Learning Convex Approximations for AC-OPF with Zero-Injection Feasibility Guarantees

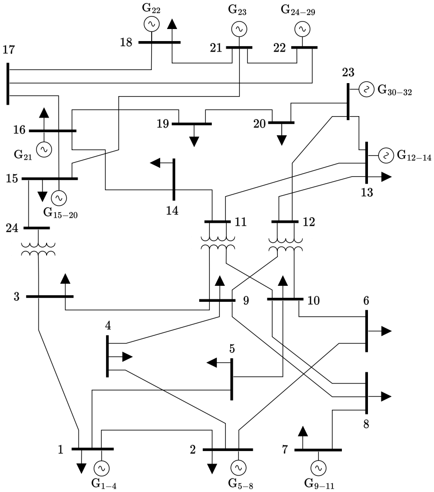
Gonzalo E. Constante Flores, Can Li

PURDUE UNIVERSITY

PSE Seminar 2023

Preliminaries

- The generating units and demands are distributed throughout a network, which is composed of *nodes* and *edges*.
- The matrix encoding the network information is known as *nodal admittance matrix*.
- Every node is characterized by a complex voltage and a complex net power injection.



AC Optimal Power Flow (AC-OPF)

- The AC-OPF (or some approximation thereof) is the cornerstone of the operation and planning of power systems and is generally solved multiple times a day by power system operators worldwide.
- Its goal is to determine the most economical production levels of generating units to supply the demand while satisfying physical and engineering constraints.
  - **Physical constraints** model the nonconvex governing physical laws, Ohm's law and Kirchhoff law, known as *power flow equations*.
  - **Engineering constraints** model voltage, angle difference, transmission, and generation limits.
- **In this talk**: Complex voltages in rectangular coordinates $\Rightarrow$ nonconvex QCQP.

AC Optimal Power Flow Formulation

The AC-OPF can be formulated as a nonconvex quadratically constrained optimization problem

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & Ax = b, \\ & g_i(x) \leq 0,\ i = 1, \ldots, m_2 \\ & x \in \mathbb{R}^n. \end{aligned}$$

- $f : \mathbb{R}^n \to \mathbb{R}$ the convex objective function
- $A \in \mathbb{R}^{n \times m_1}$ the constraint coefficient matrix
- $b \in \mathbb{R}^{m_1}$ the constraint coefficient vector
- $g : \mathbb{R}^n \to \mathbb{R}$ the quadratic (convex and nonconvex) constraints

The AC-OPF is known to be NP-hard [Bienstock et al., 2019].

Relaxations and approximations for AC-OPF

Substantial efforts have been devoted to finding tractable surrogates

Relaxations

- Provide lower bounds.
- Infeasibility certificates.
- Linear: Copper plate and Network flow [Coffrin, H. Hijazi, et al., 2016]
- Second-order conic: [Jabr, 2006; Kocuk et al., 2016]
- Quadratic convex: [Coffrin, H. L. Hijazi, et al., 2016]
- Semidefinite: [Bai et al., 2008]

Approximations

- Based on two ideas:
  1. Engineering assumptions (line parameters, voltage magnitudes, angle differences)
  2. Linearization/convexification points.
- Linear: LPAC [Coffrin and Hentenryck, 2014], IV-Flow [O'Neill et al., 2012; Castillo et al., 2016]
- Convex: SOC [Jabr, 2007], QPAC [Coffrin, H. Hijazi, et al., 2015], Our work!

For a comprehensive review, [Molzahn et al., 2019].

AC-OPF Reformulation

Reformulating the AC-OPF as a Difference-of-Convex-Functions Program

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & Ax = b, \\ & g_i(x) \le 0,\ i = 1, \dots, m_2 \\ & x \in \mathbb{R}^n \end{aligned} \quad \Longrightarrow \quad \begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & Ax = b, \\ & \hat{g}_i(x) - \check{g}_i(x) \le 0,\ i = 1, \dots, m_2, \\ & x \in \mathbb{R}^n, \end{aligned}$$

where $\hat{g}(x)$, and $\check{g}(x)$ are convex functions.

- The reformulated problem is still nonconvex due to $\check{g}(x)$.
- Convexify using a first-order Taylor series approximation [Yuille et al., 2003; Lipp et al., 2016].

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & Ax = b, \\ & \hat{g}(x) - \check{g}(\tilde{x}) - \nabla \check{g}(\tilde{x})^\top (x - \tilde{x}) \le 0, \\ & x \in \mathbb{R}^n. \end{aligned}$$

The QCAC Approximation

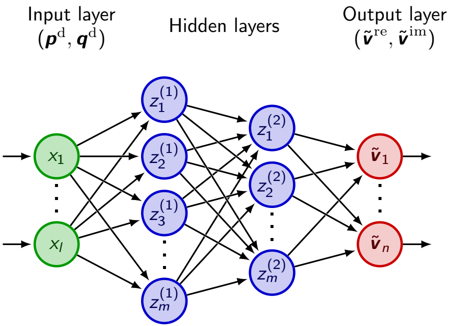
- A feasible convexification point, $\tilde{x}$, renders an inner convex approximation of the original problem.
- What if the convexification point, $\tilde{x}$, is not feasible?

$$
\begin{aligned}
\min_{x,s} \quad & f(x) + \lambda s \\
\text{s.t.} \quad & Ax = b, \\
& \hat{g}(x) - \check{g}(\tilde{x}) - \nabla \check{g}(\tilde{x})^\top (x - \tilde{x}) \leq s, \\
& x \in \mathbb{R}^n, \; s \in \mathbb{R}_{\geq 0},
\end{aligned}
$$

  where λ is a penalty term and s is a nonnegative slack variable.
- Note that no further assumptions are made to convexify the problem!
- Next, how can we predict good convexification points?
  We can leverage solutions of historical instances using *End-to-end learning*.

End-to-end learning

- Learning the mapping from the input parameters of an optimization problem to its solution.
- Given dataset $\{(\boldsymbol{x}_\ell, \boldsymbol{y}_\ell^*)\}_{\ell \in \mathcal{L}}$, where $\boldsymbol{y}_\ell^*$ denotes a solution to the optimization problem for the input $\boldsymbol{x}_\ell$.
- In the context of the AC-OPF: the input is the nodal demand vector and the output corresponds to the rectangular coordinates of the nodal voltages at the solution.
- **Main challenge**: Enforce constraints on the predictions!



Standard feed-forward NN

Enforcing constraints in neural networks

Soft methods

- Penalizing constraint violations
  - Augmented loss function
  - PINNs [Nellikkath et al., 2022]
  - Sensitivity-informed [Singh et al., 2022]
- Augmented Lagrangian methods
  - ALM [Fioretto et al., 2020]

Hard methods

- Implicit layers [Amos et al., 2017]
- Postprocessing [Zamzam et al., 2020; Li et al., 2022; Pan et al., 2023]
- Self-supervised [Donti et al., 2021; Chen et al., 2023]

What constraints do we want to enforce?

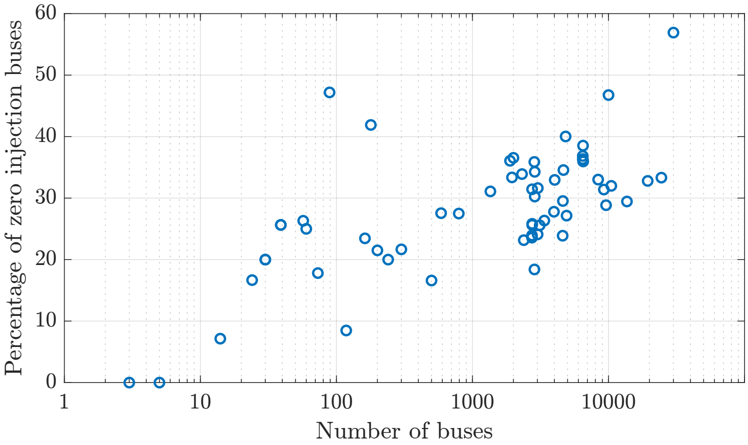
The relationship between current injections and nodal voltages, known as *Ohm's law*, is linear:

$$\begin{bmatrix} \boldsymbol{i}^{\mathrm{re}} \\ \boldsymbol{i}^{\mathrm{im}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G} & -\boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{G} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^{\mathrm{re}} \\ \boldsymbol{v}^{\mathrm{im}} \end{bmatrix}.$$

- However, some of the current injections, the ones with generation and/or demand, are unknown before solving the problem.
- There is a subset of nodes whose current injections are known a priori and equal to zero. Such nodes are called *zero-injection nodes*.
- This talk: A hard method to enforce Ohm's law of zero-injection nodes.

Zero-injection buses

Buses (nodes) without generation or demand.



Enforcing hard linear equality constraints

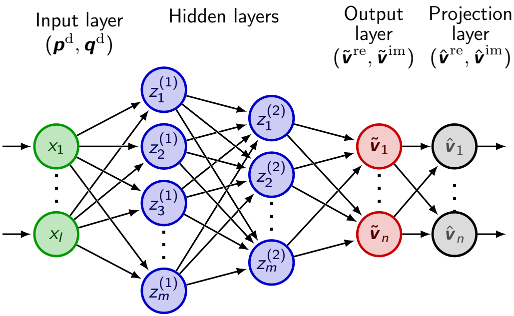
Can we enforce hard linear equality constraints using only an explicit layer?

- Explicit layers in feedforward NNs can be expressed as

$$\begin{bmatrix} \hat{\boldsymbol{v}}^{\mathrm{re}} \\ \hat{\boldsymbol{v}}^{\mathrm{im}} \end{bmatrix} = \sigma\left(\boldsymbol{W} \begin{bmatrix} \tilde{\boldsymbol{v}}^{\mathrm{re}} \\ \tilde{\boldsymbol{v}}^{\mathrm{im}} \end{bmatrix} + \boldsymbol{b} \right),$$

  where $\sigma(\cdot)$ denotes the nonlinear activation function.

- **Goal:** Find $\sigma(\cdot)$, $\boldsymbol{W}$, and $\boldsymbol{b}$ of the *projection layer* such that a set of linear equalities is satisfied during training and inference.



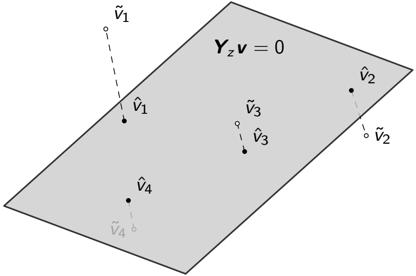
Proposed NN with projection layer

Projecting predicted variables

Orthogonal projection of the predicted variables $(\hat{\boldsymbol{v}}^{\mathrm{re}}, \hat{\boldsymbol{v}}^{\mathrm{im}})$ onto

$$\boldsymbol{Y}_z \begin{bmatrix} \boldsymbol{v}^{\mathrm{re}} \\ \boldsymbol{v}^{\mathrm{im}} \end{bmatrix} = 0,$$

where $\boldsymbol{Y}_z = \begin{bmatrix} \boldsymbol{G}_z & -\boldsymbol{B}_z \\ \boldsymbol{B}_z & \boldsymbol{G}_z \end{bmatrix}$.



Orthogonal projection of predicted voltages, $\tilde{\boldsymbol{v}}$, onto the nullspace of $\boldsymbol{Y}_z$.

Determining weights and biases

The orthogonal projection onto a linear set of equalities can be formulated as a quadratic problem

$$(\hat{\boldsymbol{v}}^{\mathrm{re}}, \hat{\boldsymbol{v}}^{\mathrm{im}}) \in \operatorname*{arg\,min}_{\boldsymbol{v}^{\mathrm{re}}, \boldsymbol{v}^{\mathrm{im}}} \quad \|\tilde{\boldsymbol{v}}^{\mathrm{re}} - \boldsymbol{v}^{\mathrm{re}}\|_2^2 + \|\tilde{\boldsymbol{v}}^{\mathrm{im}} - \boldsymbol{v}^{\mathrm{im}}\|_2^2$$

$$\text{s.t.} \quad \boldsymbol{Y}_z \begin{bmatrix} \boldsymbol{v}^{\mathrm{re}} \\ \boldsymbol{v}^{\mathrm{im}} \end{bmatrix} = 0.$$

Its closed-form solution is given by

$$\begin{bmatrix} \hat{\boldsymbol{v}}^{\mathrm{re}} \\ \hat{\boldsymbol{v}}^{\mathrm{im}} \end{bmatrix} = \boldsymbol{A}^* \begin{bmatrix} \tilde{\boldsymbol{v}}^{\mathrm{re}} \\ \tilde{\boldsymbol{v}}^{\mathrm{im}} \end{bmatrix},$$

where $\boldsymbol{A}^* = \mathbb{I} - \boldsymbol{Y}_z^\top \left(\boldsymbol{Y}_z \boldsymbol{Y}_z^\top\right)^{-1} \boldsymbol{Y}_z$.

Determining weights and biases

- The closed-form solution can be represented as an explicit layer

$$\begin{bmatrix} \hat{\boldsymbol{v}}^{\mathrm{re}} \\ \hat{\boldsymbol{v}}^{\mathrm{im}} \end{bmatrix} = \boldsymbol{A}^{*} \begin{bmatrix} \tilde{\boldsymbol{v}}^{\mathrm{re}} \\ \tilde{\boldsymbol{v}}^{\mathrm{im}} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \hat{\boldsymbol{v}}^{\mathrm{re}} \\ \hat{\boldsymbol{v}}^{\mathrm{im}} \end{bmatrix} = \sigma\left(\boldsymbol{W} \begin{bmatrix} \tilde{\boldsymbol{v}}^{\mathrm{re}} \\ \tilde{\boldsymbol{v}}^{\mathrm{im}} \end{bmatrix} + \boldsymbol{b} \right),$$

  where $\sigma(\cdot)$ is a linear activation function, $\boldsymbol{W} = \boldsymbol{A}^{*}$, and $\boldsymbol{b} = 0$.

- The matrix $\boldsymbol{A}^{*} = \mathbb{I} - \boldsymbol{Y}_z^{\top}\left(\boldsymbol{Y}_z \boldsymbol{Y}_z^{\top}\right)^{-1} \boldsymbol{Y}_z$, which corresponds to the weights of the projection layer, *only depends on the topology of the network* and is independent of the operating conditions. Hence, $\boldsymbol{A}^{*}$ is only computed once for training and inference.

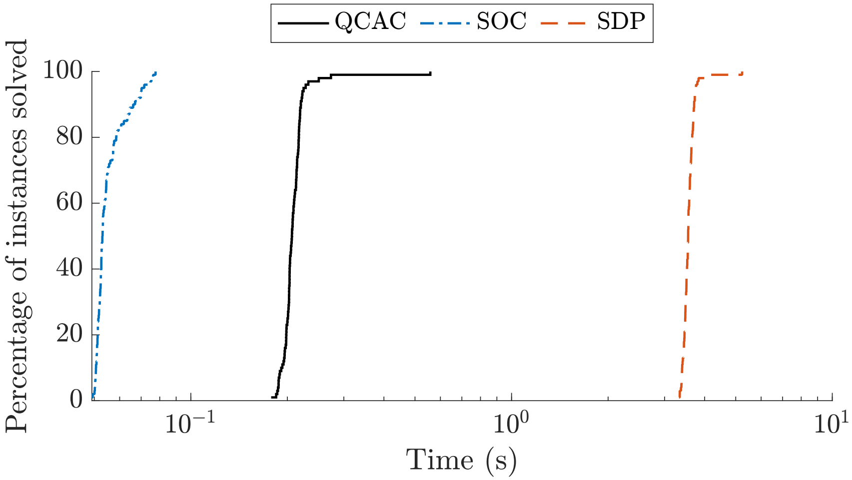
Numerical results

- Congested condition of the IEEE 118-bus system from the PGLib [Babaeinejadsarookolaee et al., 2021].
- 100 samples for random active and reactive power demands, ±40% and ±15% from the base case, respectively.

Table: Optimality gap comparison

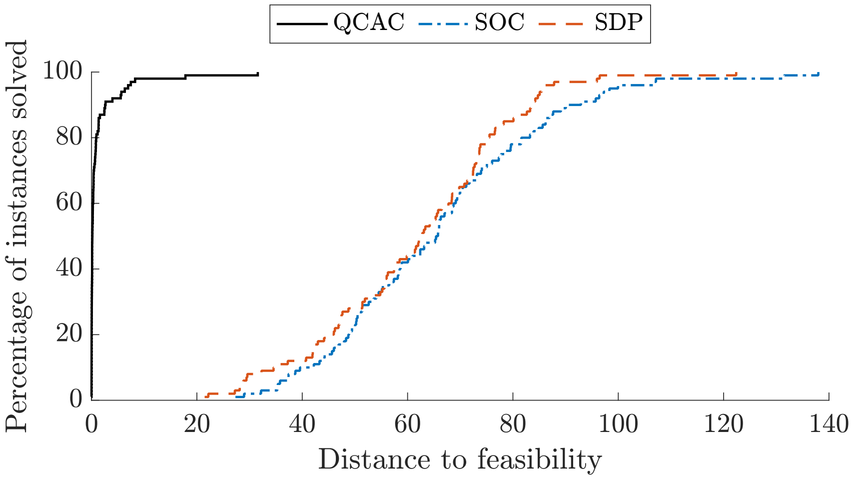
Model	Optimality gap (%)		
	Median	Min	Max
QCAC approximation	**1.4824**	**0.006**	**12.0822**
SOC relaxation	25.2628	10.5189	32.574
SDP relaxation	10.0873	3.9027	15.1559

What about solution time?



One order of magnitude faster and more accurate than the SDP relaxation!

Even more important, what about distance to feasibility?



And significantly closer to being feasible!

Generation dispatch correlation

Table: Correlation coefficient comparison

Model	Correlation coefficient	
	Active power	Reactive power
QCAC approximation	**0.9949**	**0.8943**
SOC relaxation	0.8605	0.6101
SDP relaxation	0.9607	0.5601

- Better correlation in generation dispatch makes the model more suitable for applications sensitive to active and reactive power generation.
- For instance, unit commitment and optimal reactive power dispatch.

Learning Convex Approximations for AC-OPF with Zero-Injection Feasibility Guarantees

Gonzalo E. Constante Flores, Can Li

PURDUE UNIVERSITY

PSE Seminar 2023