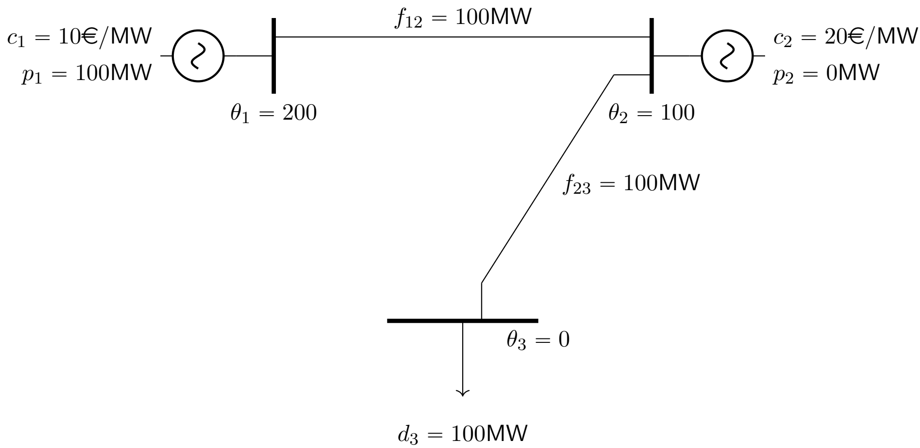
If $f_{13} \leqslant 40$, the expensive unit also generates (cost=1800€)



Network limits increate the cost!!!

Motivating example

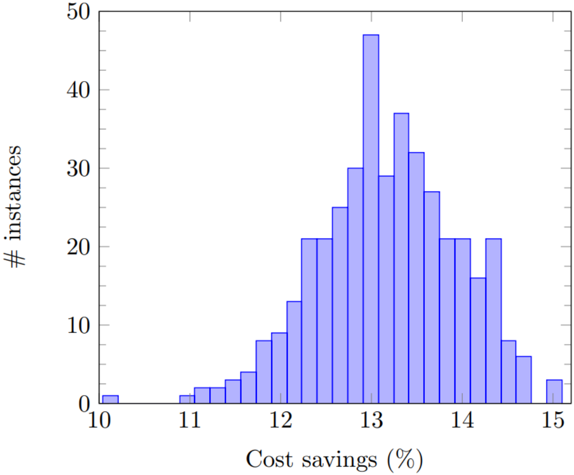
If line 13 is disconnected, cost$=$ 1000€



Disconnecting lines can reduce cost!!!

Motivating example

In the 118-bus system, the average cost saving is 13.2%



Formulation

The optimal power flow (OPF) is formulated as a linear optimization problem

$$
\begin{aligned}
\min_{p_i, f_{ij}, \theta_i} \quad & \sum_i c_i \, p_i && \text{(1a)} \\
\text{s.t.} \quad & \sum_{(i,j)\in\mathcal{L}_i^-} f_{ij} - \sum_{(i,j)\in\mathcal{L}_i^+} f_{ij} = p_i - d_i, \quad \forall i && \text{(1b)} \\
& f_{ij} = b_{ij}(\theta_i - \theta_j), \quad \forall (i,j) \in \mathcal{L} && \text{(1c)} \\
& \underline{p}_i \leqslant p_i \leqslant \overline{p}_i, \quad \forall i && \text{(1d)} \\
& -\underline{f}_{ij} \leqslant f_{ij} \leqslant \overline{f}_{ij}, \quad \forall (i,j) \in \mathcal{L} && \text{(1e)}
\end{aligned}
$$

Formulation

The optimal transmission switching (OTS) requires binary variables x_{ij} and is formulated as a mixed-integer non-linear problem ...

$$\min_{p_i, f_{ij}, \theta_i, x_{ij}} \quad \sum_i c_i \, p_i \tag{2a}$$

$$\text{s.t.} \quad \sum_{(i,j) \in \mathcal{L}_i^-} f_{ij} - \sum_{(i,j) \in \mathcal{L}_i^+} f_{ij} = p_i - d_i, \quad \forall i \tag{2b}$$

$$f_{ij} = x_{ij} b_{ij} (\theta_i - \theta_j), \quad \forall (i,j) \in \mathcal{L} \tag{2c}$$

$$\underline{p}_i \leqslant p_i \leqslant \overline{p}_i, \quad \forall i \tag{2d}$$

$$- x_{ij} \underline{f}_{ij} \leqslant f_{ij} \leqslant x_{ij} \overline{f}_{ij}, \quad \forall (i,j) \in \mathcal{L} \tag{2e}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in \mathcal{L} \tag{2f}$$

... that can be directly solved using optimization solvers such as Gurobi.

Formulation

To avoid the non-linear terms in

$$f_{ij} = x_{ij} b_{ij}(\theta_i - \theta_j)$$

We replace it by

$$b_{ij}(\theta_i - \theta_j) - \overline{M}_{ij}(1 - x_{ij}) \leqslant f_{ij} \leqslant b_{ij}(\theta_i - \theta_j) - \underline{M}_{ij}(1 - x_{ij})$$

Together with equation

$$-x_{ij}\underline{f}_{ij} \leqslant f_{ij} \leqslant x_{ij}\overline{f}_{ij}$$

We have that:

- If $x_{ij} = 1 \Rightarrow b_{ij}(\theta_i - \theta_j) \leqslant f_{ij} \leqslant b_{ij}(\theta_i - \theta_j)$ and $-\underline{f}_{ij} \leqslant f_{ij} \leqslant \overline{f}_{ij}$

- If $x_{ij} = 0 \Rightarrow f_{ij} = 0$ and $\underline{M}_{ij} \leqslant b_{ij}(\theta_i - \theta_j) \leqslant \overline{M}_{ij}$

Formulation

The OTS is reformulated as a mixed-integer linear problem

$$\min_{p_i, f_{ij}, \theta_i, x_{ij}} \quad \sum_i c_i \, p_i \tag{3a}$$

$$\text{s.t.} \quad \sum_{(i,j) \in \mathcal{L}_i^-} f_{ij} - \sum_{(i,j) \in \mathcal{L}_i^+} f_{ij} = p_i - d_i, \quad \forall i \tag{3b}$$

$$b_{ij}(\theta_i - \theta_j) - \overline{M}_{ij}(1 - x_{ij}) \leqslant f_{ij}, \quad \forall (i,j) \in \mathcal{L} \tag{3c}$$

$$f_{ij} \leqslant b_{ij}(\theta_i - \theta_j) - \underline{M}_{ij}(1 - x_{ij}), \quad \forall (i,j) \in \mathcal{L} \tag{3d}$$

$$\underline{p}_i \leqslant p_i \leqslant \overline{p}_i, \quad \forall i \tag{3e}$$

$$-x_{ij} \underline{f}_{ij} \leqslant f_{ij} \leqslant x_{ij} \overline{f}_{ij}, \quad \forall (i,j) \in \mathcal{L} \tag{3f}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in \mathcal{L} \tag{3g}$$

$\underline{M}_{ij}$ and $\overline{M}_{ij}$ must be valid bounds for $b_{ij}(\theta_i - \theta_j)$ if line is open

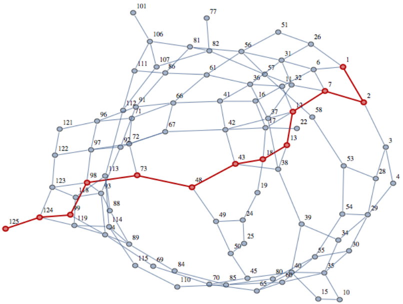
$\underline{M}_{ij}$ and $\overline{M}_{ij}$ must be small enough to avoid computational issues

Literature review

Fattahi et al. (2019) find a bound on $\overline{M}_{ij}^{\mathrm{OPT}}$ if there exists a connected spanning subgraph of the network with non-switchable lines

$$\overline{M}_{i'j'}^{\mathrm{OPT}} \leqslant b_{i'j'} \sum_{(k,l)\in SP_{i'j'}} \frac{\overline{f}_{kl}}{b_{kl}}$$

where $SP_{i'j'}$ is the shortest path of **connected lines** between nodes i' and j' (very easy to compute using Dijkstra's algorithm)



Literature review

Learning-based approaches to solve the OTS:

- Johnson et al. (2021): solve K linear problems fixing binary variables to those of nearest neighbors and choose the cheapest solution.
  - 👍 Fast and interpretable
  - 👎 Probably suboptimal
- Yang and Oren (2019); Han and Hill (2022); Bugaje et al. (2023) learn the line status using neural network.
  - 👍 Fast and precise
  - 👎 Not interpretable and hard to train

Is it possible to create a learning-assisted methodology that is fast and precise while remaining interpretable?

The proposed learning-assisted methodology reduce the computational burden of the OTS focusing on:

- Fixing some binary variables
- Finding tighter big-M values

Methodology

We compare the following approaches:

- Bench: set big-M as Fattahi et al. (2019) and solve MIP.

Methodology

We compare the following approaches:

- Bench: set big-M as Fattahi et al. (2019) and solve MIP.
- Direct: set binaries to rounded average of nearest neighbors.

Methodology

We compare the following approaches:

- Bench: set big-M as Fattahi et al. (2019) and solve MIP.
- Direct: set binaries to rounded average of nearest neighbors.
- Linear: solve K LPs fixing binaries to nearest neighbors.

Methodology

We compare the following approaches:

- Bench: set big-M as Fattahi et al. (2019) and solve MIP.
- Direct: set binaries to rounded average of nearest neighbors.
- Linear: solve K LPs fixing binaries to nearest neighbors.
- FatM: update shortest paths based on unanimous vote of nearest neighbors to recompute Fattahi's big-M. Binaries are not fixed.

Methodology

We compare the following approaches:

- Bench: set big-M as Fattahi et al. (2019) and solve MIP.
- Direct: set binaries to rounded average of nearest neighbors.
- Linear: solve K LPs fixing binaries to nearest neighbors.
- FatM: update shortest paths based on unanimous vote of nearest neighbors to recompute Fattahi's big-M. Binaries are not fixed.
- FixB: fix binaries based on unanimous vote of nearest neighbors.

Methodology

We compare the following approaches:

- Bench: set big-M as Fattahi et al. (2019) and solve MIP.
- Direct: set binaries to rounded average of nearest neighbors.
- Linear: solve K LPs fixing binaries to nearest neighbors.
- FatM: update shortest paths based on unanimous vote of nearest neighbors to recompute Fattahi's big-M. Binaries are not fixed.
- FixB: fix binaries based on unanimous vote of nearest neighbors.
- FixB-FatM: a combination of the two previous approaches.

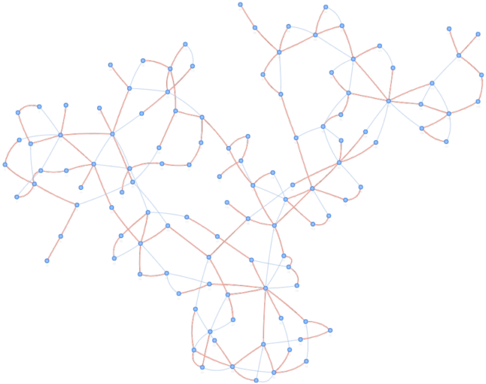
Methodology

We compare the following approaches:

- Bench: set big-M as Fattahi et al. (2019) and solve MIP.
- Direct: set binaries to rounded average of nearest neighbors.
- Linear: solve K LPs fixing binaries to nearest neighbors.
- FatM: update shortest paths based on unanimous vote of nearest neighbors to recompute Fattahi's big-M. Binaries are not fixed.
- FixB: fix binaries based on unanimous vote of nearest neighbors.
- FixB-FatM: a combination of the two previous approaches.
- AngM: big-M are set to maximum/minimum angle differences of all observed data. A security factor > 1 can be used.

Numerical simulations

- 118-bus system with 186-lines (69 switchable lines)
- 500 instances with different demands ($\pm 10\%$)
- Gurobi with mipgap at 0.01% and maximum time 1 hour



Numerical simulations

Approach	K	# opt	# sub	# inf	gap-ave	gap-max	time (s)
Bench	-	-	-	-	-	-	145.00
Direct	50	0	500	0	2.06%	14.14%	0.00
Direct	499	0	500	0	2.63%	8.38%	0.00
Linear	50	51	449	0	0.08%	1.06%	0.04
Linear	499	127	373	0	0.04%	0.71%	0.39
FatM	50	500	0	0	-	-	109.95
FixB	50	500	0	0	-	-	16.39
FixB-FatM	50	500	0	0	-	-	12.33
$1\times$*AngM*	499	495	5	0	0.002%	0.39%	0.70
$1.1\times$*AngM*	499	500	0	0	-	-	0.78

- K: number of nearest neighbors.
- # opt: number of optimal instances.
- # sub: number of suboptimal instances.
- # inf: number of infeasible instances.
- gap-ave, gap-max: average and maximum gap compared to *Bench*.
- time: average time of 500 instances (in seconds).

Numerical simulations

Approach	K	# opt	# sub	# inf	gap-ave	gap-max	time (s)
Bench	-	-	-	-	-	-	145.00
Direct	50	0	500	0	2.06%	14.14%	0.00
Direct	499	0	500	0	2.63%	8.38%	0.00
Linear	50	51	449	0	0.08%	1.06%	0.04
Linear	499	127	373	0	0.04%	0.71%	0.39
FatM	50	500	0	0	-	-	109.95
FixB	50	500	0	0	-	-	16.39
FixB-FatM	50	500	0	0	-	-	12.33
$1\times$ *AngM*	499	495	5	0	0.002%	0.39%	0.70
$1.1\times$ *AngM*	499	500	0	0	-	-	0.78

- The average time of *Bench* amount to 145s.
- 12 instances are not solved to global optimality in one hour.

Numerical simulations

Approach	K	# opt	# sub	# inf	gap-ave	gap-max	time (s)
Bench	-	-	-	-	-	-	145.00
Direct	50	0	500	0	2.06%	14.14%	0.00
Direct	499	0	500	0	2.63%	8.38%	0.00
Linear	50	51	449	0	0.08%	1.06%	0.04
Linear	499	127	373	0	0.04%	0.71%	0.39
FatM	50	500	0	0	-	-	109.95
FixB	50	500	0	0	-	-	16.39
FixB-FatM	50	500	0	0	-	-	12.33
$1\times$*AngM*	499	495	5	0	0.002%	0.39%	0.70
$1.1\times$*AngM*	499	500	0	0	-	-	0.78

- *Direct* is really fast but suboptimal.
- *Linear* gets the optimal solution in some instances.
- *Linear* with $K = 499$ can be competitive.

Numerical simulations

Approach	K	# opt	# sub	# inf	gap-ave	gap-max	time (s)
Bench	-	-	-	-	-	-	145.00
Direct	50	0	500	0	2.06%	14.14%	0.00
Direct	499	0	500	0	2.63%	8.38%	0.00
Linear	50	51	449	0	0.08%	1.06%	0.04
Linear	499	127	373	0	0.04%	0.71%	0.39
FatM	50	500	0	0	-	-	109.95
FixB	50	500	0	0	-	-	16.39
FixB-FatM	50	500	0	0	-	-	12.33
$1\times$*AngM*	499	495	5	0	0.002%	0.39%	0.70
$1.1\times$*AngM*	499	500	0	0	-	-	0.78

- *FatM* does not solve all instances within one hour, but average time is lower than *Bench*.
- *FixB* solves all instances within one hour and significantly reduces computational time.
- *FixB-FatM* yields the best results.

Numerical simulations

Approach	K	# opt	# sub	# inf	gap-ave	gap-max	time (s)
Bench	-	-	-	-	-	-	145.00
Direct	50	0	500	0	2.06%	14.14%	0.00
Direct	499	0	500	0	2.63%	8.38%	0.00
Linear	50	51	449	0	0.08%	1.06%	0.04
Linear	499	127	373	0	0.04%	0.71%	0.39
FatM	50	500	0	0	-	-	109.95
FixB	50	500	0	0	-	-	16.39
FixB-FatM	50	500	0	0	-	-	12.33
$1\times$*AngM*	499	495	5	0	0.002%	0.39%	0.70
$1.1\times$*AngM*	499	500	0	0	-	-	0.78

- Learning big-M values using past data significantly reduces time.
- The security factor reduces suboptimality without affecting time.

Numerical simulations

Approach	K	# opt	# sub	# inf	gap-ave	gap-max	time (s)
Bench	-	-	-	-	-	-	145.00
Linear	499	127	373	0	0.04%	0.71%	0.39
FixB-FatM	50	500	0	0	-	-	12.33
$1.1\times$ *AngM*	499	500	0	0	-	-	0.78

- If optimality is not crucial, *Linear* is the fastest approach.
- *AngM* with security factor solves all instances to optimality with a slight increase in computational time.

Numerical simulations

What if we increase demand variability from $\pm 10\%$ to $\pm 20\%$?

Approach	K	# opt	# sub	# inf	gap-ave	gap-max	time (s)
Bench	-	-	-	-	-	-	510.90
Linear	499	44	456	0	0.153%	1.50%	0.33
FixB-FatM	50	496	4	0	0.002%	0.72%	115.02
$1.1\times$ *AngM*	499	499	1	0	0.000%	0.02%	1.93

- Average time of *Bench* increases.
- Suboptimal instances of *Linear* increase.
- Computational time of *FixB-FatM* increases.
- *AngM* is fast (265x speedup) and precise (only 1 suboptimal).

Numerical simulations

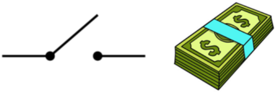
What if we use normal distributions with correlation?

	K	# opt	# sub	# inf	gap-ave	gap-max	time (s)
Bench	-	-	-	-	-	-	289.50
Linear	499	488	12	0	0.001%	0.11%	0.41
FixB-FatM	50	499	1	0	0.000%	0.17%	0.57
$1.1\times$ *AngM*	499	500	0	0	-	-	0.29

- *AngM* outperforms the other approaches in precision and time.

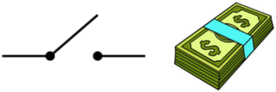
Conclusions

- The optimal transmission switching (OTS) determines the lines that can be disconnected to reduce the operating cost.



Conclusions

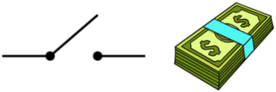
- The optimal transmission switching (OTS) determines the lines that can be disconnected to reduce the operating cost.



- The OTS is formulated as a mixed-integer linear problem with bigMs that is computationally difficult to solve.

Conclusions

- The optimal transmission switching (OTS) determines the lines that can be disconnected to reduce the operating cost.



- The OTS is formulated as a mixed-integer linear problem with bigMs that is computationally difficult to solve.

- We propose a learning-assisted approach to find tight bigMs and reduce the computational burden of the OTS.

Thanks for the attention!! Questions??

Optimization and Analytics for Sustainable energY Systems

More info: oasys.uma.es

Email: spineda@uma.es

References

Al-Amin B. Bugaje, Jochen L. Cremer, and Goran Strbac. Real-time transmission switching with neural networks. *IET Generation, Transmission & Distribution*, 17(3):696–705, 2023. doi: https://doi.org/10.1049/gtd2.12698. URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/gtd2.12698.

Salar Fattahi, Javad Lavaei, and Alper Atamtürk. A bound strengthening method for optimal transmission switching in power systems. *IEEE Transactions on Power Systems*, 34(1):280–291, 2019.

Tong Han and David Hill. Learning-based topology optimization of power networks. *IEEE Transactions on Power Systems*, 2022.

Emma S. Johnson, Shabbir Ahmed, Santanu S. Dey, and Jean-Paul Watson. A k-nearest neighbor heuristic for real-time dc optimal transmission switching, 2021. URL https://arxiv.org/abs/2003.10565.

Zhu Yang and Shmuel Oren. Line selection and algorithm selection for transmission switching by machine learning methods. In *2019 IEEE Milan PowerTech*, pages 1–6. IEEE, 2019.