# ECE 598 Computational Power Systems

## ADMM – Intro

Vladimir Dvorkin

University of Michigan

# Last lecture recap

- Look around you and form teams of 2 people (1 min)
- Quickly review your notes or the slide deck (1 min)
- Share your **three** personal highlights with your partner (3 min)
- Get iClicker app ready

# Quiz 2: Which statement about Lagrange duality is TRUE?

**A** Karush–Kuhn–Tucker optimality conditions are necessary and sufficient for convex problems and sufficient for non-convex problems.

**B** Because of complementarity slackness and primal feasibility, the Lagrangian of the economic dispatch equals to the dispatch cost at optimality.

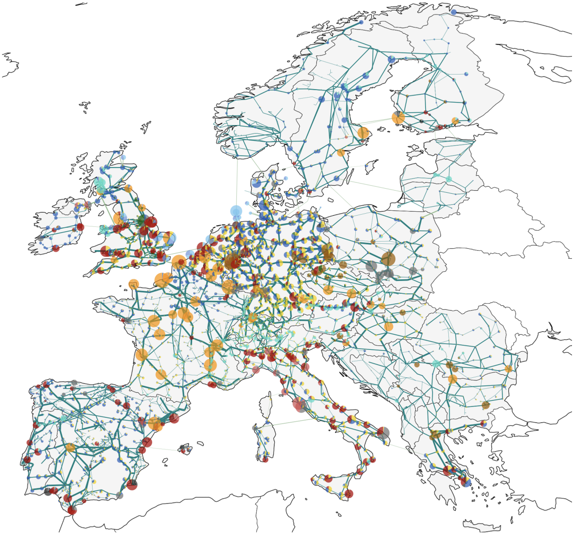
**C** For convex economic dispatch problem, weak duality holds, while strong duality is difficult to establish.

**D** Electricity price is the cost of the marginal generator; we compute the price as the dual variable of the maximum dispatch constraint of the marginal generator.

# Coordination of distributed energy resources (DERs)

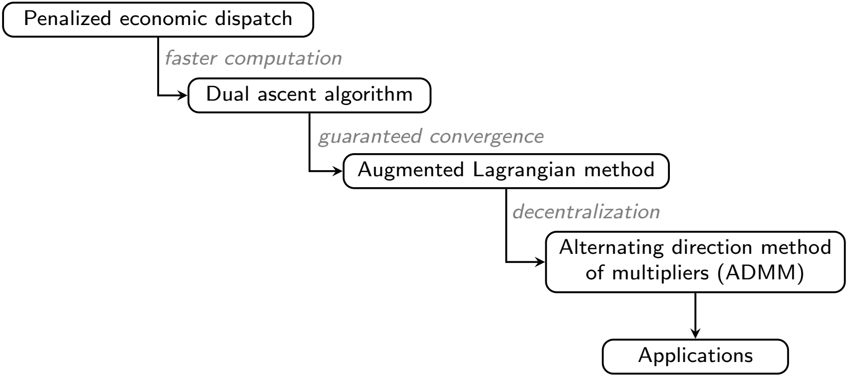
How to dispatch DERs (EV, PV, batteries) without knowing too much about them?

# Cross-border coordination



How independent grid operators solve the *continental* dispatch problem?

# Outline


Penalized economic dispatch
faster computation
Dual ascent algorithm
guaranteed convergence
Augmented Lagrangian method
decentralization
Alternating direction method of multipliers (ADMM)
Applications

# Penalized economic dispatch

- Economic dispatch (ED) problem

$$\begin{aligned} \underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad & c(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{1}^\top \mathbf{p} = d \quad : \nu \end{aligned}$$

where $c : \mathbb{R}^n \mapsto \mathbb{R}$ is convex cost function, $\mathbf{p} \in \mathbb{R}^n$ is generator dispatch

[1] $\nabla_{\mathbf{p}} \left( \frac{\nu}{2} \|\mathbf{1}^\top \mathbf{p} - d\|_2^2 \right)$ grows linearly in $\mathbf{p}$, the Lipschitz constant $\nu \|\mathbf{1}\| = \nu\sqrt{n}$

# Penalized economic dispatch

- Economic dispatch (ED) problem

$$\begin{aligned} \underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad & c(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{1}^\top \mathbf{p} = d \quad : \nu \end{aligned}$$

  where $c : \mathbb{R}^n \mapsto \mathbb{R}$ is convex cost function, $\mathbf{p} \in \mathbb{R}^n$ is generator dispatch

- **Proposition**: Let's solve an unconstrained problem for some large value $\nu$

$$\underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad c(\mathbf{p}) + \frac{\nu}{2} \left\| \mathbf{1}^\top \mathbf{p} - d \right\|_2^2$$

[1] $\nabla_{\mathbf{p}} \left( \frac{\nu}{2} \|\mathbf{1}^\top \mathbf{p} - d\|_2^2 \right)$ grows linearly in $\mathbf{p}$, the Lipschitz constant $\nu \|\mathbf{1}\| = \nu\sqrt{n}$

# Penalized economic dispatch

- Economic dispatch (ED) problem

$$\begin{aligned} \underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad & c(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{1}^\top \mathbf{p} = d \quad : \nu \end{aligned}$$

where $c : \mathbb{R}^n \mapsto \mathbb{R}$ is convex cost function, $\mathbf{p} \in \mathbb{R}^n$ is generator dispatch

- **Proposition**: Let's solve an unconstrained problem for some large value $\nu$

$$\underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad c(\mathbf{p}) + \frac{\nu}{2} \left\| \mathbf{1}^\top \mathbf{p} - d \right\|_2^2$$

**Pros:** As $\nu \to \infty$, the solution approaches that of ED

**Cons:** The rate of convergence is proportional to Lipschitz constant $\nu\sqrt{n}$[1].

[1] $\nabla_{\mathbf{p}} \left( \frac{\nu}{2} \|\mathbf{1}^\top \mathbf{p} - d\|_2^2 \right)$ grows linearly in $\mathbf{p}$, the Lipschitz constant $\nu \|\mathbf{1}\| = \nu\sqrt{n}$

# Penalized economic dispatch

- Economic dispatch (ED) problem

$$\begin{aligned} \underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad & c(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{1}^\top \mathbf{p} = d \quad : \nu \end{aligned}$$

  where $c : \mathbb{R}^n \mapsto \mathbb{R}$ is convex cost function, $\mathbf{p} \in \mathbb{R}^n$ is generator dispatch

- **Proposition**: Let's solve an unconstrained problem for some large value $\nu$

$$\underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad c(\mathbf{p}) + \frac{\nu}{2} \left\| \mathbf{1}^\top \mathbf{p} - d \right\|_2^2$$

  **Pros:** As $\nu \rightarrow \infty$, the solution approaches that of ED

  **Cons:** The rate of convergence is proportional to Lipschitz constant $\nu\sqrt{n}$[1].

- We will work with Lagrange duality to find a better approach to solving ED

[1] $\nabla_{\mathbf{p}} \left( \frac{\nu}{2} \|\mathbf{1}^\top \mathbf{p} - d\|_2^2 \right)$ grows linearly in $\mathbf{p}$, the Lipschitz constant $\nu \|\mathbf{1}\| = \nu\sqrt{n}$

# Lagrange duality

- ED optimization

$$\begin{aligned} \underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad & c(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{1}^\top \mathbf{p} = d \quad : \nu \end{aligned}$$

- Partial Lagrangian function of ED

$$\max_{\nu} \min_{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}} \mathcal{L}(\mathbf{p}, \nu) = c(\mathbf{p}) + \nu(d - \mathbf{1}^\top \mathbf{p})$$

- Finding the optimal solution of ED is the same as finding the *saddle point* of $\mathcal{L}$

$$\mathcal{L}(\mathbf{p}^\star, \nu^\star) = c(\mathbf{p}^\star) + \nu^\star(d - \mathbf{1}^\top \mathbf{p}^\star) = c(\mathbf{p}^\star)$$

# Dual ascent algorithm

$$\mathcal{L}(\mathbf{p}^\star, \nu^\star) = c(\mathbf{p}^\star) + \nu^\star(d - \mathbf{1}^\top \mathbf{p}^\star) = c(\mathbf{p}^\star)$$

- One way to optimize Lagrangian function is via *dual ascent*
- Consider a sequence of iterations $k = 1, \dots, K$ starting from $(\mathbf{p}^0, \nu^0)$
- At every iteration $k$, the primal-dual update for some $\rho > 0$:

$$\begin{aligned} \mathbf{p}^{k+1} &= \arg\min_{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}} \mathcal{L}(\mathbf{p}, \nu^k) && \textit{primal update} \\ \nu^{k+1} &= \nu^k + \rho(d - \mathbf{1}^\top \mathbf{p}^{k+1}) && \textit{dual ascent} \end{aligned}$$

- What is the economic intuition behind dual ascent?

# Dual ascent algorithm

$$\mathcal{L}(\mathbf{p}^\star, \nu^\star) = c(\mathbf{p}^\star) + \nu^\star(d - \mathbf{1}^\top \mathbf{p}^\star) = c(\mathbf{p}^\star)$$

- One way to optimize Lagrangian function is via *dual ascent*
- Consider a sequence of iterations $k = 1, \ldots, K$ starting from $(\mathbf{p}^0, \nu^0)$
- At every iteration $k$, the primal-dual update for some $\rho > 0$:

$$\begin{aligned} \mathbf{p}^{k+1} &= \arg\min_{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}} \mathcal{L}(\mathbf{p}, \nu^k) && \textit{primal update} \\ \nu^{k+1} &= \nu^k + \rho(d - \mathbf{1}^\top \mathbf{p}^{k+1}) && \textit{dual ascent} \end{aligned}$$

- What is the economic intuition behind dual ascent?
- For certain problem classes, dual ascent yields efficient, convergent algorithms to an optimal primal-dual solution $(\mathbf{p}^\star, \nu^\star)$
- However, it may fail for some problems in power systems
- **Example:** in ED with linear cost $c(\mathbf{p}) = \mathbf{c}^\top \mathbf{p}$, the dual function

$$\phi(\boldsymbol{\nu}) = \mathcal{L}(\mathbf{p}^\star, \nu) = \begin{cases} d\nu, & \text{if } \mathbf{c} - \mathbf{1}\nu = \mathbf{0} \\ -\infty(\text{unbounded}), & \text{if otherwise} \end{cases} \qquad (\textit{Lec.2,p.14})$$

is unbounded without additional constraints $\Rightarrow$ no meaningful primal update

# Augmented Lagrangian

- Intuitively, the dual ascent fails to converge because the Lagrangian does not penalize the power balance constraint strongly enough

- Remedy: **Augmented Lagrangian**

$$\mathcal{L}_\rho(\mathbf{p}, \nu) = c(\mathbf{p}) + \nu(d - \mathbf{1}^\top \mathbf{p}) + \frac{\rho}{2} \left\| d - \mathbf{1}^\top \mathbf{p} \right\|_2^2$$

  where $\rho > 0$ is a penalty parameter

- The augmented Lagrangian can be regarded as the Lagrangian function for

$$\begin{aligned} \underset{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}}{\text{minimize}} \quad & c(\mathbf{p}) + \frac{\rho}{2} \left\| d - \mathbf{1}^\top \mathbf{p} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{1}^\top \mathbf{p} = d \quad : \nu \end{aligned}$$

- Even though the dispatch cost can be linear, the overall objective is quadratic

- Hence, the dual function is always bounded (*Lec.2,p.14*) $\Rightarrow$ primal update exists

# Augmented Lagrangian method

- Augmented Lagrangian

$$\mathcal{L}_\rho(\mathbf{p},\nu) = c(\mathbf{p}) + \nu(d - \mathbf{1}^\top \mathbf{p}) + \frac{\rho}{2}\left\| d - \mathbf{1}^\top \mathbf{p}\right\|_2^2$$

- At every iteration $k$, the primal-dual update:

$$\begin{aligned}
\mathbf{p}^{k+1} &= \arg\min_{\underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}}} \mathcal{L}_\rho(\mathbf{p}, \nu^k) && \textit{primal update} \\
\nu^{k+1} &= \nu^k + \rho(d - \mathbf{1}^\top \mathbf{p}^{k+1}) && \textit{dual ascent}
\end{aligned}$$

# Augmented Lagrangian method

- Augmented Lagrangian

$$\mathcal{L}_\rho(\mathbf{p},\nu) = c(\mathbf{p}) + \nu(d - \mathbf{1}^\top\mathbf{p}) + \frac{\rho}{2}\left\|d - \mathbf{1}^\top\mathbf{p}\right\|_2^2$$

- At every iteration $k$, the primal-dual update:

$$\begin{aligned} \mathbf{p}^{k+1} &= \arg\min_{\underline{\mathbf{p}}\leqslant\mathbf{p}\leqslant\overline{\mathbf{p}}} \mathcal{L}_\rho(\mathbf{p},\nu^k) && \textit{primal update} \\ \nu^{k+1} &= \nu^k + \rho(d - \mathbf{1}^\top\mathbf{p}^{k+1}) && \textit{dual ascent} \end{aligned}$$

- Working with augmented Lagrangian, we end up solving the original problem

$$\begin{aligned} \mathbf{0} =& \nabla_{\mathbf{p}}\mathcal{L}_\rho(\mathbf{p}^{k+1},\nu^k) \quad \textit{primal update for aug. problem} \\ =& \nabla_{\mathbf{p}}c(\mathbf{p}^{k+1}) + \mathbf{1}\nu^k + \rho\mathbf{1}(d - \mathbf{1}^\top\mathbf{p}^{k+1}) \\ =& \nabla_{\mathbf{p}}c(\mathbf{p}^{k+1}) + \mathbf{1}\nu^{k+1} = \nabla_{\mathbf{p}}\mathcal{L}(\mathbf{p}^{k+1},\nu^{k+1}) \quad \textit{Lag. of the original problem} \end{aligned}$$

- Thus, $\mathbf{p}^{k+1}$ minimizes the Lagrangian function of the original problem

# Alternating direction method of multipliers

Two drawbacks of the augmented Lagrangian method:

- **Computation:** solves optimization for primal updates (no closed-form sol.)
- **Centralization**: every generator shares opt. data (as in centralized ED)

## Alternating direction method of multipliers

Two drawbacks of the augmented Lagrangian method:

- **Computation:** solves optimization for primal updates (no closed-form sol.)
- **Centralization**: every generator shares opt. data (as in centralized ED)

Alternating direction method of multipliers (**ADMM**) solves these two issues:

- Fix $\nu$ and $p_2, \dots, p_n$, solve for $p_1$ :

$$p_1^{k+1} = \arg \min_{\underline{p}_1 \leqslant p_1 \leqslant \overline{p}_1} \mathcal{L}_\rho(p_1, p_2^k, \dots, p_n^k, \nu^k)$$

- Fix $\nu$ and all $p_1, \dots, p_n$ but $p_i$, solve for $p_i$ :

$$p_i^{k+1} = \arg \min_{\underline{p}_i \leqslant p_i \leqslant \overline{p}_i} \mathcal{L}_\rho(p_1^{k+1}, \dots, p_i, \dots, p_n^k, \nu^k)$$

- Fix $\nu$ and $p_1, \dots, p_{n-1}$, solve for $p_n$ :

$$p_n^{k+1} = \arg \min_{\underline{p}_n \leqslant p_n \leqslant \overline{p}_n} \mathcal{L}_\rho(p_1^{k+1}, \dots, p_{n-1}^{k+1}, p_n, \nu^k)$$

- Fix all $p_1, \dots, p_n$ and update the dual using dual ascent:

$$\nu^{k+1} = \nu^k + \rho(d - \sum_{i=1}^n p_i^{k+1})$$

# Closed-form solution for ADMM sub-problems

$$
\begin{aligned}
p_i^{k+1} &= \arg \min_{\underline{p}_i \leqslant p_i \leqslant \overline{p}_i} \mathcal{L}_\rho(p_1^{k+1}, \ldots, p_i, \ldots, p_n^k, \nu^k) \\
&= \arg \min_{\underline{p}_i \leqslant p_i \leqslant \overline{p}_i} c_i(p_i) - \nu^k p_i + \frac{\rho}{2} \| \underbrace{d - \textstyle\sum_{j=1,\ldots,i-1} p_j^{k+1} - \sum_{j=i+1,\ldots,n} p_j^k}_{\Delta p_i^k \quad \text{(power mismatch)}} - p_i \|_2^2
\end{aligned}
$$

$$
(\nabla_{p_i} c_i(p_i) - \nu^k - \rho(\Delta p_i^k - p_i) = 0)
$$

$$
= \begin{cases} \frac{1}{2c_{2i}+\rho}(\nu^k + \rho \Delta p_i^k - c_{1i}), & \text{if quad. cost } c_i(p_i) = c_{1i} p_i + c_{2i} p_i^2 \\ \Delta p_i^k - \frac{1}{\rho}(c_{1i} + \nu^k), & \text{if linear cost } c_i(p_i) = c_{1i} p_i \end{cases}
$$

followed by projection on the feasible dispatch range $[\underline{p}_i, \overline{p}_i]$

- Response to price $\nu^k$ and power mismatch $\Delta p_i^k$ with dispatch decision $p_i^{k+1}$
- Local computation: no optimization parameters are shared (only decisions)
- Solving the sub-problem in closed-form is much easier than solving optimization

# Exchange ADMM for parallel computations

- The previous algorithm updates dispatch sequentially $\Rightarrow$ no parallelization
- Exchange ADMM: following economic intuition, enables parallelization
- Let $\Delta p^k = d - \sum_{i=1}^n p_i^k$ be the total power mismatch at iteration $k$

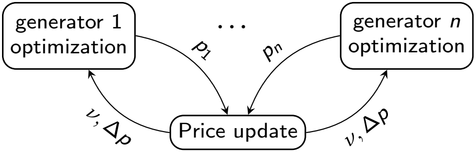
  Parallel primal update $\forall i = 1, \dots, n$ :

  $$p_i^{k+1} = \underset{\underline{p}_i \leqslant p_i \leqslant \overline{p}_i}{\operatorname{argmin}} \; c_i(p_i) - \nu^k p_i + \frac{\rho}{2} \left\| p_i - (p_i^k - \Delta p^k) \right\|_2^2$$

  followed by price update:

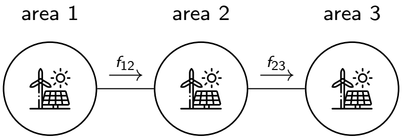
  $$\nu^{k+1} = \nu^k + \rho \Delta p^k$$

- Decentralized, parallel implementation



# Consensus ADMM – motivation

- Exchange ADMM requires a central entity for dual variable update
- Consensus ADMM requires dual update only among “neighbors”
- ... the dual update is distributed among agents (no central entity!)
- Coordinating power flows in tie-lines between different areas:

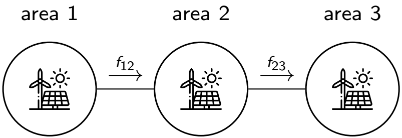



$$
\begin{aligned}
\underset{\mathbf{p}\in\mathcal{P},\mathbf{f}\in\mathcal{F}}{\text{minimize}} \quad & c_1(p_1) + c_2(p_2) + c_3(p_3) \\
\text{subject to} \quad & p_1 - d_1 = f_{12} && \text{area 1} \\
& p_2 - d_2 = f_{23} - f_{12} && \text{area 2} \\
& p_3 - d_3 = -f_{23} && \text{area 3}
\end{aligned}
$$

- Duplicate power flow variables and solve as a consensus problem

# Consensus ADMM – motivation

- Exchange ADMM requires a central entity for dual variable update
- Consensus ADMM requires dual update only among "neighbors"
- ... the dual update is distributed among agents (no central entity!)
- Coordinating power flows in tie-lines between different areas:




$$
\begin{aligned}
\underset{\mathbf{p}\in\mathcal{P},\mathbf{f}\in\mathcal{F}}{\text{minimize}} \quad & c_1(p_1) + c_2(p_2) + c_3(p_3) & \\
\text{subject to} \quad & p_1 - d_1 = f_{12}^1 & \text{area 1} \\
& p_2 - d_2 = f_{23}^2 - f_{12}^2 & \text{area 2} \\
& p_3 - d_3 = -f_{23}^3 & \text{area 3} \\
& f_{12}^1 = f_{12}^2 = f_{12} \quad : \nu_{12} & \text{consensus for 1 and 2} \\
& f_{23}^2 = f_{23}^3 = f_{23} \quad : \nu_{23} & \text{consensus for 2 and 3}
\end{aligned}
$$

- Duplicate power flow variables and solve as a consensus problem

# Consensus ADMM – formulation

Global optimization problem:

$$\begin{aligned} \underset{\mathbf{p},\mathbf{f}}{\text{minimize}} \quad & c(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{p} - \mathbf{d} = \mathbf{A}\mathbf{f} \\ & \underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}} \\ & \underline{\mathbf{f}} \leqslant \mathbf{f} \leqslant \overline{\mathbf{f}} \end{aligned}$$

Adjacency matrix $\mathbf{A}$:

$$a_{ij} = \begin{cases} +1, & \text{if node } i \text{ exports flow } j \\ -1, & \text{if node } i \text{ imports flow } j \\ 0, & \text{if otherwise} \end{cases}$$

3-area example: $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{bmatrix}$

# Consensus ADMM – formulation

Global optimization problem:

$$\begin{aligned} \underset{\mathbf{p},\mathbf{f}}{\text{minimize}} \quad & c(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{p} - \mathbf{d} = \mathbf{A}\mathbf{f} \\ & \underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}} \\ & \underline{\mathbf{f}} \leqslant \mathbf{f} \leqslant \overline{\mathbf{f}} \end{aligned}$$

Consensus optimization for $n$ areas:

$$\begin{aligned} \underset{\mathbf{p},\mathbf{f},\mathbf{f}_1,\ldots,\mathbf{f}_n}{\text{minimize}} \quad & \textstyle\sum_{i=1}^{n} c_i(p_i) \\ \text{subject to} \quad & p_i - d_i = a_i^\top \mathbf{f}_i, \ \forall i \\ & \mathbf{f}_i - \mathbf{f} = \mathbf{0} \, : \, \boldsymbol{\nu}_i, \ \forall i \\ & \underline{\mathbf{f}} \leqslant \mathbf{f} \leqslant \overline{\mathbf{f}} \end{aligned}$$

# Consensus ADMM – formulation

Global optimization problem:

$$\begin{aligned} \underset{\mathbf{p},\mathbf{f}}{\text{minimize}} \quad & c(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{p} - \mathbf{d} = \mathbf{A}\mathbf{f} \\ & \underline{\mathbf{p}} \leqslant \mathbf{p} \leqslant \overline{\mathbf{p}} \\ & \underline{\mathbf{f}} \leqslant \mathbf{f} \leqslant \overline{\mathbf{f}} \end{aligned}$$

Consensus optimization for $n$ areas:

$$\begin{aligned} \underset{\mathbf{p},\mathbf{f},\mathbf{f}_1,\ldots,\mathbf{f}_n}{\text{minimize}} \quad & \textstyle\sum_{i=1}^n c_i(p_i) \\ \text{subject to} \quad & p_i - d_i = a_i^\top \mathbf{f}_i, \ \forall i \\ & \mathbf{f}_i - \mathbf{f} = \mathbf{0} : \boldsymbol{\nu}_i, \ \forall i \\ & \underline{\mathbf{f}} \leqslant \mathbf{f} \leqslant \overline{\mathbf{f}} \end{aligned}$$

Consensus ADMM iterations:

- Private variable update for all $i = 1, \ldots, n$

$$\begin{aligned} \mathbf{f}_i^{k+1} = \underset{p_i,\mathbf{f}_i}{\text{argmin}} \quad & c_i(p_i) + \boldsymbol{\nu}_i^{k\top}\mathbf{f}_i + \frac{\rho}{2}\left\|\mathbf{f}_i - \mathbf{f}^k\right\| \\ \text{subject to} \quad & p_i - d_i = a_i^\top \mathbf{f}_i \\ & \underline{p}_i \leqslant p_i \leqslant \overline{p}_i \end{aligned}$$

- Consensus variable update

$$\begin{aligned} \mathbf{f}^{k+1} = \underset{\mathbf{f}}{\text{argmin}} \quad & -\boldsymbol{\nu}_i^{k\top}\mathbf{f} + \textstyle\sum_{i=1}^n \frac{\rho}{2}\left\|\mathbf{f}_i^{k+1} - \mathbf{f}\right\| \\ \text{subject to} \quad & \underline{\mathbf{f}} \leqslant \mathbf{f} \leqslant \overline{\mathbf{f}} \end{aligned}$$

- Dual variable update

$$\boldsymbol{\nu}_i^{k+1} = \boldsymbol{\nu}_i^k + \rho(\mathbf{f}_i^{k+1} - \mathbf{f}^{k+1})$$

# Resources


- Boyd, S., *et al*. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine learning, 3(1), 1-122. [§2,§3,§7]
- Wright, J., & Ma, Y. (2022). High-dimensional data analysis with low-dimensional models: Principles, computation, and applications. Cambridge University Press. [§8.4,§8.5]
- Molzahn, Daniel K., et al. "A survey of distributed optimization and control algorithms for electric power systems." IEEE Transactions on Smart Grid 8.6 (2017): 2941-2962.

- Look around you and form teams of 2 people (1 min)
- Formulate and code the problem below working in pairs

**In-class Exercise 1:** Solve the problem below using

1. Ipopt solver
2. penalized formulation
3. dual ascent algorithm
4. augmented Lagrangian method
5. Alternating direction method of multipliers (ADMM)

$$\begin{aligned} \underset{x_1, x_2}{\text{minimize}} \quad & \tfrac{1}{2}(x_1 - 5)^2 + \tfrac{1}{2}(x_2 + 3)^2 \\ \text{subject to} \quad & x_1 + x_2 = 10 \quad : \nu \end{aligned}$$