ECE 598: Computational Power Systems **Winter 2025**

Lecture 5: ADMM– Applications to OPF

Lecturer: Vladimir Dvorkin *Scribe(s): Yanyong Mao and Zhehua Xia*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

5.1 Recap of DC-OPF

- DC power flow model considers only active power flow. **Assumptions** made for more efficient computation:

  1. Transmission lines has low r/x ratios (about $1/5 - 1/10$ for voltage level 220-400 kV).

  $$\mathbf{G} \approx \mathbf{0} \quad and \quad b_{nm} = \frac{x_{nm}}{r_{nm}^2 + x_{nm}^2}$$

  2. Small angle difference: $sin(\theta_n - \theta_m) \approx \theta_n - \theta_m$
  3. Voltage magnitudes: $V_n \approx 1$

- **DC power flow model**

$$P_n \approx \sum_{m:n\sim m} P_{nm} = \sum_{m:n\sim m} b_{nm}(\theta_n - \theta_m)$$

- **$b\theta$-formulation:** suits consensus ADMM (voltage decoupling)

$$\begin{aligned}
\min_{\boldsymbol{p},\boldsymbol{\theta}} \quad & c(\boldsymbol{p}) && \text{generation cost} \\
\text{subject to} \quad & \boldsymbol{B\theta} = \boldsymbol{p} - \boldsymbol{d} && \text{active power balance} \\
& \underline{\boldsymbol{p}} \leq \boldsymbol{p} \leq \overline{\boldsymbol{p}} && \text{min/max gen p-limits} \\
& |f(\boldsymbol{\theta})| \leq \overline{\mathbf{f}} && \text{power flow limits}
\end{aligned}$$

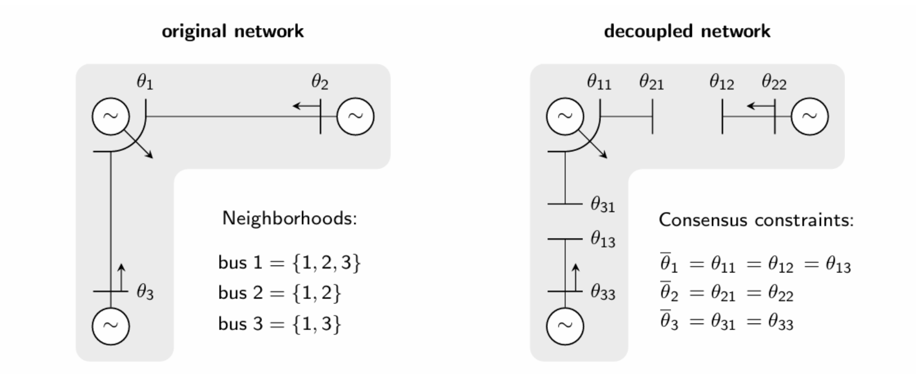
- **PTDF-based formulation:** suits exchange ADMM (LMP exchange)

$$\begin{aligned}
\min_{\boldsymbol{p}} \quad & c(\boldsymbol{p}) && \text{generation cost} \\
\text{subject to} \quad & \mathbf{1}^\top(\boldsymbol{p} - \boldsymbol{d}) = 0 && \text{active power balance} \\
& |\boldsymbol{F}(\boldsymbol{p} - \boldsymbol{d})| \leq \overline{\boldsymbol{f}} && \text{power flow limits} \\
& \underline{\boldsymbol{p}} \leq \boldsymbol{p} \leq \overline{\boldsymbol{p}} && \text{min/max gen p-limits}
\end{aligned}$$

$f : \mathbb{R}^n \to \mathbb{R}^e$ maps the vector of n voltage angles to e power flows

5.2 Consensus ADMM for DC-OPF

5.2.1 Decoupled DC-OPF and consensus



- Each node creates copies of their own and has copies of neighboring voltage angles (e.g. θ_{12} is the voltage angle of bus 1 copied at bus 2)
- Consensus constraints force local copies to agree on the same values
- Relaxing consensus constraints leads to a set of smaller OPF problems. Each node only needs to independently solve its local OPF without considering the voltage angle values of other nodes, which significantly reduces computational complexity **DC-OPF with decoupled voltage variables:**

$$
\begin{array}{lll}
\underset{\boldsymbol{p},\boldsymbol{\theta_1},\ldots,\boldsymbol{\theta_n},\bar{\boldsymbol{\theta}}}{\text{minimize}} & \sum_{i=1}^{n} c_i(p_i) & \\
\text{subject to} & \boldsymbol{B}_i^\top \boldsymbol{\theta}_i = \boldsymbol{p}_i - \boldsymbol{d}_i & \text{power balance for each bus i} \\
& \underline{\boldsymbol{p}}_i \leq \boldsymbol{p}_i \leq \bar{\boldsymbol{p}}_i & \text{gen limits for each bus i} \\
& |f_i(\boldsymbol{\theta}_i)| \leq \bar{\mathbf{f}}_i & \text{flow limits for adjacent to bus i lines} \\
& \boldsymbol{\theta}_i = \bar{\boldsymbol{\theta}} & \text{voltage consensus for each bus i}
\end{array}
$$

- Solutions to original and decoupled DC-OPF problems are the same

5.2.2 Consensus ADMM for decoupled DC-OPF

$$
\begin{array}{lll}
\underset{\boldsymbol{p},\boldsymbol{\theta_1},\ldots,\boldsymbol{\theta_n},\bar{\boldsymbol{\theta}}}{\text{minimize}} & \sum_{i=1}^{n} c_i(p_i) + \sum_{i=1}^{n} \boldsymbol{\mu}_i^\top (\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}) + \sum_{i=1}^{n} \frac{\rho}{2} \|\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}\|_2^2 & \\
\text{subject to} & \boldsymbol{B}_i^\top \boldsymbol{\theta}_i = p_i - d_i, & \forall t = 1, \ldots, n \\
& \underline{\boldsymbol{p}}_i \leq \boldsymbol{p}_i \leq \bar{\boldsymbol{p}}_i, & \forall t = 1, \ldots, n \\
& |f_i(\boldsymbol{\theta}_i)| \leq \bar{\mathbf{f}}_i, & \forall t = 1, \ldots, n
\end{array}
$$

- Dualize the consensus constraint $\rightarrow$ the problem is separable per bus
- Add the regularization term for consensus constraints with some $\rho > 0$

5.2.3 Consensus ADMM algorithm

Consensus ADMM algorithm is used to solve large-scale optimization problems. Its basic idea is to decompose the original problem into multiple sub problems, solve each node independently, and then coordinate the global solution through a consensus step.

for $k = 1, \ldots, K$ **do**
update local copies of voltage angles
for $i = 1, \ldots, n$ **do**

$$\boldsymbol{\theta}_i^k = \arg\min_{p_i, \boldsymbol{\theta_i}} c_i(p_i) + (\boldsymbol{\mu}_i^{k-1})^\top \boldsymbol{\theta}_i + \frac{\rho}{2}\|\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}^{k-1}\|_2^2$$

subject to local OPF constraints
end for
update consensus variable

$$\bar{\boldsymbol{\theta}}^k = \arg\min_{\bar{\boldsymbol{\theta}}} -\sum_{i=1}^{n}(\boldsymbol{\mu}_i^{k-1})^\top \bar{\boldsymbol{\theta}} + \sum_{i=1}^{n} \frac{\rho}{2}\|\boldsymbol{\theta}_i^{k-1} - \bar{\boldsymbol{\theta}}\|_2^2$$

update the dual variable
for $i = 1, \ldots, n$ **do**

$$\boldsymbol{\mu}_i^k = \boldsymbol{\mu}_i^{k-1} + \rho(\boldsymbol{\theta}_i^k - \bar{\boldsymbol{\theta}}^k)$$

end for
end for=0

- n = number of buses, k = iteration
- Iterate until the primal residuals $\|\boldsymbol{\theta}_i^k - \bar{\boldsymbol{\theta}}^k\| \leq \epsilon_{tot} \forall i$
- Suffices to communicate only with neighbors

5.3 Exchange ADMM for DC-OPF

5.3.1 Decentralizing DC-OPF

$$\underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad c(\boldsymbol{p}) - (\mathbf{1}\lambda - \boldsymbol{F}^\top \overline{\mu} + \boldsymbol{F}^\top \underline{\mu})^\top (\boldsymbol{p} - \boldsymbol{d}) + \frac{\rho}{2}\|\mathbf{1}^\top(\boldsymbol{p} - \boldsymbol{d})\|_2^2 \tag{5.1}$$

$$+\frac{\rho}{2}\|max\{\boldsymbol{F}(\boldsymbol{p} - \boldsymbol{d}) - \overline{\boldsymbol{f}}, 0\}\|_2^2 + \frac{\rho}{2}\|max\{\overline{\boldsymbol{f}} - \boldsymbol{F}(\boldsymbol{p} - \boldsymbol{d}), 0\}\|_2^2 \tag{5.2}$$

- Dualize the coupling constraints, regroup the terms to form LMPs
- Add regularization terms for coupling constraints with some $\rho > 0$
- Since the power flow must be non-negative, the max operator is used to constrain it to zero or a larger value, ensuring physical feasibility.

5.3.2 Exchange ADMM

- At every iteration k, the central agent computes LMPs

$$\pi^k = \mathbf{1}\lambda^k - \boldsymbol{F}^\top \overline{\mu}^k + \boldsymbol{F}^\top \underline{\mu}^k \quad \in \mathbb{R}^n$$

- and private constraint mismatch for each agent $i = 1, \dots, n$

$$\Delta p_i^k = \mathbf{1}^\top (\boldsymbol{p}^k - \boldsymbol{d}) - p_i^k \quad \in \mathbb{R}$$
$$\Delta f_{+i}^k = \boldsymbol{F}(\boldsymbol{p}^k - \boldsymbol{d}) - \overline{\boldsymbol{f}} - \boldsymbol{F}_i^\top p_i^k \quad \in \mathbb{R}^e$$
$$\Delta f_{-i}^k = \overline{\boldsymbol{f}} - \boldsymbol{F}(\boldsymbol{p}^k - \boldsymbol{d}) - \boldsymbol{F}_i^\top p_i^k \quad \in \mathbb{R}^e$$

Algorithm

---

for $k = 1, \dots, K$ **do**
update dispatch decision in response to LMP and constraint mismatch
for $i = 1, \dots, n$ **do**

$$p_i^k = \arg \min_{\underline{p}_i \le p_i \le \overline{p}_i} c_i(p_i) - \pi_i^{k-1} p_i + \frac{\rho}{2} \|\Delta p_i^{k-1} - p_i\|_2^2 \tag{5.3}$$
$$+ \frac{\rho}{2} \|max\{\Delta f_{+i}^{k-1} - \boldsymbol{F}_i^\top p_i, 0\}\|_2^2 + \frac{\rho}{2} \|max\{\Delta f_{-i}^{k-1} - \boldsymbol{F}_i^\top p_i, 0\}\|_2^2 \tag{5.4}$$

end for
update dual variable
for $i = 1, \dots, e$ **do**

$$\overline{\boldsymbol{\mu}}_i^k = max\{\overline{\boldsymbol{\mu}}_i^{k-1} + \rho(\boldsymbol{F}(\boldsymbol{p}^k - \boldsymbol{d}) - \overline{\boldsymbol{f}}), 0\} \tag{5.5}$$
$$\underline{\boldsymbol{\mu}}_i^k = max\{\underline{\boldsymbol{\mu}}_i^{k-1} + \rho(\overline{\boldsymbol{f}} - \boldsymbol{F}(\boldsymbol{p}^k - \boldsymbol{d})), 0\} \tag{5.6}$$

$$\lambda^k = \lambda^{k-1} + \rho(\mathbf{1}^\top (\boldsymbol{p}^k - \boldsymbol{d})) \tag{5.7}$$
$$\pi^k = \pi^{k-1} + \mathbf{1}\lambda^k - \boldsymbol{F}^\top \overline{\mu}^k + \boldsymbol{F}^\top \underline{\mu}^k \tag{5.8}$$
$$\tag{5.9}$$

for $= 1, \dots, n$ **do**

$$\Delta p_i^k = \mathbf{1}^\top (\boldsymbol{p}^k - \boldsymbol{d}) - p_i^k \tag{5.10}$$
$$\Delta f_{+i}^k = \boldsymbol{F}(\boldsymbol{p}^k - \boldsymbol{d}) - \overline{\boldsymbol{f}} - \boldsymbol{F}_i^\top p_i^k \tag{5.11}$$
$$\Delta f_{-i}^k = \overline{\boldsymbol{f}} - \boldsymbol{F}(\boldsymbol{p}^k - \boldsymbol{d}) - \boldsymbol{F}_i^\top p_i^k \tag{5.12}$$
$$\tag{5.13}$$

update auxiliary variables
end for=0

- Iterate until all primal residuals reach tolerance ϵ_{tot}
- Requires centralized update of LMPs and constraint mismatches
- The sub-problem admit a closed-form solution p_i

5.4 ADMM for distribution AC-OPF

5.4.1 Overview

In this section, we reviewed the modeling of AC distribution networks using a relaxed branch flow formulation and its linearized version. Then, we discussed the formulation of the AC-OPF problem, its dualization, and finally the ADMM algorithm with closed-form updates for both voltage and reactive power variables.

5.4.2 Distribution System Models

Relaxed Branch Flow Model

For radial (tree-like) distribution networks, the full branch flow model is written as

$$\begin{aligned} \sum_k P_{mk} &= P_{nm} - r_{nm} i_{nm} + p_m \\ \sum_k Q_{mk} &= Q_{nm} - x_{nm} i_{nm} + q_m \\ v_m &= v_n - 2r_{nm}P_{nm} - 2x_{nm}Q_{nm} + (r_{nm}^2 + x_{nm}^2) i_{nm} \\ i_{nm} &= \frac{P_{nm}^2 + Q_{nm}^2}{v_n} \end{aligned}$$

where

- n is the upstream node, m is the midstream node, and k are downstream nodes.
- To avoid quadratic terms, $v = |V|^2$ and $i = |I|^2$.
- The formulation drops voltage and current phase information.

Linearized Distribution Flow (LinDistFlow)

To furthur overcome the nonlinearity of the full branch flow equations, the LinDistFlow model is introduced by neglecting loss-related terms $r_{nm}i_{nm}, x_{nm}i_{nm},$ and $(r_{nm}^2 + x_{nm}^2)i_{nm}$, so that voltage drop and line power flows are approximately *linearly* related to power injections. Although this overestimates voltage magnitudes and underestimates current magnitudes, it successfully linearizes the DistFlow equations with limited error.

In its compact form, the voltage at each non-root node is approximated by

$$\mathbf{v} = v_0 \mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q}$$

where:

- v_0 is the known substation (root node) voltage (typically 1 p.u.).
- $\mathbf{R}$ and $\mathbf{X}$ are symmetric positive definite matrices derived from the network's branch-bus incidence notation. (For derivation, refer to page 11 of this slides.)
- $\mathbf{p}$ and $\mathbf{q}$ are the vectors of active and reactive power injections.

5.4.3 Distribution AC-OPF Based on LinDistFlow

The distribution AC-OPF problem is proposed to control distributed energy resources (DERs) such that voltage magnitudes remain within prescribed limits. Assuming that active power injections are known, and the reactive power is controllable, we can then optimize reactive power injection to achieve our objectives, such as minimizing voltage deviations. The optimization problem is formulated as:

$$\begin{aligned} \underset{\mathbf{v},\hat{\mathbf{q}}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{q}^\top \mathbf{C}\mathbf{q} \\ \text{subject to} \quad & \mathbf{v} = v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q} + \hat{\mathbf{q}}) \\ & \underline{\mathbf{v}} \leqslant \mathbf{v} \leqslant \overline{\mathbf{v}} \\ & \underline{\mathbf{q}} \leqslant \mathbf{q} \leqslant \overline{\mathbf{q}} \end{aligned}$$

where, the term $\|v - \mathbf{1}\|_2^2$ represents the squared voltage deviation from the nominal value (1 p.u.), and $\mathbf{q}$ denote the incremental reactive power that we control to optimize the system.

This is a simplified but often used AC power flow model. The full AC solution can be recovered from the solution it gives.

5.4.4 Dualization and ADMM Formulation

The compact LinDisFlow equation contains two optimization variables, so we can dualize this coupling equality constraint. Then we can add a quadratic penalty term to enhance the Lagrangian (refer to AL method in Lecture-4), which results in

$$\mathcal{L}(v, q, \lambda) = \frac{1}{2}\mathbf{q}^\top \mathbf{C}\mathbf{q} + \boldsymbol{\lambda}^\top (v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q} + \hat{\mathbf{q}}) - \mathbf{v}) + \frac{\rho}{2}\|v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q} + \hat{\mathbf{q}}) - \mathbf{v}\|_2^2$$

with dual variable vector $\boldsymbol{\lambda}$ and penalty parameter $\rho > 0$. Notice that the problem becomes separable with respect to the voltage v and reactive power q variables.

Voltage Sub-problem

The voltage update step is derived from minimizing the augmented Lagrangian with respect to $\mathbf{v}^k$ while keeping $\mathbf{q}$ fixed at the previous iteration value. The voltage sub-problem is:

$$\begin{aligned} \mathbf{v}^k = \arg\min_{\mathbf{v}} \quad & -\lambda^{k-1\top}\mathbf{v} + \frac{\rho}{2}\|v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q} + \hat{\mathbf{q}}) - \mathbf{v}\|_2^2 \\ \text{subject to} \quad & \underline{\mathbf{v}} \leqslant \mathbf{v}^k \leqslant \overline{\mathbf{v}} \end{aligned}$$

Due to the quadratic structure and fixation of other variables, the solution can be obtained in closed form:

Step 1: Compute the unconstrained update by setting derivative to zero:

$$\tilde{\mathbf{v}}^k = \frac{1}{\rho}\boldsymbol{\lambda}^{k-1} + v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q}^{k-1} + \hat{\mathbf{q}})$$

Step 2: Project onto feasible region:

$$\mathbf{v}^k = \max\{\underline{\mathbf{v}}, \min\{\tilde{\mathbf{v}}^k, \overline{\mathbf{v}}\}\}$$

Reactive Power Sub-problem

Similarly, the reactive power update is obtained by minimizing the augmented Lagrangian with respect to $\mathbf{q}$, with $\mathbf{v}$ fixed at the previous iteration:

$$\begin{aligned} \mathbf{q}^k = \arg\min_{\mathbf{q}} \quad & \mathbf{q}^\top \mathbf{C}\mathbf{q} + {\boldsymbol{\lambda}^{k-1}}^\top \mathbf{X}\mathbf{q} + \frac{\rho}{2}\|v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q}+\hat{\mathbf{q}}) - \mathbf{v}^{k-1}\|_2^2 \\ \text{subject to} \quad & \underline{\mathbf{q}} \leqslant \mathbf{q}^k \leqslant \overline{\mathbf{q}} \end{aligned}$$

Again, this sub-problem has the closed-form solution as well:

Step 1: Compute the unconstrained update:

$$\tilde{\mathbf{q}}^k = -(\mathbf{C} + \rho\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top \left(\boldsymbol{\lambda}^{k-1} + \rho(v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}\hat{\mathbf{q}} - \mathbf{v}^{k-1})\right)$$

Step 2: Project onto feasible region:

$$\mathbf{q}^k = \max\{\underline{\mathbf{q}}, \min\{\tilde{\mathbf{q}}^k, \overline{\mathbf{q}}\}\}$$

5.4.5 Overall ADMM Algorithm for AC-OPF

The complete ADMM iterating procedure for the distribution AC-OPF is summarized as follows:

for $k = 1, \ldots, K$ **do**

Primal update: voltage

$\tilde{\mathbf{v}}^k = \frac{1}{\rho}\boldsymbol{\lambda}^{k-1} + v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q}^{k-1} + \hat{\mathbf{q}})$

$\mathbf{v}^k = \max\{\underline{\mathbf{v}}, \min\{\tilde{\mathbf{v}}^k, \overline{\mathbf{v}}\}\}$

Primal update: reactive power

$\tilde{\mathbf{q}}^k = -(\mathbf{C} + \rho\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top \left(\boldsymbol{\lambda}^{k-1} + \rho(v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}\hat{\mathbf{q}} - \mathbf{v}^{k-1})\right)$

$\mathbf{q}^k = \max\{\underline{\mathbf{q}}, \min\{\tilde{\mathbf{q}}^k, \overline{\mathbf{q}}\}\}$

Dual update

$\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^{k-1} + \rho(v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q}^k + \hat{\mathbf{q}}) - \mathbf{v}^k)$

Check convergence

$\|v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q}^k + \hat{\mathbf{q}}) - \mathbf{v}^k\|_2 \leqslant \varepsilon_{tol}$

end for=0

Due to the characteristic of ADMM, the availability of a closed-form solution here avoids any need for iterative optimization, which greatly simplifies the updates.

5.4.6 Brief Conclusion

1. The use of the LinDistFlow model simplifies the complex nonlinear AC power flow equations, which makes the optimization problem more tractable.
2. The quadratic nature of the sub-problems (with simple constraints) results in closed-form solutions, which eliminates the need for optimization solvers within each ADMM iteration.

3. These closed-form updates not only reduce computation requirement but also enable a distributed implementation by further using consenses ADMM, where each DER can update its local variables with minimal communication.