WASHINGTON STATE UNIVERSITY

Preliminary Exam Presentation

Scalable Multi-Period Optimal Power Flow for Active Distribution Systems

Aryan Ritwajeet Jha

Pursuing PhD (ECE) Power Systems

Introduction and Motivation

- What leads to MPOPF problem?
- What’s MPOPF problem?
- How to tackle it?
- Intended Contributions of this PhD

Introduction and Motivation

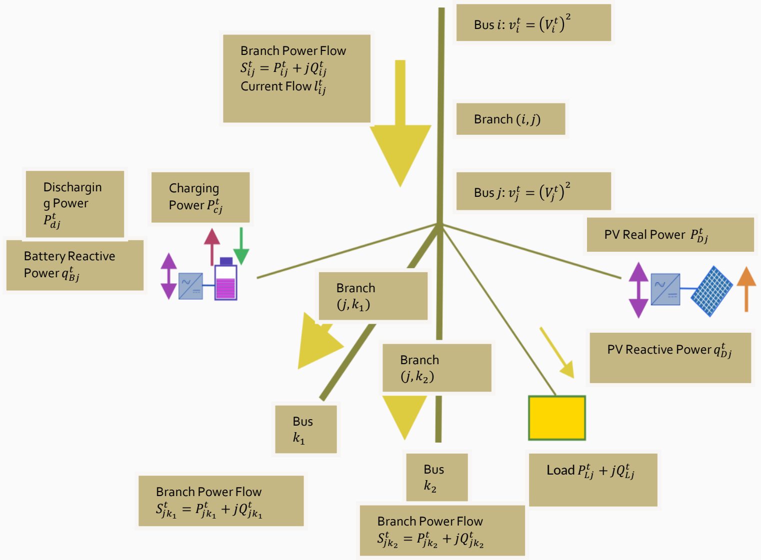
What introduces the Multi-Period Optimal Power Flow (MPOPF)? Problem?

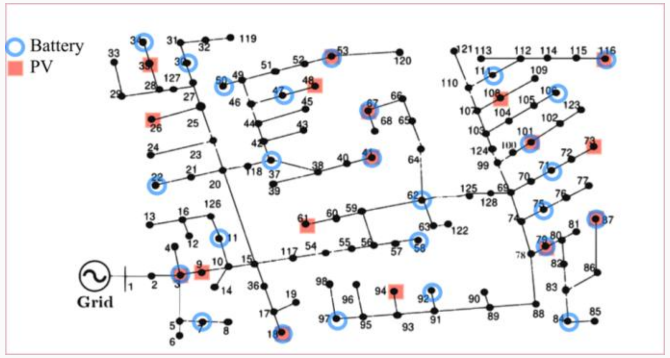
- The distribution grid is changing due to rising penetration of **Grid-Edge Devices (GEDs)** such as PVs and **Batteries**
- Eg. EVs, On-grid storage, flexible buildings

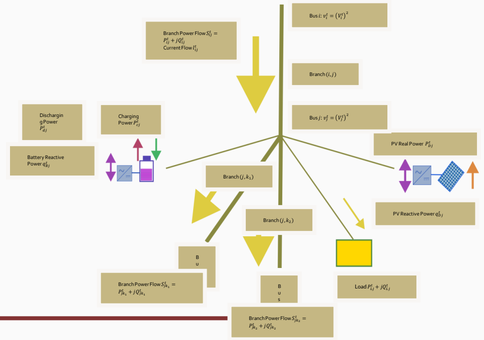
- These battery devices introduce **inter-temporal** behaviour – their actions at one time affect future states
  - This creates temporal coupling constraints
- As a result, traditional OPF (single-period) is no longer sufficient
  - The problem becomes Multi-Period OPF (MPOPF)

$$B_j^t = B_j^{t-1} + \Delta t \eta_c P_{c_j}^t - \Delta t \frac{1}{\eta_d} P_{d_j}^t$$

- MPOPF problems are much larger (size scales with **#devices x time-horizon**)
  - Leading to computational bottlenecks

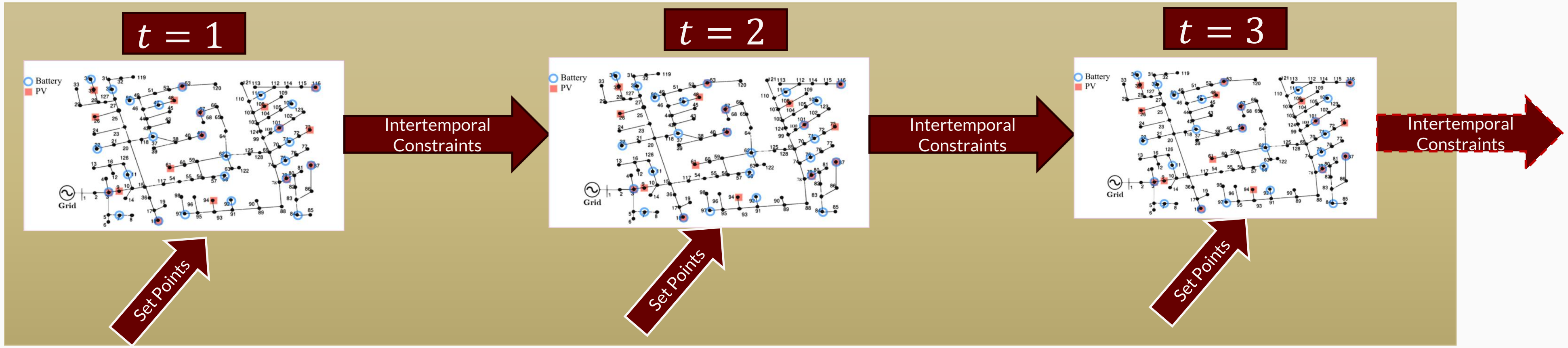






Introduction and Motivation

MPOPF Problem Visual



Rectangle sizes representative of size of each Optimization Problem

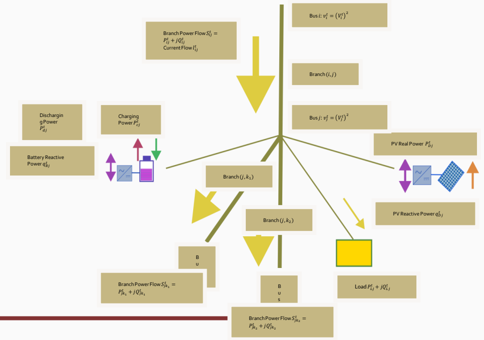
For nonlinear optimization problems, computational burden increases *superlinearly* with size!

Introduction and Motivation

Why Decomposition Approach?

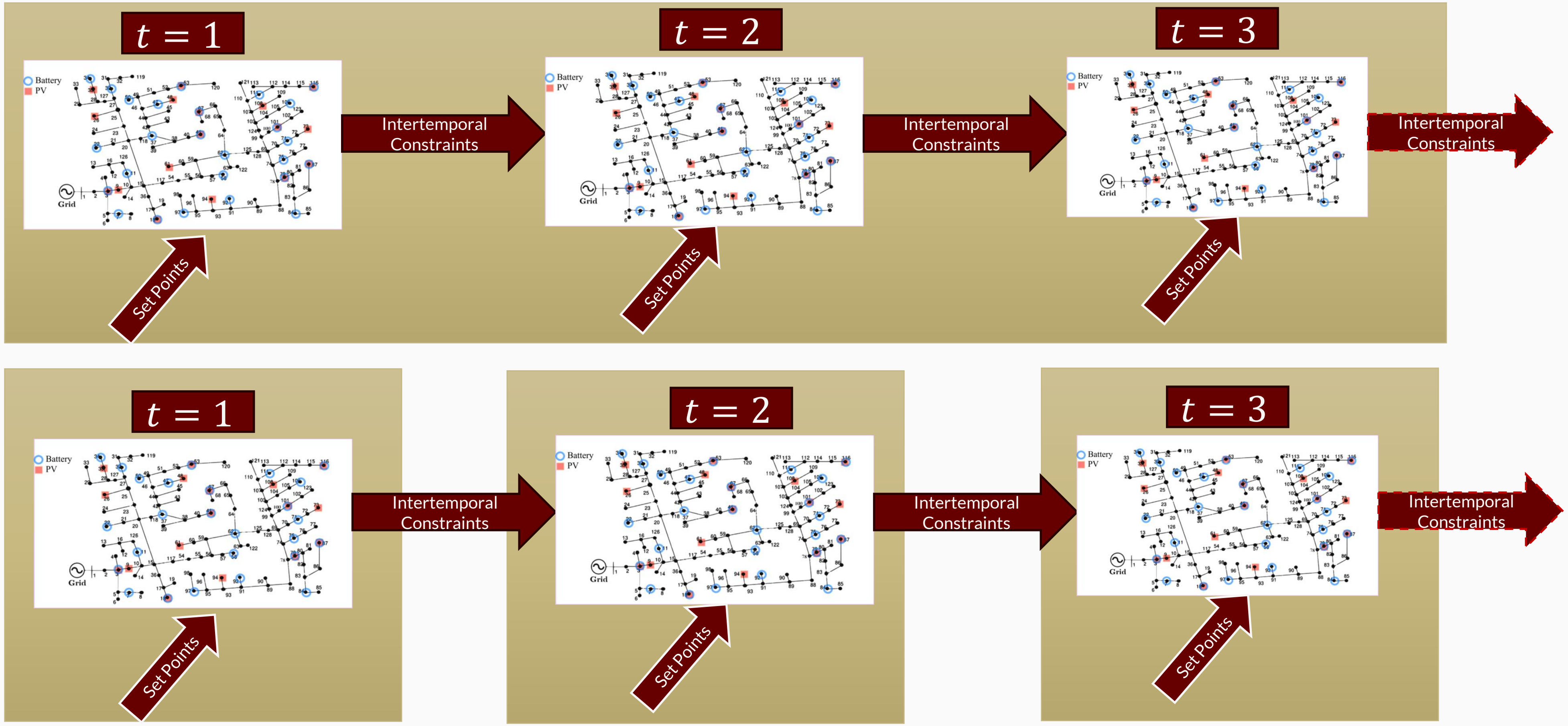
- Previously techniques like Successive Linearization [usmain] or Two-stage hierarchical optimization [Nawaf2018] have been employed for MPOPF which can miss global optimum
- Realistic modelling often requires **nonlinear** distribution system representations
  - But these are computationally expensive
- **Decomposition-based algorithms** can break MPOPF into **smaller, parallelizable subproblems**
- This enables:
  - Scalability to larger feeders and larger timescales
  - Faster or near real-time solutions
  - **Adoption by DSOs** when GED coordination becomes operationally necessary

***Accurate modelling** + **Scalable Decomposition** is the key to enabling future grid operations*



Introduction and Motivation

Why Decomposition Approach? Visual



Rectangle sizes representative of size of each Optimization Problem

For nonlinear optimization problems, computational burden increases *superlinearly* with size!

Introduction and Motivation

Intended Contributions of my PhD

- A **framework for** solving the **MPOPF** problem…
  - Which has a **systematic procedure to model** components of power distribution system in a manner faithful to their behaviour yet computationally efficient to solve for
  - That **employs** tailored **decomposition algorithms** which can exploit model's properties to come up with an even faster solution
  - Which has **provision for comparison of output** solution **with** those of **trusted softwares**, say OpenDSS
  - Whose **procedure** may be **theoretically justified**

An example of

Decomposition Algorithms Implemented

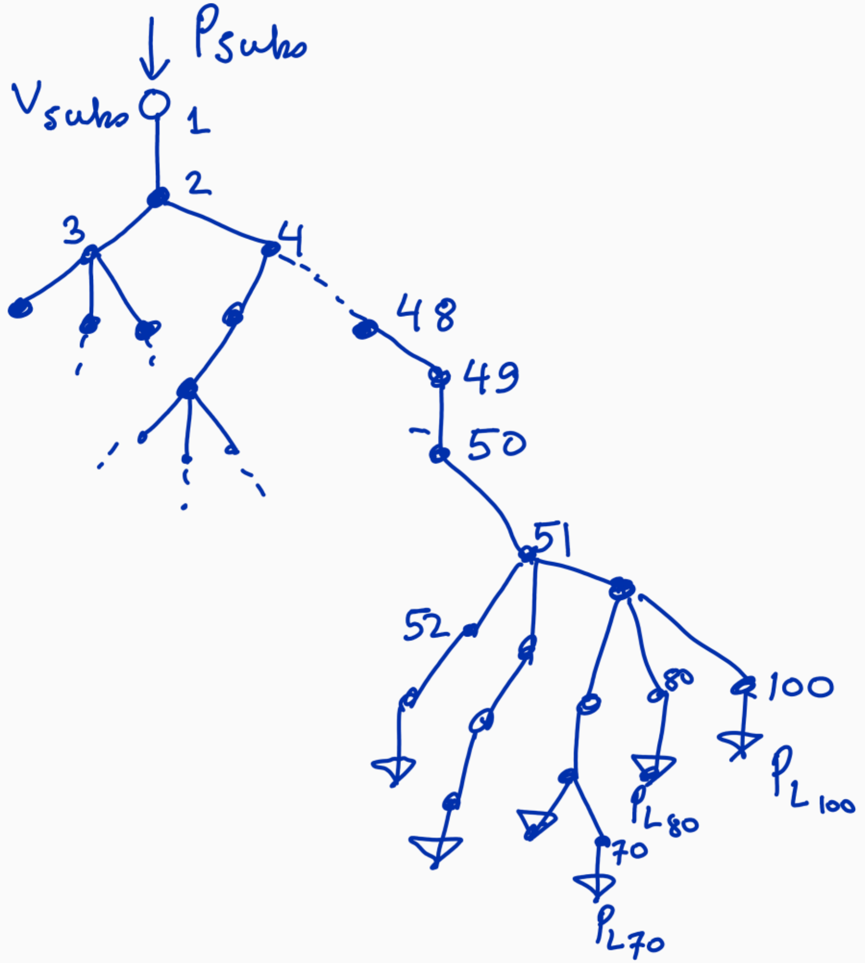
1. **MPDOPF - Spatial Decomposition** Algorithm
2. **tADMM - Temporal Decomposition** Algorithm
3. **DDP - Temporal Decomposition** Algorithm
4. Future Works

MPDOPF - Spatial Decomposition of the MPOPF problem using ENApp

- Intuition for Algorithm
- Simulation and Results

MPDOPF - Spatially Decomposition MPOPF

Intuition for Spatial Decomposition



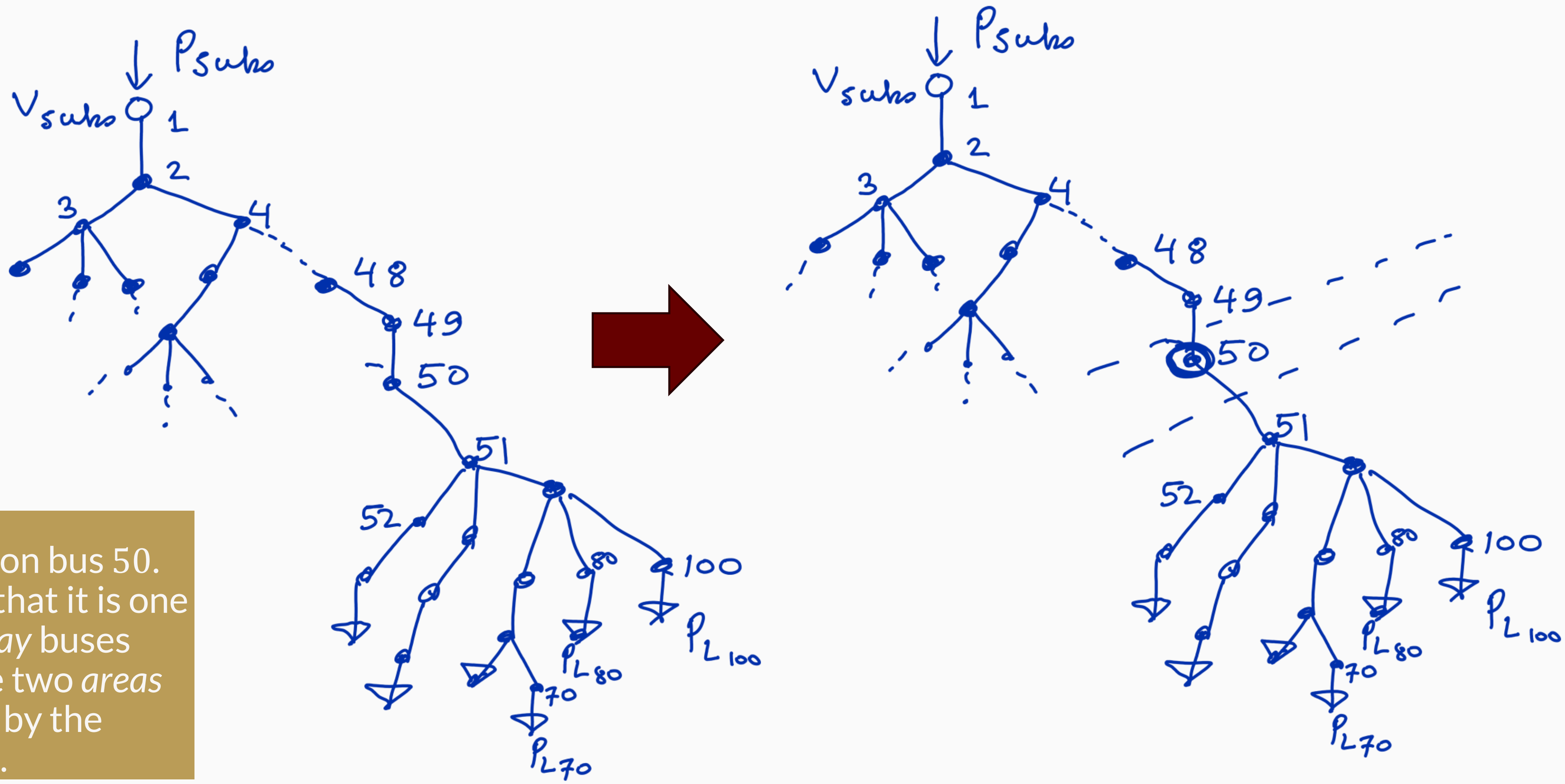
Imagine we have a 100 bus distribution system.
Substation bus is 1.
We wish to solve for its OPF, but would like to avoid solving the whole system in one go.

Let us focus on bus 50.

Source for Algorithm for OPF: ENApp [1]

MPDOPF - Spatially Decomposition MPOPF

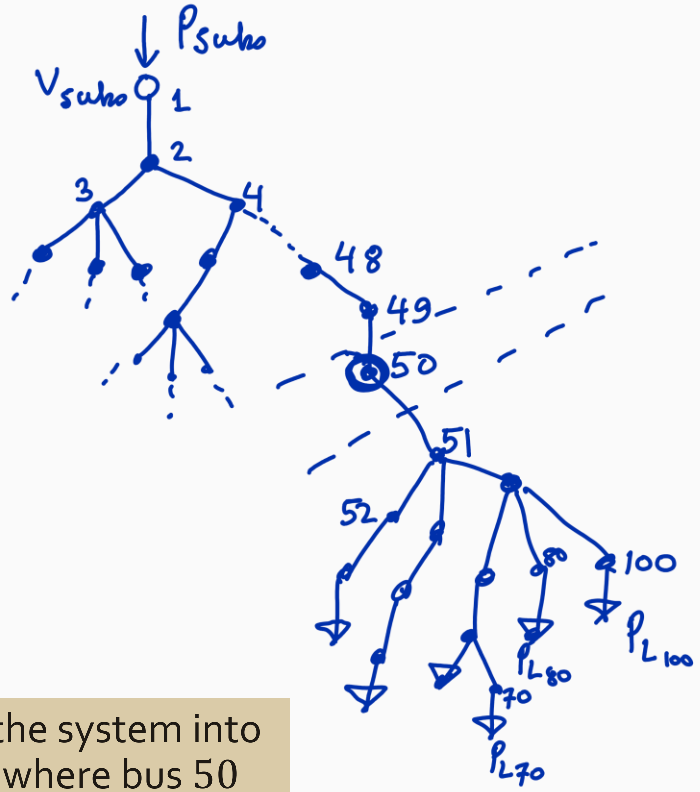
Intuition for Spatial Decomposition



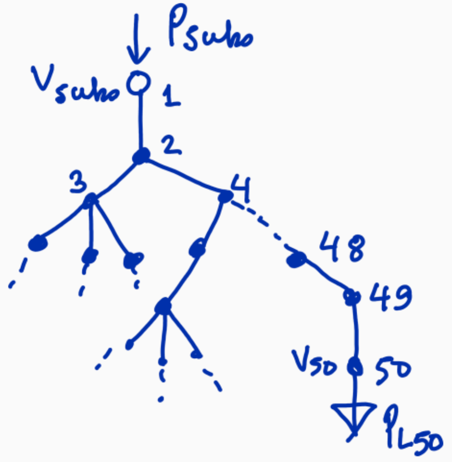
Let us focus on bus 50. We can see that it is one of the *gateway* buses between the two *areas* demarcated by the dashed lines.

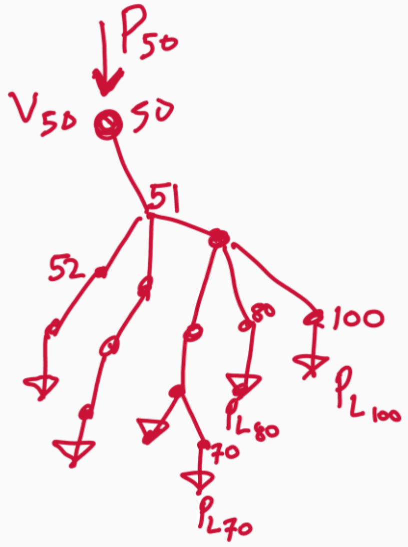
MPDOPF - Spatially Decomposition MPOPF

Intuition for Spatial Decomposition



We can divide the system into two *areas*, one where bus 50 behaves like a load node (whose load we don't know) and one where bus 50 behaves as a *substation* node (whose voltage we do don't know).





Now we have two OPF problems, each of half the size (and **less than half** the computation time) of the original problem

MPDOPF - Spatially Decomposition MPOPF

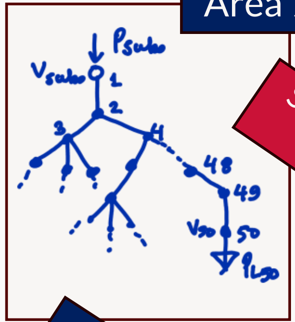
Intuition for Spatial Decomposition via OPF (single period)

Start

Each *macro-iteration* (k):

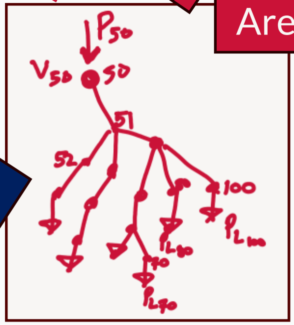
k++

Area 1



Solve OPF to get P_{50}
Set $P_{L50} = P_{50}$

Area 2



Solve OPF to get for v_{50}
Set $v_{Subs} = v_{50}$

Convergence means:

$$\left|v_{50}^{\{k+1\}} - v_{50}^{\{k\}}\right| < \epsilon$$

$$\left|S_{50}^{\{k+1\}} - S_{50}^{\{k\}}\right| < \epsilon$$

Exchange boundary variables, and continue.

No

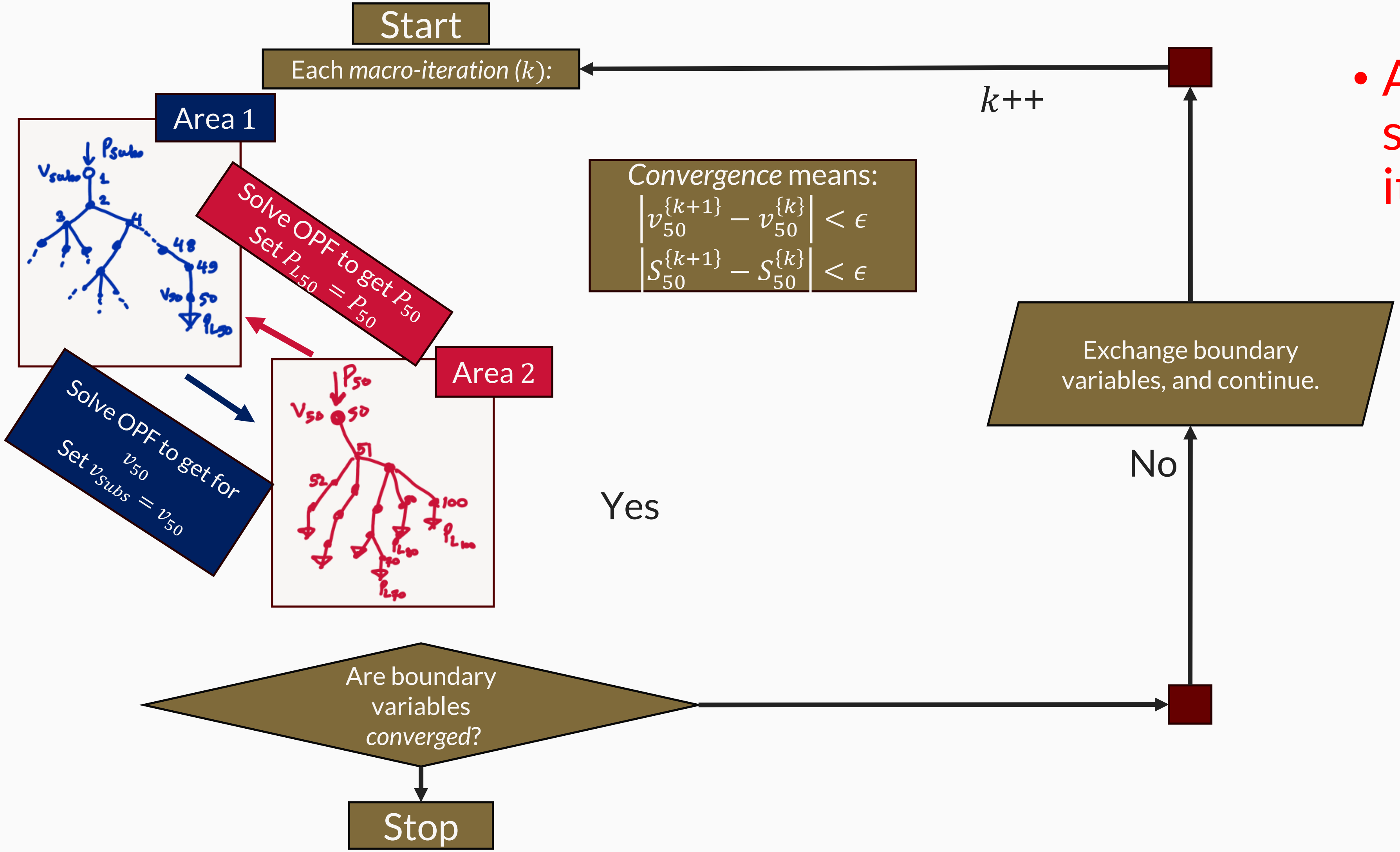
Yes

Are boundary variables *converged?*

Stop

MPDOPF - Spatially Decomposition MPOPF

Intuition for Spatial Decomposition via OPF (single period)



- Add final slide showing MPOPF iterations

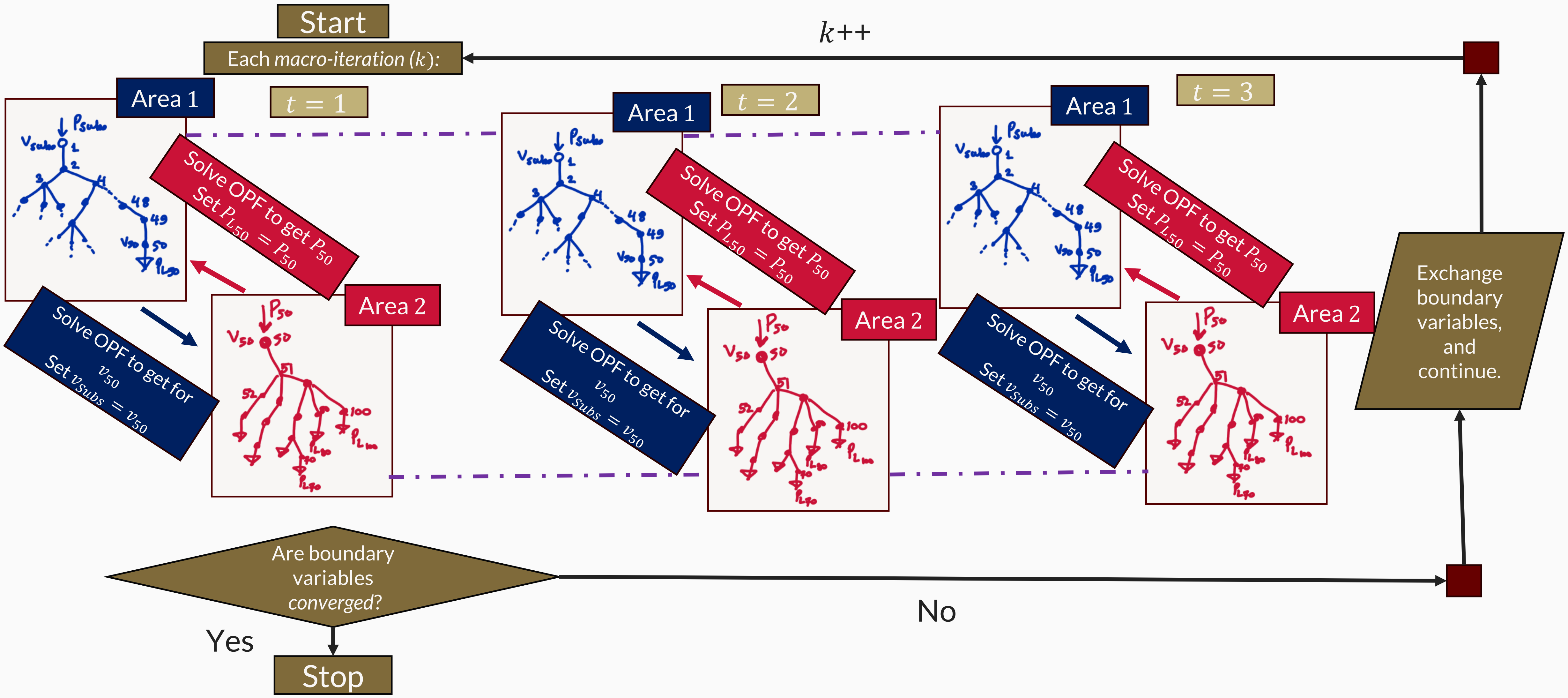
MPDOPF - Spatially Decomposition MPOPF

Spatial Decomposition workflow for MPOPF

Convergence means:

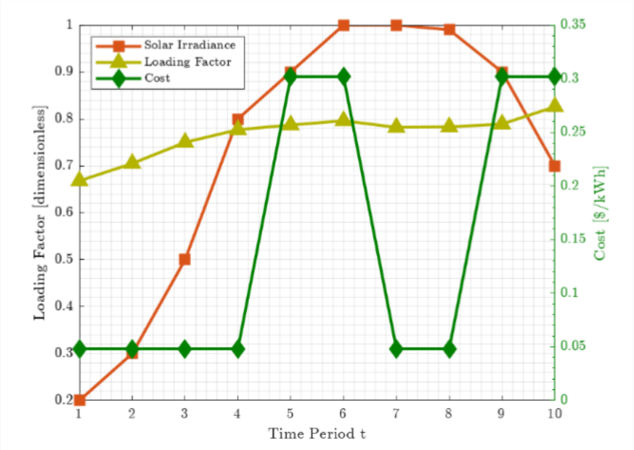
$$\left|v_{50}^{\{k+1\}} - v_{50}^{\{k\}}\right| < \epsilon$$

$$\left|S_{50}^{\{k+1\}} - S_{50}^{\{k\}}\right| < \epsilon$$



MPDOPF - Spatially Decomposition MPOPF

Key Results at a Glance – Scenario Setup



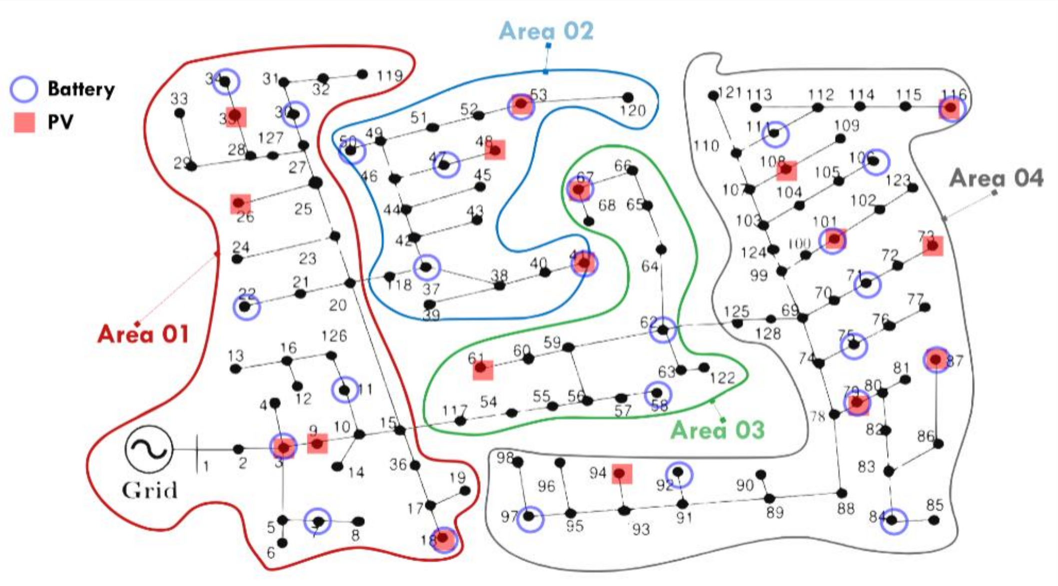
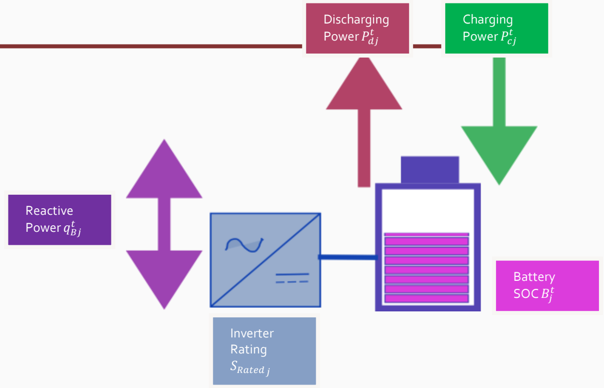


Figure 4: IEEE 123 node system divided into four areas

- $T = 10h, \Delta t = 1h$
- IEEE 123 1ph system with 17PVs and 26 Batteries divided into 4 Areas
- NLP Problem (Not SOCP relaxed) – so somewhat slow
- MATLAB fmincon+SQP

MPDOPF - Spatially Decomposition MPOPF

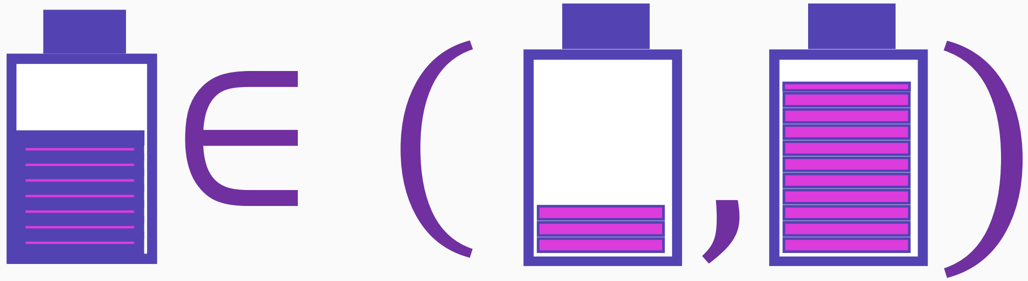
Key Results at a Glance – Scenario Setup (Battery Modelling)



Battery SOC Equation

$$B^t_j = B^{t-1}_j + \Delta t \eta_c P^t_{c_j} - \Delta t \frac{1}{\eta_d} P^t_{d_j}$$

Note: Values here are typical in OPF literature [2, 3]

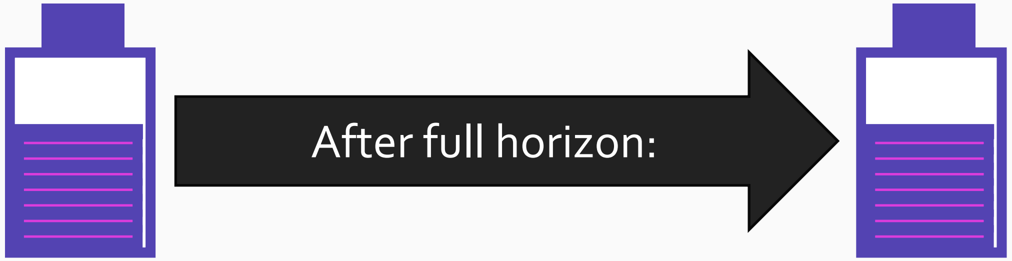


$$B^t_j \in [soc_{min}, soc_{max}] * E_{Rated,j}$$ Battery SOC Limits

$$soc_{min}, soc_{max} = 0.30, 0.95$$ Battery SOC Limits

$$\eta_c, \eta_d = 0.95, 0.95$$ Charging/Discharging Efficiencies

$$B^0_j = 0.625 E_{Rated,j}$$ Initial SOC



$$B^T_j = B^0_j$$ Terminal SOC constraint

MPDOPF - Spatially Decomposition MPOPF

Key Results at a Glance – Scenario Setup (Full NLP Model)

$$\min \sum_{t=1}^{T}\left[C^t P^t_{Subs}\Delta t + \alpha \sum_{j\in\mathcal{B}} \left\{(1-\eta_C)P^t_{C_j} + \left(\frac{1}{\eta_D}-1\right)P^t_{D_j}\right\}\right] \tag{1}$$

Subject to the constraints (2) to (12) as given below:

$$\sum_{(j,k)\in\mathcal{L}} \{P^t_{jk}\} - (P^t_{ij} - r_{ij} l^t_{ij}) = \left(P^t_{d_j} - P^t_{c_j}\right) + p^t_{D_j} - p^t_{L_j} \tag{2}$$

$$\sum_{(j,k)\in\mathcal{L}} \{Q^t_{jk}\} - (Q^t_{ij} - x_{ij} l^t_{ij}) = q^t_{D_j} + q^t_{B_j} - q^t_{L_j} \tag{3}$$

$$v^t_j = v^t_i - 2(r_{ij}P^t_{ij} + x_{ij}Q^t_{ij}) + \{r^2_{ij}+x^2_{ij}\}\, l^t_{ij} \tag{4}$$

$$(P^t_{ij})^2 + (Q^t_{ij})^2 = l^t_{ij} v^t_i \tag{5}$$

$$P^t_{Subs} \ge 0 \tag{6}$$

$$v^t_j \in [V^2_{min}, V^2_{max}] \tag{7}$$

$$q^t_{D_j} \in \left[-\sqrt{{S_{D_{R,j}}}^2 - {p^t_{D_j}}^2}, \sqrt{{S_{D_{R,j}}}^2 - {p^t_{D_j}}^2}\right] \tag{8}$$

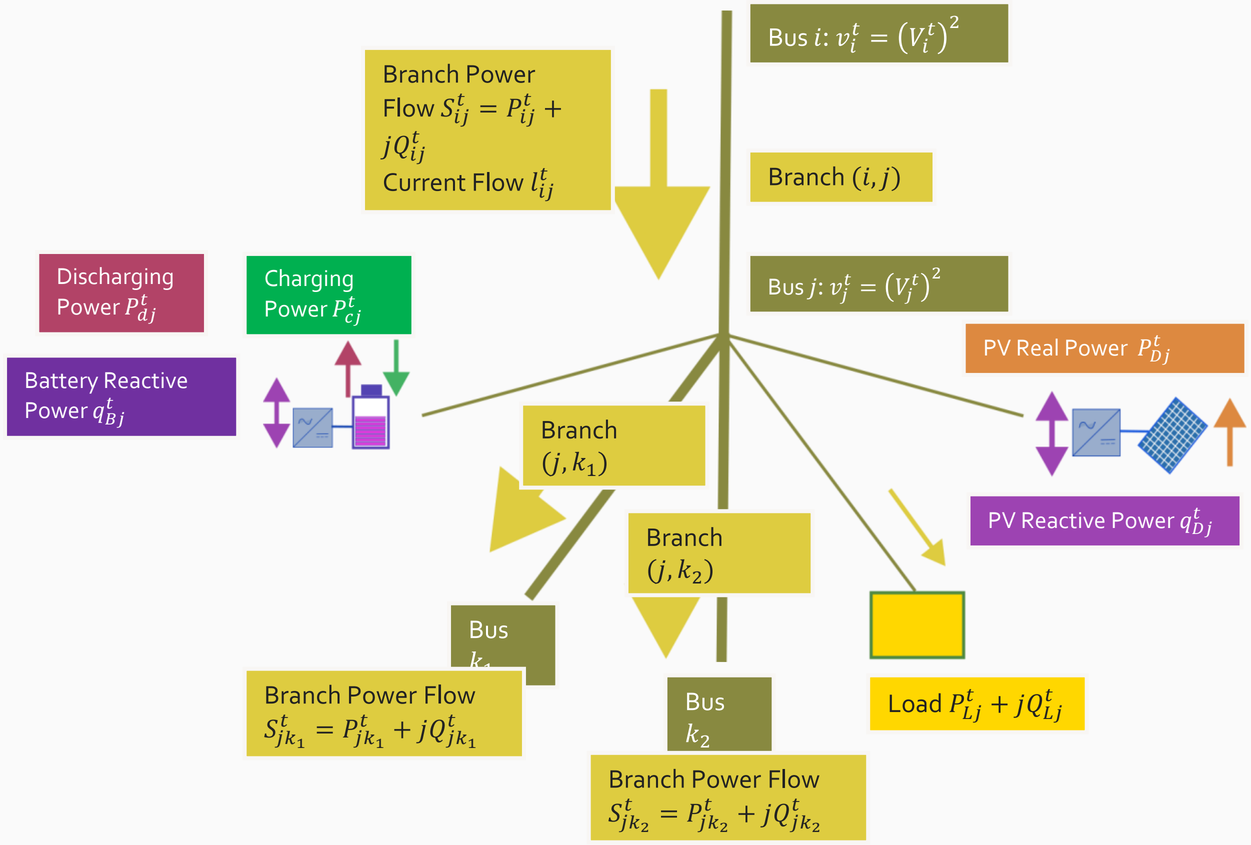
$$B^t_j = B^{t-1}_j + \Delta t \eta_c P^t_{c_j} - \Delta t \frac{1}{\eta_d} P^t_{d_j} \tag{9}$$

$$P^t_{c_j}, P^t_{d_j} \in \left[0, P_{B_{R_j}}\right], \quad B^0_j = B^T_j \tag{10}$$

$$q^t_{B_j} \in \left[-\sqrt{0.44}P_{B_{R,j}}, \sqrt{0.44}P_{B_{R,j}}\right] \tag{11}$$

$$B^t_j \in [soc_{min}B_{R,j}, soc_{max}B_{R,j}] \tag{12}$$

Branch Flow Model (Nonlinear)
used as per [bfm01-low]



MPDOPF - Spatially Decomposition MPOPF

Key Results at a Glance – Optimality and performance for IEEE 123 1ph T=10

Metric	MPCOPF	MPDOPF
Largest subproblem		
Decision variables	3150	1320
Linear constraints	5831	2451
Nonlinear constraints	635	265
Simulation results		
Substation power cost ($)	576.31	576.30
Substation real power (kW)	4308.28	4308.14
Line loss (kW)	75.99	76.12
Substation reactive power (kVAR)	574.18	656.24
PV reactive power (kVAR)	116.92	160.64
Battery reactive power (kVAR)	202.73	76.01
Computation		
Number of Iterations	-	5
Total Simulation Time (s)	521.25	49.87

Smaller Problem Size

Same Converged Objective Value

Decision Variables could be different

Appreciable Computation Speedup
(10 times) !

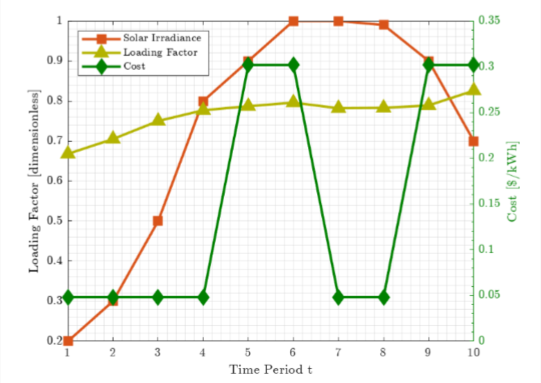
Result: Proposed MPDOPF Method cuts down the computational complexity of the MPOPF problem, making way for faster solution!

MPDOPF - Spatially Decomposition MPOPF

Key Results at a Glance – ACOPF feasibility analyses for IEEE 123 1ph T=10

Metric	MPDOPF	OpenDSS
Full horizon		
Substation real power (kW)	8544.04	8544.40
Line loss (kW)	148.94	148.87
Substation reactive power (kVAR)	1252.03	1243.36
Max. all-time discrepancy		
Voltage (pu)	0.0002	
Line loss (kW)	0.0132	
Substation power (kW)	0.4002	

Feasibility tested against OpenDSS



tADMM – temporal ADMM

- Consensus based ADMM
- Formulation
- Results

Sources: [ADMM notes CMU], [JalalKazempourYT]

tADMM – temporal ADMM

Consensus ADMM Schematic

21.4 Consensus ADMM

Consider a general problem

$$\min_x \sum_{i=1}^{B} f_i(x).$$

The consensus ADMM approach begins by reparametrizing the above problem to the following form:

$$\min_{x_1,\dots,x_B,x} \sum_{i=1}^{B} f_i(x_i) \quad s.t. \quad x_i = x \quad \forall\, i \in [B].$$

By such transformation, the updates of x_i at each ADMM step are independent and therefore can be run in parallel.

The detailed ADMM steps:

$$x_i^{(k)} = \arg\min_{x_i} f_i(x_i) + \frac{\rho}{2}\|x_i - x^{(k-1)} + w_i^{(k-1)}\|_2^2 \quad i = 1,\dots,B$$

$$x^{(k)} = \frac{1}{B}\sum_{i=1}^{B}(x_i^{(k)} + w_i^{(k-1)})$$

$$w_i^{(k)} = w_i^{(k-1)} + x_i^{(k-1)} - x^{(k)} \quad i = 1,\dots,B$$

Consensus ADMM Formulation

$$x_i^{(k)} = \arg\min_{x_i} f_i(x_i) + \frac{\rho}{2}\|x_i - x^{(k-1)} + w_i^{(k-1)}\|_2^2 \quad i = 1,\dots,B$$

$$x^{(k)} = \frac{1}{B}\sum_{i=1}^{B}(x_i^{(k)} + w_i^{(k-1)})$$

$$w_i^{(k)} = w_i^{(k-1)} + x_i^{(k-1)} - x^{(k)} \quad i = 1,\dots,B$$

Consensus ADMM Updates

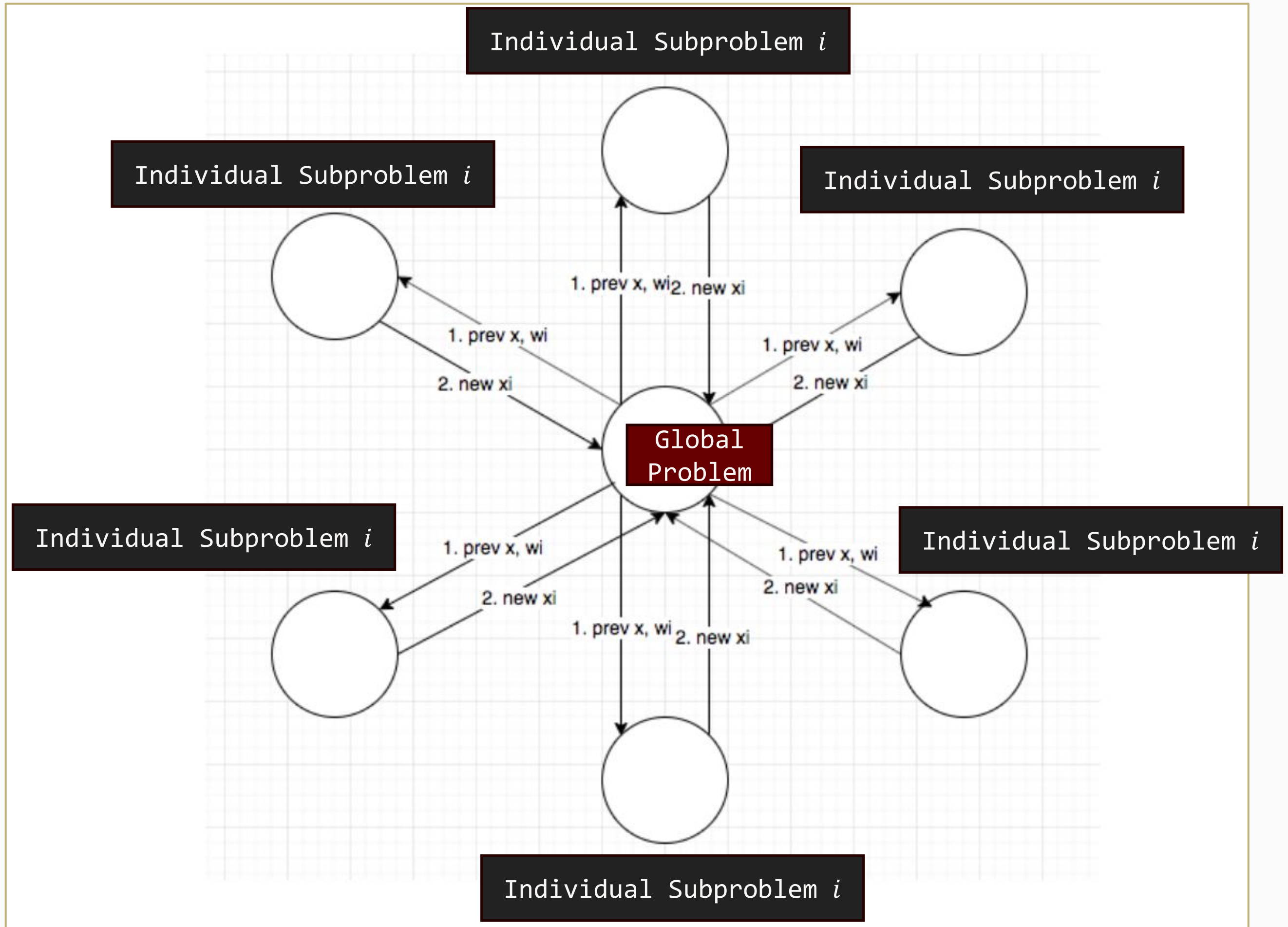
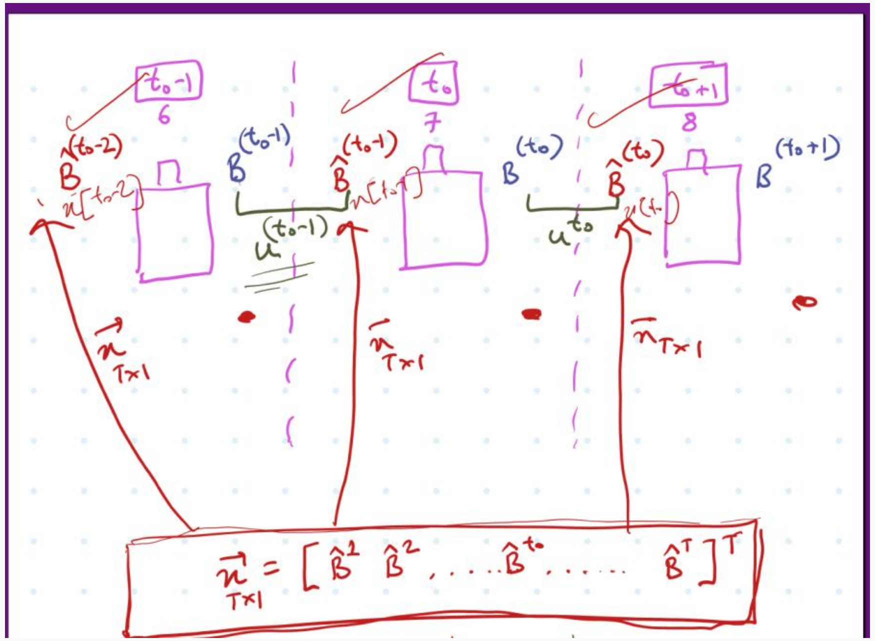


Figure 21.2: A graph illustration of consensus ADMM algorithm

tADMM – temporal ADMM

Adaptation of Consensus ADMM for the MPOPF problem



- $\mathbf{B}_j^{t_0}[\mathbf{t}]$ (Blue): Local SOC variables for battery j in subproblem t_0, evaluated at time t. These are the primal variables optimized in each subproblem.
- $\hat{\mathbf{B}}_j[\mathbf{t}]$ (Red): Global consensus SOC for battery j at time t. This represents the agreed-upon SOC trajectory that all subproblems aim to converge to.
- $\mathbf{u}_j^{t_0}[\mathbf{t}]$ (Green): Local scaled dual variables for battery j in subproblem t_0, for time t. These accumulate the consensus violation and guide convergence.

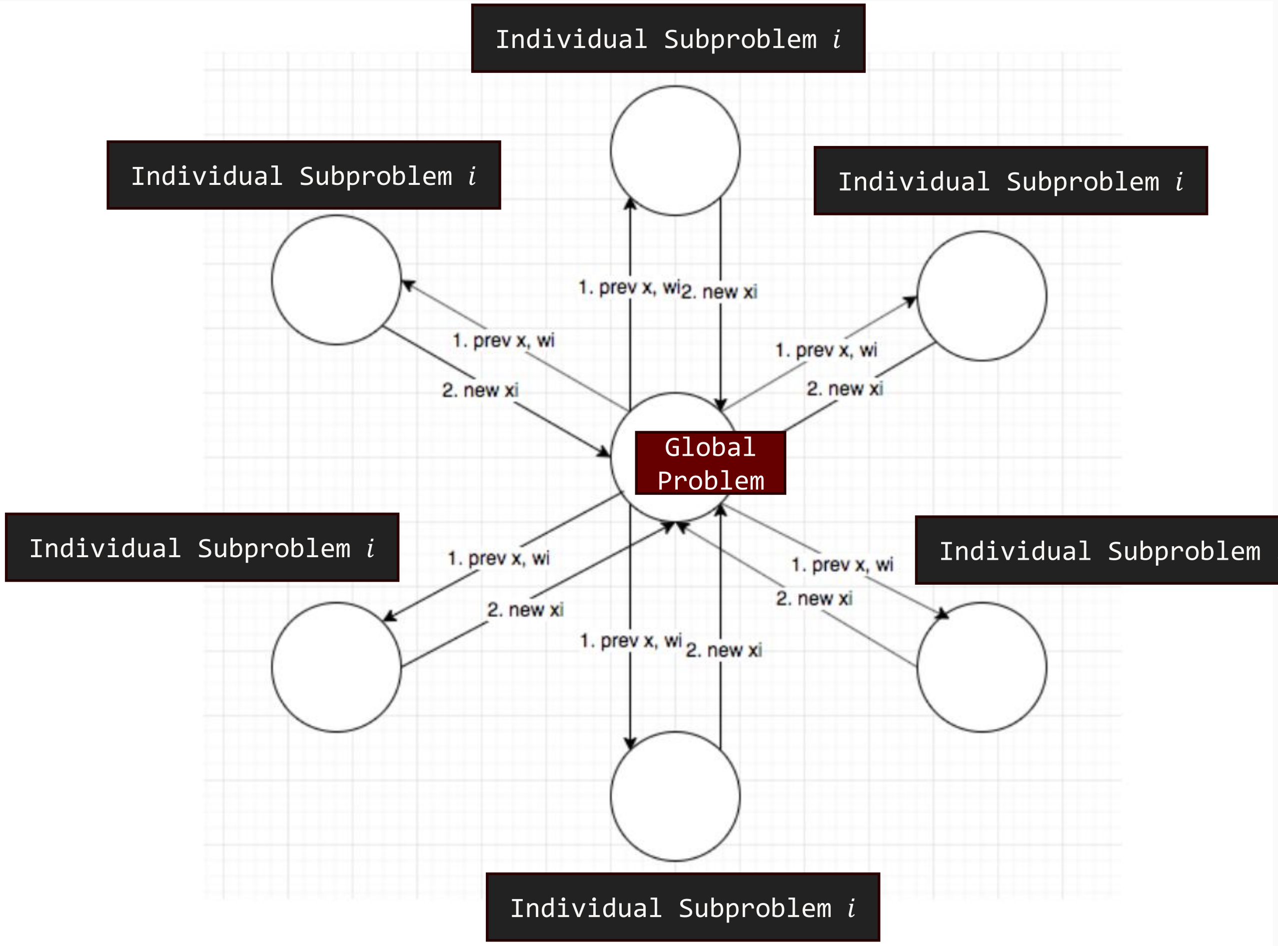


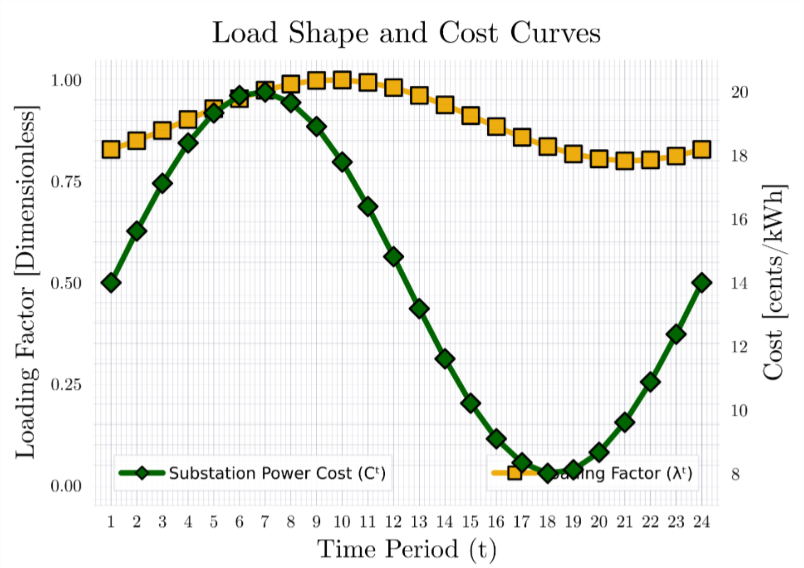
Figure 21.2: A graph illustration of consensus ADMM algorithm

Source: [ADMM notes CMU]

tADMM – temporal ADMM

Algorithm formulation for copper plate example: Update 1 Primal Update

- Copper Plate = 1 Substation + 1 Battery supplying 1 Load for T time-steps



Input Curves:
$\lambda^t, C^t \; \forall t \in \mathrm{T}$

tADMM – temporal ADMM

Algorithm formulation for copper plate example: Update 1 Primal Update

For each subproblem $t_0 \in \{1, 2, \ldots, T\}$:

$$\min_{P_{\text{subs}}^{t_0}, P_B^{t_0}, \mathbf{B}^{\mathbf{t_0}}} \quad C^{t_0} \cdot P_{\text{subs}}^{t_0} + C_B \cdot (P_B^{t_0})^2 + \frac{\rho}{2} \left\| \mathbf{B}^{\mathbf{t_0}} - \hat{\mathbf{B}} + \mathbf{u}^{\mathbf{t_0}} \right\|_2^2 \tag{50}$$

Subject to SOC Dynamics for Entire Trajectory:

$$\mathbf{B}^{\mathbf{t_0}}[\mathbf{1}] = B_0 - P_B^{t_0} \cdot \Delta t \tag{51}$$

$$\mathbf{B}^{\mathbf{t_0}}[\mathbf{t}] = \mathbf{B}^{\mathbf{t_0}}[\mathbf{t-1}] - P_B^{t_0} \cdot \Delta t, \quad \forall t \in \{2, \ldots, T\} \tag{52}$$

$$P_{\text{subs}}^{t_0} + P_B^{t_0} = P_L[t_0] \tag{53}$$

$$-P_{B,R} \le P_B^{t_0} \le P_{B,R} \tag{54}$$

$$\underline{B} \le \mathbf{B}^{\mathbf{t_0}}[\mathbf{t}] \le \overline{B}, \quad \forall t \in \{1, \ldots, T\} \tag{55}$$

- Primal Updates x T
- Each subproblem t_0 optimizes for its non-temporal constraints only
- However the temporal constraints (SOC trajectory) is computed for all time-steps
- This ensures that ADMM penalty term can compare the full trajectory $\boldsymbol{B}^{t0}$ with the consensus $\widehat{\boldsymbol{B}}$
- Uses last known values of $\widehat{\boldsymbol{B}}$ and $\boldsymbol{u}^{t0}$ and computes $\boldsymbol{B}^{t0}$

tADMM – temporal ADMM

Algorithm formulation for copper plate example: Update 2 Consensus Update

$$\hat{\mathbf{B}}[\mathrm{t}] = \mathrm{clamp}\left(\frac{1}{T}\sum_{t_0=1}^{T}\left(\mathbf{B}^{\mathbf{t_0}}[\mathrm{t}] + \mathbf{u}^{\mathbf{t_0}}[\mathrm{t}]\right), \underline{B}, \overline{B}\right) \tag{56}$$

$$\forall t \in \{1, 2, \ldots, T-1\}$$

- Consensus Update x 1
- Uses last known values of $\boldsymbol{B}^{t0}$ and $\boldsymbol{u}^{t0}$ and computes $\widehat{\boldsymbol{B}}$

The **consensus update** computes a ***negotiated*** **battery schedule** $\widehat{\boldsymbol{B}}$ by **averaging** the proposed values across all time steps

Note that **clamping** isn't analytically required, but is there **for floating point leakage catching purposes**

tADMM – temporal ADMM

Algorithm formulation for copper plate example: Update 3 Dual Update

$$\mathbf{u}^{t_0}[\mathrm{t}] := \mathbf{u}^{t_0}[\mathrm{t}] + \left(\mathbf{B}^{t_0}[\mathrm{t}] - \hat{\mathbf{B}}[\mathrm{t}]\right) \quad (59)$$
$$\forall t_0 \in \{1, \ldots, T\}, \forall t \in \{1, \ldots, T\}$$

- Dual Update x T
- Uses last known values of $\boldsymbol{B}^{t0}$ and $\widehat{\boldsymbol{B}}$ and computes $\boldsymbol{u}^{t0}$

Dual variables $\boldsymbol{u}^{t0}$ act like ***negotiation pressure***

They **get updated** in a manner which **penalizes disagreement** of a subproblem to the global consensus in a future iteration

tADMM – temporal ADMM

Algorithm formulation for copper plate example: Convergence Criteria

$$\|r^k\|_2 = \frac{1}{|\mathscr{B}|}\sqrt{\sum_{j\in\mathscr{B}}\sum_{t=1}^{T}\left(\frac{1}{T}\sum_{t_0=1}^{T}\mathrm{B}_{\mathrm{j}}^{\mathrm{t_0}}[\mathrm{t}] - \hat{\mathrm{B}}_{\mathrm{j}}[\mathrm{t}]\right)^2} \leq \epsilon_{\mathrm{pri}} \qquad (48)$$

1. Primal Residual
(Consensus Violation)

Each subproblem's own copy of SOC variables should resemble that of the global copy

$$\|s^k\|_2 = \frac{\rho}{|\mathscr{B}|}\sqrt{\sum_{j\in\mathscr{B}}\sum_{t=1}^{T}\left(\hat{\mathrm{B}}_{\mathrm{j}}^{\mathrm{k}}[\mathrm{t}] - \hat{\mathrm{B}}_{\mathrm{j}}^{\mathrm{k-1}}[\mathrm{t}]\right)^2} \leq \epsilon_{\mathrm{dual}} \qquad (49)$$

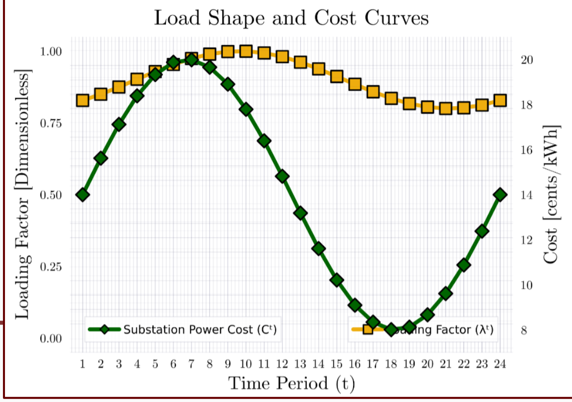
2. Dual Residual
(Consensus Change)

The consensus value should stabilize to a fixed point

Adaptive ADMM (ρ) was utilized based on relative values of these norms

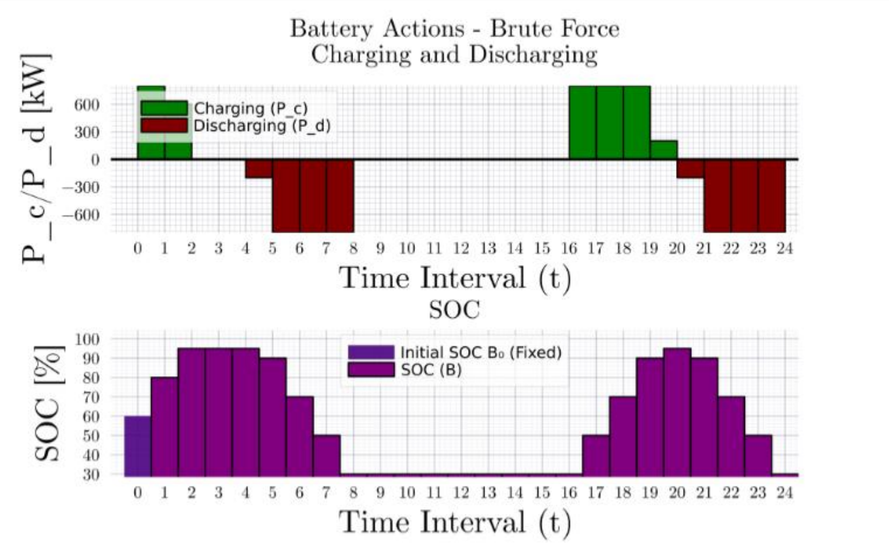
tADMM – temporal ADMM

MPOPF Simulation results for copper plate example

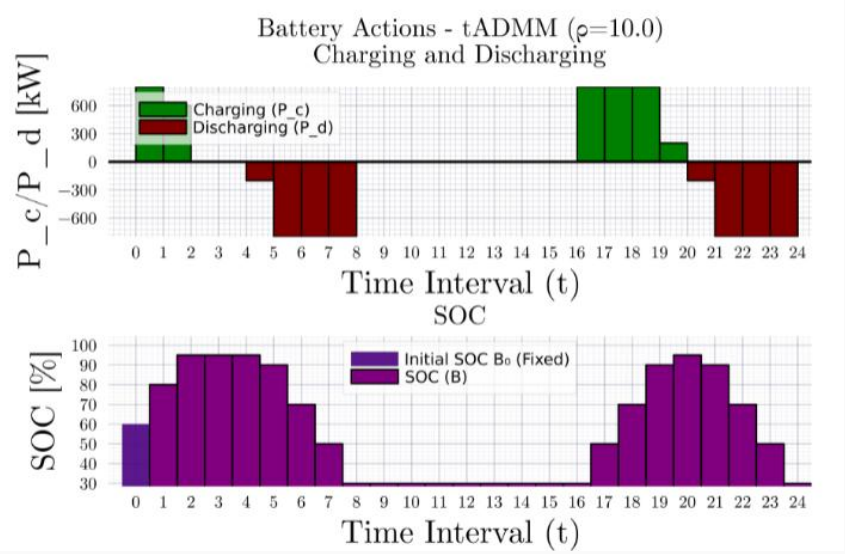


Input Curves

- **Battery Action plots** showcase that the results (control actions for battery dispatch) are identical



Brute Forced Battery Action Plots



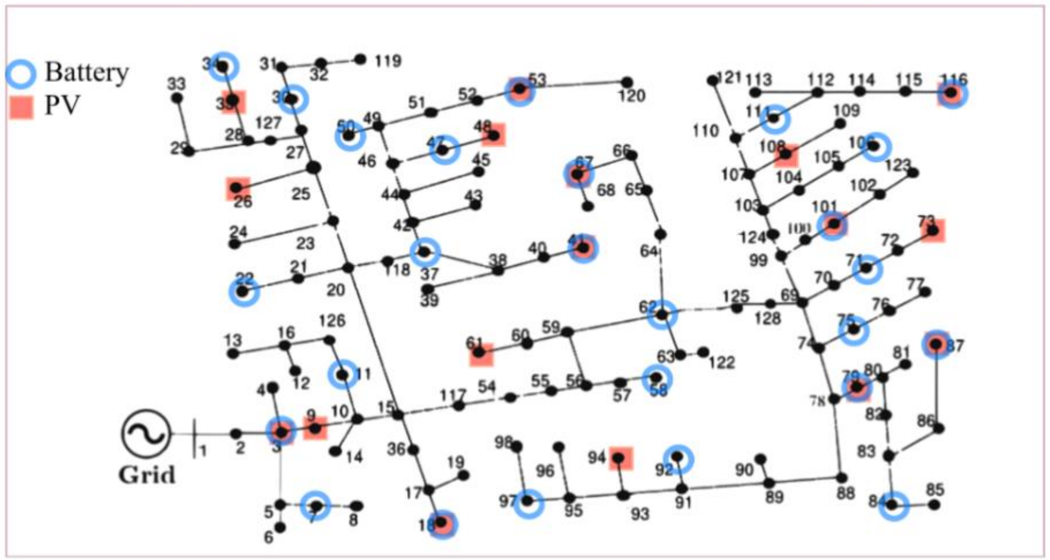
tADMM Battery Action Plots

All other results such as $P^t_{Subs}, P^t_{SubsCost}$ (not shown here) are also identical

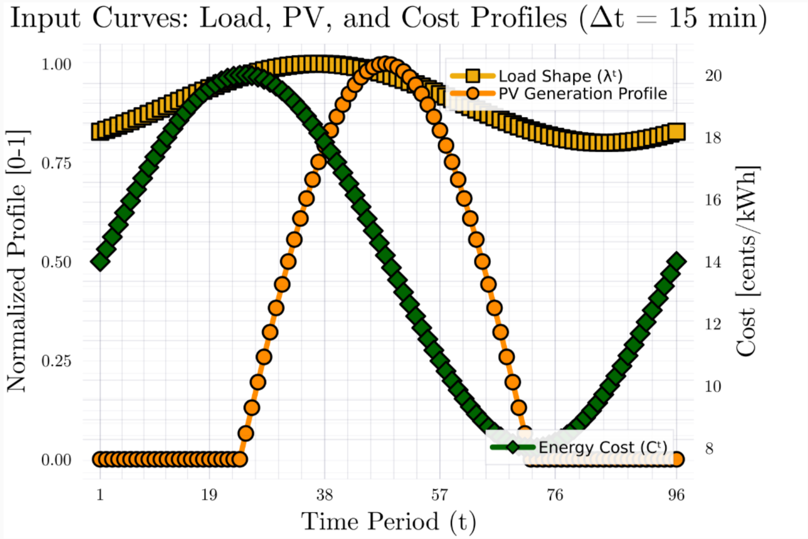
tADMM – temporal ADMM

Update modelling formulation for Full Network Model – Test Case Description

- **IEEE 123 1ph System** with **17 PVs** and **26 Batteries** – represented in both **LinDistFlow** (linear) **and BFM-NL** (nonlinear) models



IEEE123A_1ph Network



Input Curves:
$\lambda_L^t, \lambda_{PV}^t, C^t \; \forall t \in 1{:}\mathrm{T}, \mathrm{T} = 96, \Delta t = 15min$

tADMM – temporal ADMM

Update modelling formulation for Full Network Model – Update 1 Primal Update

Objective Function: Cost of Borrowed Substation Power + Cost of Battery Operation + Augmented Lagrangian Term

$$\min_{\substack{P^{t_0}_{\text{Subs}}, Q^{t_0}_{\text{Subs}}, \\ P^{t_0}_{ij}, Q^{t_0}_{ij}, v^{t_0}_{j}, q^{t_0}_{D,j}, \\ P^{t}_{B,j}, \mathbf{B}^{t_0}_{j}[t] \\ \forall j \in \mathscr{B}, t \in \mathscr{T}}} \quad c^{t_0} \cdot P^{t_0}_{\text{Subs}} \cdot P_{\text{BASE}} \cdot \Delta t + C_B \sum_{j \in \mathscr{B}} \left(P^{t_0}_{B,j}\right)^2 \cdot P^2_{\text{BASE}} \cdot \Delta t$$
$$+ \frac{\rho}{2} \sum_{j \in \mathscr{B}} \sum_{t=1}^{T} \left(\mathbf{B}^{t_0}_{j}[t] - \hat{\mathbf{B}}_{j}[t] + \mathbf{u}^{t_0}_{j}[t]\right)^2 \tag{34}$$

- Primal Updates x T
- This model is for LinDistFlow, but SOCP based BFM-NL has also been implemented by making the relevant adjustments (model not shown here)
- Uses last known values of $\widehat{\boldsymbol{B}_j}$ and $\boldsymbol{u}_j^{t0}$ and computes $\boldsymbol{B}_j^{t0}$ $\forall\, j \in B$

tADMM – temporal ADMM

Update modelling formulation for Full Network Model – Update 1 Primal Update

Objective Function: Cost of Borrowed Substation Power + Cost of Battery Operation + Augmented Lagrangian Term

$$\min_{\substack{P^{t0}_{\text{Subs}}, Q^{t0}_{\text{Subs}},\\ P^{t0}_{ij}, Q^{t0}_{ij}, v^{t0}_j, q^{t0}_{D,j},\\ P^{t}_{B,j}, \mathbf{B}^{t0}_{j}[t]\\ \forall j \in \mathscr{B}, t \in \mathscr{T}}} \quad c^{t_0} \cdot P^{t_0}_{\text{Subs}} \cdot P_{\text{BASE}} \cdot \Delta t + C_B \sum_{j \in \mathscr{B}} \left(P^{t_0}_{B,j}\right)^2 \cdot P^2_{\text{BASE}} \cdot \Delta t + \frac{\rho}{2} \sum_{j \in \mathscr{B}} \sum_{t=1}^{T} \left(\mathbf{B}^{t_0}_{j}[t] - \hat{\mathbf{B}}_{j}[t] + \mathbf{u}^{t_0}_{j}[t]\right)^2 \qquad (34)$$

t_0 specific Network Constraints: Temporally Uncoupled

$$\text{Real power balance (substation):} \quad P^{t_0}_{\text{Subs}} - \sum_{(1,j) \in \mathscr{L}_1} P^{t_0}_{1j} = 0 \qquad (35)$$

$$\text{Real power balance (nodes):} \quad P^{t_0}_{ij} - \sum_{(j,k) \in \mathscr{L}} P^{t_0}_{jk} = P^{t_0}_{B,j} + p^{t_0}_{D,j} - p^{t_0}_{L,j}, \quad \forall (i,j) \in \mathscr{L}, \qquad (36)$$

$$\text{Reactive power balance (substation):} \quad Q^{t_0}_{\text{Subs}} - \sum_{(1,j) \in \mathscr{L}_1} Q^{t_0}_{1j} = 0 \qquad (37)$$

$$\text{Reactive power balance (nodes):} \quad Q^{t_0}_{ij} - \sum_{(j,k) \in \mathscr{L}} Q^{t_0}_{jk} = q^{t_0}_{D,j} - q^{t_0}_{L,j}, \quad \forall (i,j) \in \mathscr{L}, \qquad (38)$$

$$\text{KVL constraints:} \quad v^{t_0}_i - v^{t_0}_j = 2(r_{ij} P^{t_0}_{ij} + x_{ij} Q^{t_0}_{ij}), \quad \forall (i,j) \in \mathscr{L} \qquad (39)$$

$$\text{Voltage limits:} \quad (V_{\min,j})^2 \le v^{t_0}_j \le (V_{\max,j})^2, \quad \forall j \in \mathscr{N} \qquad (40)$$

$$\text{PV reactive limits:} \quad -\sqrt{(S_{D,j})^2 - (p^{t_0}_{D,j})^2} \le q^{t_0}_{D,j} \le \sqrt{(S_{D,j})^2 - (p^{t_0}_{D,j})^2}, \quad \forall j \in \mathscr{D} \qquad (41)$$

- Primal Updates x T
- This model is for LinDistFlow, but SOCP based BFM-NL has also been implemented by making the relevant adjustments (model not shown here)
- Uses last known values of $\widehat{B_j}$ and u_j^{t0} and computes B_j^{t0} $\forall\, j \in B$

tADMM – temporal ADMM

Update modelling formulation for Full Network Model – Update 1 Primal Update

Objective Function: Cost of Borrowed Substation Power + Cost of Battery Operation + Augmented Lagrangian Term

$$\min_{\substack{P^{t_0}_{\text{Subs}}, Q^{t_0}_{\text{Subs}}, \\ P^{t_0}_{ij}, Q^{t_0}_{ij}, v^{t_0}_j, q^{t_0}_{D,j}, \\ P^t_{B,j}, \mathbf{B}^{t_0}_j[t] \\ \forall j \in \mathscr{B}, t \in \mathscr{T}}} c^{t_0} \cdot P^{t_0}_{\text{Subs}} \cdot P_{\text{BASE}} \cdot \Delta t + C_B \sum_{j \in \mathscr{B}} \left(P^{t_0}_{B,j}\right)^2 \cdot P^2_{\text{BASE}} \cdot \Delta t$$

$$+ \frac{\rho}{2} \sum_{j \in \mathscr{B}} \sum_{t=1}^{T} \left(\mathbf{B}^{t_0}_j[t] - \hat{\mathbf{B}}_j[t] + \mathbf{u}^{t_0}_j[t]\right)^2 \tag{34}$$

t_0 specific Network Constraints: Temporally Uncoupled

Real power balance (substation): $P^{t_0}_{\text{Subs}} - \sum_{(1,j) \in \mathscr{L}_1} P^{t_0}_{1j} = 0$ (35)

Real power balance (nodes): $P^{t_0}_{ij} - \sum_{(j,k) \in \mathscr{L}} P^{t_0}_{jk} = P^{t_0}_{B,j} + p^{t_0}_{D,j} - p^{t_0}_{L,j}$,

$\forall (i,j) \in \mathscr{L}$, (36)

Reactive power balance (substation): $Q^{t_0}_{\text{Subs}} - \sum_{(1,j) \in \mathscr{L}_1} Q^{t_0}_{1j} = 0$ (37)

Reactive power balance (nodes): $Q^{t_0}_{ij} - \sum_{(j,k) \in \mathscr{L}} Q^{t_0}_{jk} = q^{t_0}_{D,j} - q^{t_0}_{L,j}$,

$\forall (i,j) \in \mathscr{L}$, (38)

KVL constraints: $v^{t_0}_i - v^{t_0}_j = 2(r_{ij} P^{t_0}_{ij} + x_{ij} Q^{t_0}_{ij}), \quad \forall (i,j) \in \mathscr{L}$ (39)

Voltage limits: $(V_{\min,j})^2 \le v^{t_0}_j \le (V_{\max,j})^2, \quad \forall j \in \mathscr{N}$ (40)

PV reactive limits: $-\sqrt{(S_{D,j})^2 - (p^{t_0}_{D,j})^2} \le q^{t_0}_{D,j} \le \sqrt{(S_{D,j})^2 - (p^{t_0}_{D,j})^2}$,

$\forall j \in \mathscr{D}$ (41)

- Primal Updates x T
- This model is for LinDistFlow, but SOCP based BFM-NL has also been implemented by making the relevant adjustments (model not shown here)
- Uses last known values of $\widehat{B_j}$ and u_j^{t0} and computes B_j^{t0} $\forall\, j \in B$

Shared Temporally Coupled Constraints

Initial SOC: $\mathbf{B}^{t_0}_j[1] = B_{0,j} - P^1_{B,j} \cdot \Delta t, \quad \forall j \in \mathscr{B}$ (42)

SOC trajectory: $\mathbf{B}^{t_0}_j[t] = \mathbf{B}^{t_0}_j[t-1] - P^t_{B,j} \cdot \Delta t, \quad \forall t \in \{2, \ldots, T\}, j \in \mathscr{B}$ (43)

SOC limits: $\text{SOC}_{\min,j} \cdot B_{\text{rated},j} \le \mathbf{B}^{t_0}_j[t] \le \text{SOC}_{\max,j} \cdot B_{\text{rated},j}$,

$\forall t \in \mathscr{T}, j \in \mathscr{B}$ (44)

Power limits: $-P_{B,\text{rated},j} \le P^t_{B,j} \le P_{B,\text{rated},j}, \quad \forall t \in \mathscr{T}, j \in \mathscr{B}$ (45)

tADMM – temporal ADMM

Update modelling formulation for Full Network Model – Update 1 Primal Update

Objective Function: Cost of Borrowed Substation Power + Cost of Battery Operation + Augmented Lagrangian Term

$$\min_{\substack{P^{t_0}_{\mathrm{Subs}},Q^{t_0}_{\mathrm{Subs}},\\ P^{t_0}_{ij},Q^{t_0}_{ij},v^{t_0}_{j},q^{t_0}_{D,j},\\ P^{t}_{B,j},\mathbf{B}^{\mathbf{to}}_{\mathbf{j}}[\mathrm{t}]\\ \forall j\in\mathscr{B},\, t\in\mathscr{T}}} \quad c^{t_0}\cdot P^{t_0}_{\mathrm{Subs}}\cdot P_{\mathrm{BASE}}\cdot\Delta t + C_B\sum_{j\in\mathscr{B}}\left(P^{t_0}_{B,j}\right)^2\cdot P^2_{\mathrm{BASE}}\cdot\Delta t$$
$$+\frac{\rho}{2}\sum_{j\in\mathscr{B}}\sum_{t=1}^{T}\left(\mathbf{B}^{\mathbf{to}}_{\mathbf{j}}[\mathrm{t}]-\hat{\mathbf{B}}_{\mathbf{j}}[\mathrm{t}]+\mathbf{u}^{\mathbf{to}}_{\mathbf{j}}[\mathrm{t}]\right)^2 \tag{34}$$

- This is an initial formulation where the **entire SOC trajectory** is included in the augmented Lagrangian and consensus updates

- Although this introduces **additional shared linear variables**, their **impact on computational complexity** is **small** compared to the nonlinear OPF components

- This is a **conservative starting point**; ongoing work will **reduce the number of shared variables** to improve scalability

- Primal Updates x T
- This model is for LinDistFlow, but SOCP based BFM-NL has also been implemented by making the relevant adjustments (model not shown here)
- Uses last known values of $\widehat{B_j}$ and u_j^{t0} and computes B_j^{t0} $\forall\, j \in B$

Shared Temporally Coupled Constraints

$$\text{Initial SOC:}\quad \mathbf{B}^{\mathbf{to}}_{\mathbf{j}}[\mathbf{1}] = B_{0,j} - P^{1}_{B,j}\cdot\Delta t,\quad \forall j\in\mathscr{B} \tag{42}$$
$$\text{SOC trajectory:}\quad \mathbf{B}^{\mathbf{to}}_{\mathbf{j}}[\mathrm{t}] = \mathbf{B}^{\mathbf{to}}_{\mathbf{j}}[\mathrm{t}-\mathbf{1}] - P^{t}_{B,j}\cdot\Delta t,\quad \forall t\in\{2,\ldots,T\},\ j\in\mathscr{B} \tag{43}$$
$$\text{SOC limits:}\quad \mathrm{SOC}_{\min,j}\cdot B_{\mathrm{rated},j}\le \mathbf{B}^{\mathbf{to}}_{\mathbf{j}}[\mathrm{t}]\le \mathrm{SOC}_{\max,j}\cdot B_{\mathrm{rated},j},\quad \forall t\in\mathscr{T},\ j\in\mathscr{B} \tag{44}$$
$$\text{Power limits:}\quad -P_{B,\mathrm{rated},j}\le P^{t}_{B,j}\le P_{B,\mathrm{rated},j},\quad \forall t\in\mathscr{T},\ j\in\mathscr{B} \tag{45}$$

tADMM – temporal ADMM

Update modelling formulation for Full Network Model – Updates 2 and 3

$$\hat{\mathrm{B}}_{\mathrm{j}}[\mathrm{t}] = \mathrm{clamp}\left(\frac{1}{T}\sum_{t_0=1}^{T}\left(\mathrm{B}_{\mathrm{j}}^{\mathrm{to}}[\mathrm{t}] + \mathrm{u}_{\mathrm{j}}^{\mathrm{to}}[\mathrm{t}]\right), \underline{B}_j, \overline{B}_j\right) \tag{46}$$

- Consensus Updates x n_B
- Uses last known values of $\boldsymbol{B}_j^{t0}$ and $\boldsymbol{u}_j^{t0}$ and computes $\widehat{\boldsymbol{B}_j}\ \forall\ j \in B$

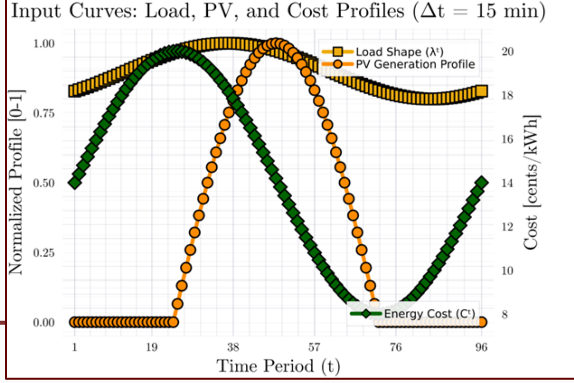
$$\mathrm{u}_{\mathrm{j}}^{\mathrm{to}}[\mathrm{t}] := \mathrm{u}_{\mathrm{j}}^{\mathrm{to}}[\mathrm{t}] + \left(\mathrm{B}_{\mathrm{j}}^{\mathrm{to}}[\mathrm{t}] - \hat{\mathrm{B}}_{\mathrm{j}}[\mathrm{t}]\right) \tag{47}$$

- Dual Updates x n_B
- Uses last known values of $\boldsymbol{B}_j^{t0}$ and $\widehat{\boldsymbol{B}_j}$ and computes $\boldsymbol{u}_j^{t0}\ \forall\ j \in B$

Both updates are pretty much the same as that for copper plate example, only differing in having to account for multiple batteries

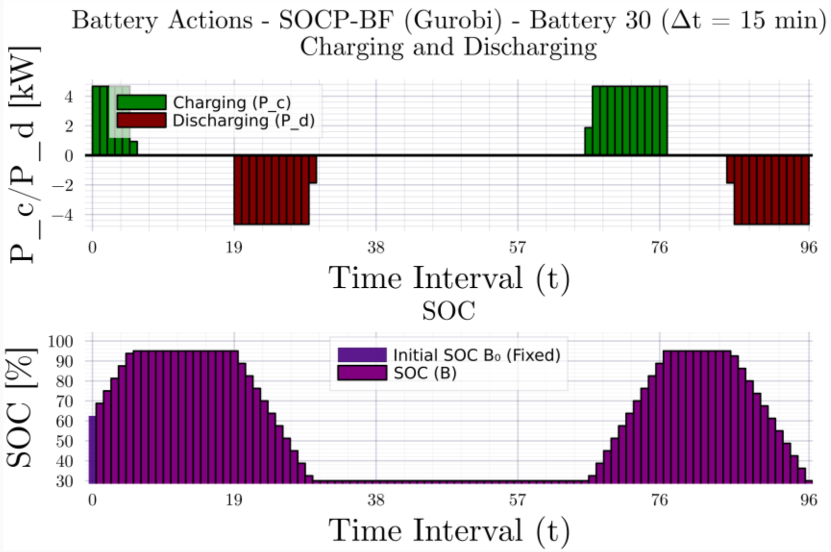
tADMM – temporal ADMM

MPOPF Simulation results for IEEE123A_1ph – Battery Actions

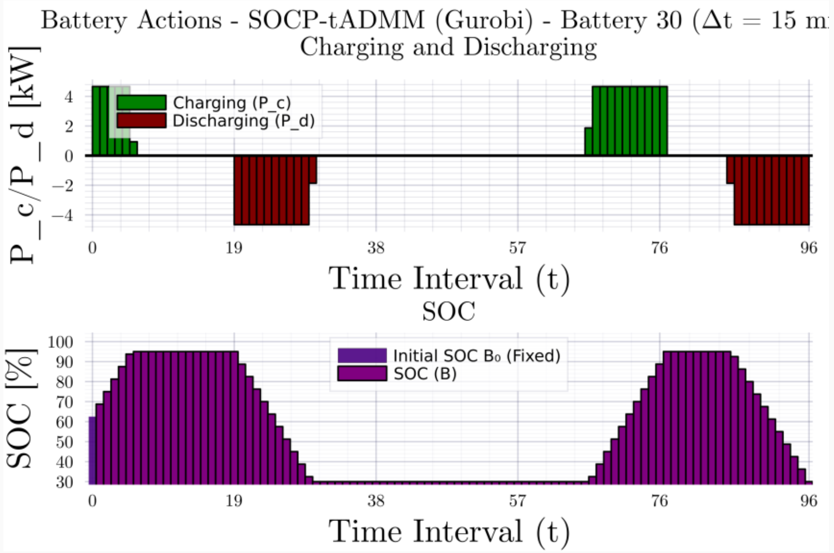


Input Curves

- These results are for nonlinear model (SOCP relaxed BFM-NL)
- **Battery Action plots** showcase that the results (control actions for battery dispatch) are identical



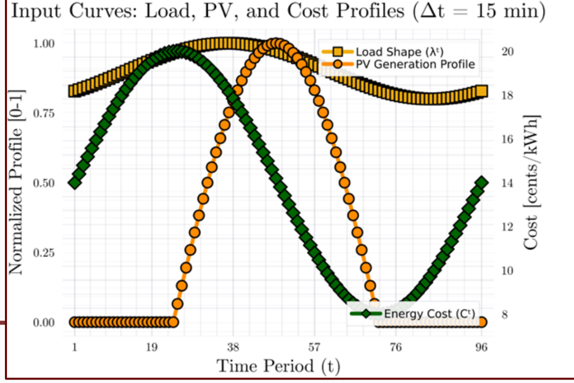
Brute Forced Battery Action Plots



tADMM Battery Action Plots

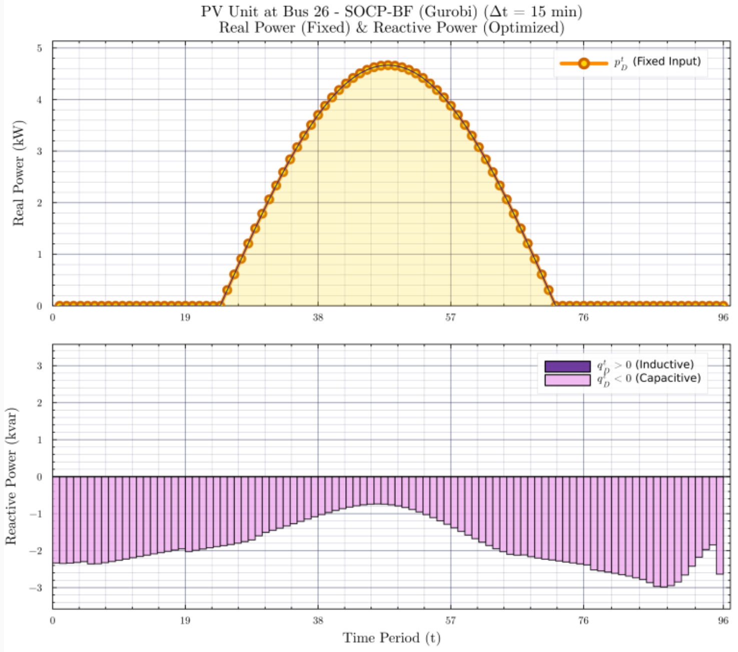
tADMM – temporal ADMM

MPOPF Simulation results for IEEE123A_1ph – PV Actions

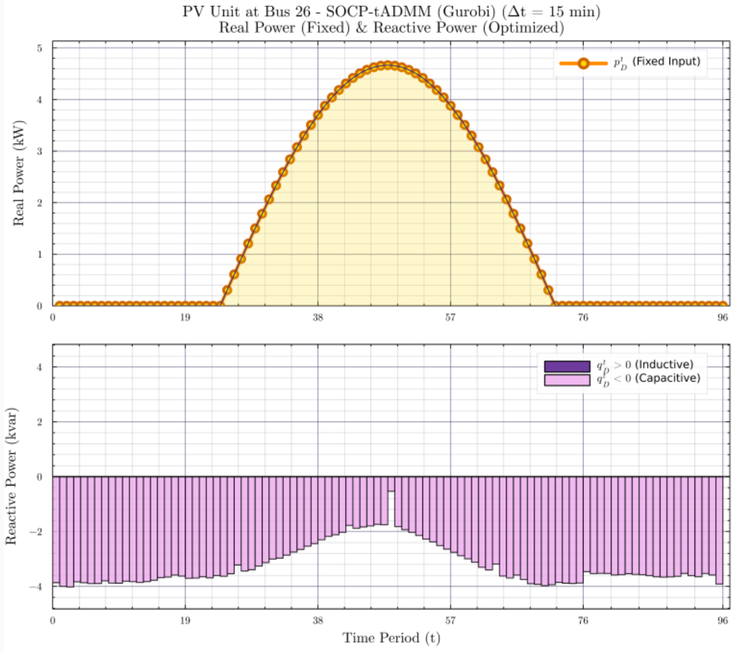


Input Curves

- **PV Reactive Power Dispatch plots** showcase are generally NOT necessarily identical



Note that PV real power dispatch is a fixed variable dependent only on λ^t_{PV} (fixed variable)

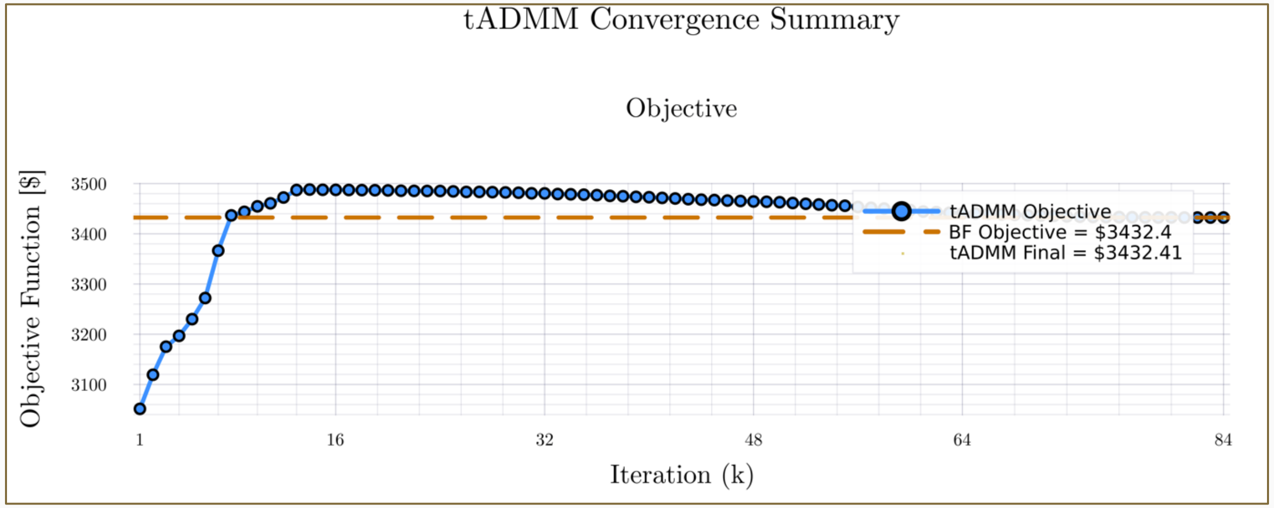


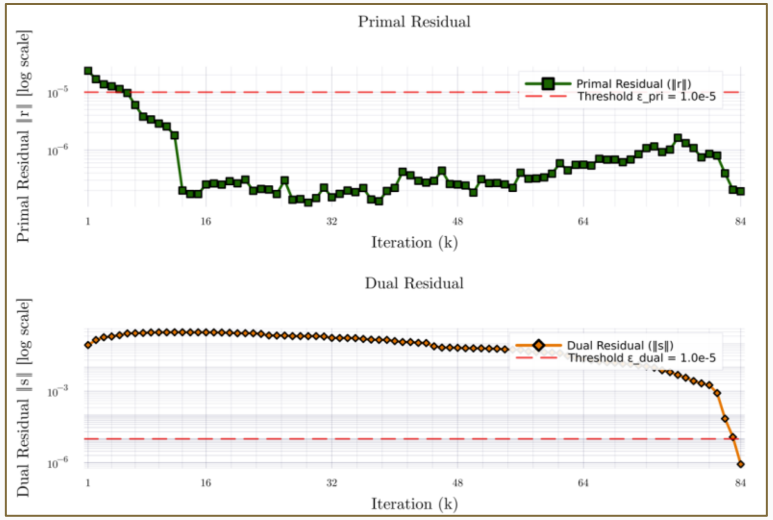
Brute Forced PV Action Plots

tADMM PV Action Plots

tADMM – temporal ADMM

MPOPF Simulation results for IEEE123A_1ph – Convergence





Convergence of Objective Function:
tADMM (Blue) vs **Brute Forced** (Orange)

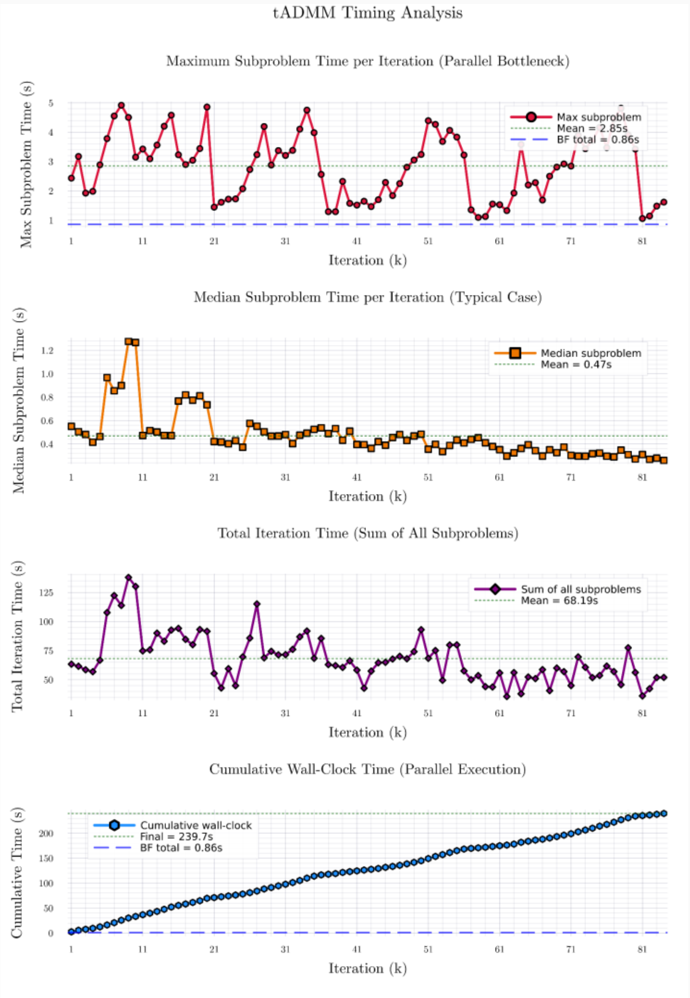
Convergence of norms:
r-norm (Green) vs **s-norm** (Orange)

--- tADMM SUBPROBLEM SIZE (one time period) ---
Number of variables: 5520
Linear constraints: 8145
Quadratic constraints (SOCP): 127
Nonlinear constraints: 0
Total constraints: 8272

--- BRUTE FORCE PROBLEM SIZE ---
Number of variables: 55680
Linear constraints: 70560
Quadratic constraints (SOCP): 12192
Nonlinear constraints: 0
Total constraints: 82752

tADMM – temporal ADMM

A temporal decomposition algorithm for MPOPF



Currently tADMM is converging slower than Brute-forced optimization which shouldn't be the case since typically solution times of nonlinear problems scale superlinearly with problem size.

We feel that this might either be due to some poor (programmatical) optimization or some other bugs.

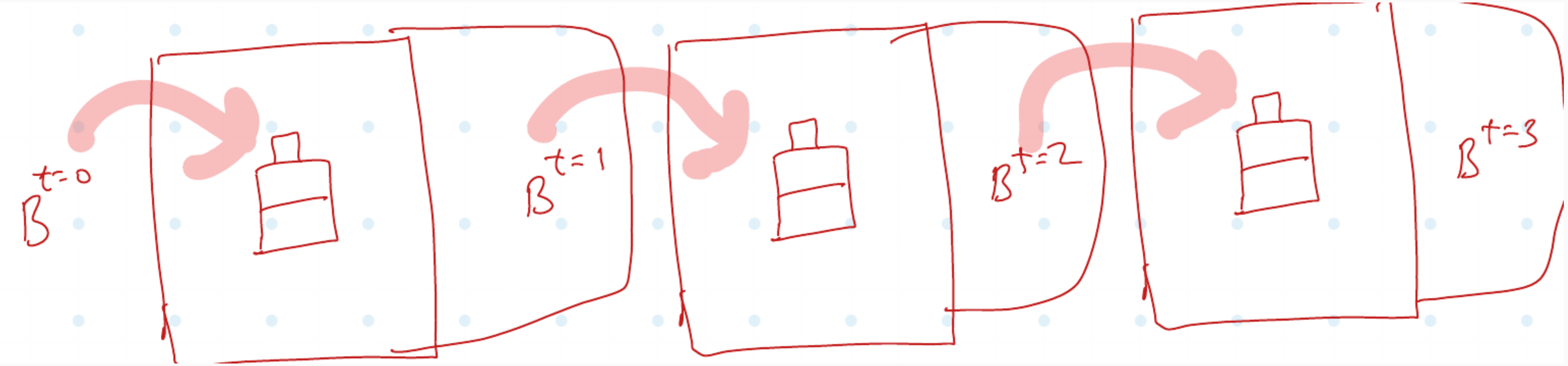
Currently under investigation

DDP – Differential Dynamic Programming

- Intuition and Derivation
- Results

DDP – Differential Dynamic Programming

Deriving DDP Formulation

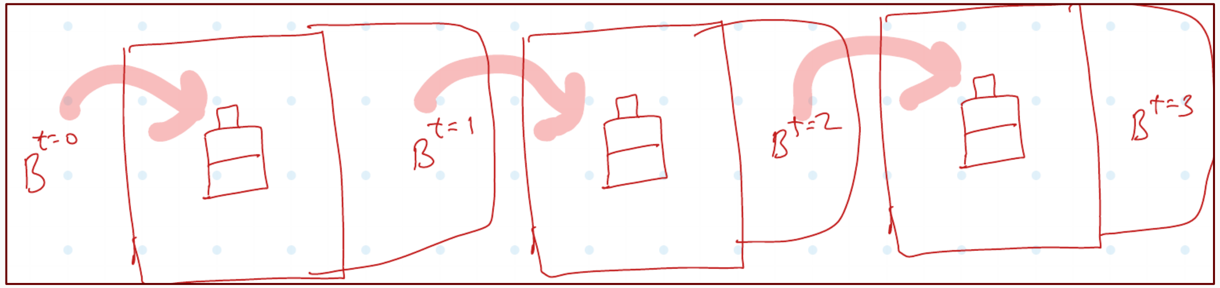


- Let's take a simple three-time step problem with a single battery and usual grid constraints
- Like in tADMM, we intend to **divide** the full MPOPF problem into individual **time-step based subproblems** $t = 1, t = 2, t = 3$

Source: DDP is used in Control Problems for QP typically, and was used in [aayushya] for MPOPF in transmission system

DDP – Differential Dynamic Programming

Deriving DDP Formulation



$$f(x^{t=1})$$
s.t.
$$g(x^{t=1}) \leq 0$$
$$B^{t=1} - \overline{B} \leq 0$$
$$\underline{B} - B^{t=1} \leq 0$$
$$h(x^{t=1}) = 0$$
$$B^{t=1} - B^{t=0} + \Delta t P_B^{t=1} = 0$$

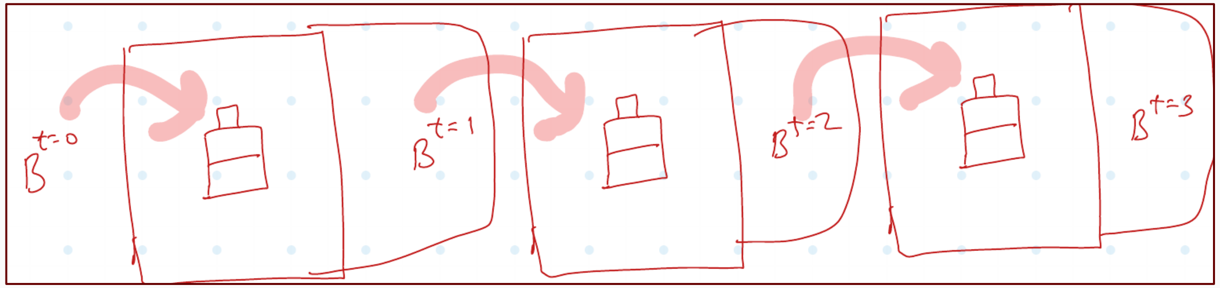
$$f(x^{t=2})$$
s.t.
$$g(x^{t=2}) \leq 0$$
$$B^{t=2} - \overline{B} \leq 0$$
$$\underline{B} - B^{t=2} \leq 0$$
$$h(x^{t=2}) = 0$$
$$B^{t=2} - B^{t=1} + \Delta t P_B^{t=2} = 0$$

$$f(x^{t=3})$$
s.t.
$$g(x^{t=3}) \leq 0$$
$$B^{t=3} - \overline{B} \leq 0$$
$$\underline{B} - B^{t=3} \leq 0$$
$$h(x^{t=3}) = 0$$
$$B^{t=3} - B^{t=2} + \Delta t P_B^{t=3} = 0$$

- We can write down every constraint (and objective function value contribution) for the MPOPF problem in a time-step by time-step basis

DDP – Differential Dynamic Programming

Deriving DDP Formulation



$$\sum_{t_0=1}^{3} f(x^{t=t_0})$$

s.t.

$$\left.\begin{aligned}
&g(x^{t=t_0}) \le 0\\
&h(x^{t=t_0}) = 0\\
&B^{t=t_0} - \overline{B} \le 0\\
&\underline{B} - B^{t=t_0} \le 0\\
&B^{t=t_0} - B^{t=t_0-1} + \Delta t \, P_B^{t=t_0} = 0
\end{aligned}\right\} \quad t_0 = 1:3$$

- We can write down every constraint (and objective function value contribution) for the MPOPF problem in a time-step by time-step basis

- The full MPOPF problem is nothing but the intersection of all of the constraints of the subproblem

DDP – Differential Dynamic Programming

Deriving DDP Formulation

- If we were to solve the MPOPF problem as a single problem (brute forced optimization), the optimizer internally will formulate a **Lagrangian L_{MPOPF}** which uses Lagrangian variables like λ **for inequalities** and μ **for equality** constraints

$$\sum_{t_0=1}^{3} f(x^{t=t_0})$$

s.t.

$$\left.\begin{aligned}
& g(x^{t=t_0}) \le 0 \\
& h(x^{t=t_0}) = 0 \\
& B^{t=t_0} - \overline{B} \le 0 \\
& \underline{B} - B^{t=t_0} \le 0 \\
& B^{t=t_0} - B^{t=t_0-1} + \Delta t\, P_B^{t=t_0} = 0
\end{aligned}\right\} \quad t_0 = 1:3$$

Lagrangian

$$L_{MPOPF} = \sum_{t_0=1}^{3} \left\{ f(x^{t=t_0}) + \lambda^{t=t_0} \cdot g(x^{t=t_0}) + \mu^{t=t_0} \cdot h(x^{t=t_0}) \right\}$$

$$+ \sum_{t_0=1}^{3} \left\{ \lambda_{\underline{B}}^{t=t_0} \left(\underline{B} - B^{t=t_0}\right) + \lambda_{\overline{B}}^{t=t_0} \left(B^{t=t_0} - \overline{B}\right) \right\}$$

$$+ \sum_{t_0=1}^{3} \mu_{soc}^{t=t_0} \left(B^{t=t_0} - B^{t=t_0-1} + \Delta t\, P_B^{t=t_0}\right)$$

DDP – Differential Dynamic Programming

Deriving DDP Formulation

- It is okay to separate out Lagrangian duals associated with State of Charge B^t and write them out separately in order to recognize that each B^t is associated with four dual variables $\lambda_{Bmin}, \lambda_{Bmax}, \mu_{SOC}^{\{t\}}, \mu_{SOC}^{\{t+1\}}$

$$L_{MPOPF} = \sum_{t_0=1}^{3} \Big\{ f(x^{t=t_0}) + \lambda^{t=t_0} \cdot g(x^{t=t_0}) + \mu^{t=t_0} \cdot h(x^{t=t_0}) \Big\}$$

Lagrangian

$$+ \sum_{t_0=1}^{3} \Big\{ \lambda_{\underline{B}}^{t=t_0} (\underline{B} - B^{t=t_0}) + \lambda_{\overline{B}}^{t=t_0} (B^{t=t_0} - \overline{B}) \Big\}$$

$$+ \sum_{t_0=1}^{3} \mu_{soc}^{t=t_0} (B^{t=t_0} - B^{t=t_0-1} + \Delta t \, P_B^{t=t_0})$$

DDP – Differential Dynamic Programming

Deriving DDP Formulation

- A solution $\boldsymbol{B^1}, \boldsymbol{B^2}, \boldsymbol{B^3}$ can only be optimal if it satisfied the **First Order Optimality Conditions** (necessary conditions)
- One of the First order optimality conditions is that the gradient of $\boldsymbol{L_{MPOPF}}$ should be zero wrt the variables, doing that for SOC variables $\boldsymbol{B^1}, \boldsymbol{B^2}, \boldsymbol{B^3}$ we get obtain:

$$L_{MPOPF} = \sum_{t_0=1}^{3} \Big\{ f(x^{t=t_0}) + \lambda^{t=t_0} \cdot g(x^{t=t_0}) + \mu^{t=t_0} \cdot h(x^{t=t_0}) \Big\}$$

Lagrangian

$$+ \sum_{t_0=1}^{3} \Big\{ \lambda_{\underline{B}}^{t=t_0} \left(\underline{B} - B^{t=t_0}\right) + \lambda_{\overline{B}}^{t=t_0} \left(B^{t=t_0} - \overline{B}\right) \Big\}$$

$$+ \sum_{t_0=1}^{3} \mu_{soc}^{t=t_0} \left(B^{t=t_0} - B^{t=t_0-1} + \Delta t \, P_B^{t=t_0}\right)$$

$$\nabla_{B^{t=1}} L_{MPOPF} = -\lambda_{\underline{B}}^{t=1} + \lambda_{\overline{B}}^{t=1} + \mu_{soc}^{t=1} - \mu_{soc}^{t=2} = 0$$

$$\nabla_{B^{t=2}} L_{MPOPF} = -\lambda_{\underline{B}}^{t=2} + \lambda_{\overline{B}}^{t=2} + \mu_{soc}^{t=2} - \mu_{soc}^{t=3} = 0$$

$$\nabla_{B^{t=3}} L_{MPOPF} = -\lambda_{\underline{B}}^{t=3} + \lambda_{\overline{B}}^{t=3} + \mu_{soc}^{t=3} = 0$$

Multi Period Optimality Conditions

DDP – Differential Dynamic Programming

Deriving DDP Formulation

- Comparing the (*) conditions with the conditions for a single time period problem, we see that **one** of the **key variables for multi-period optimality** is **missing**!

$$
\begin{aligned}
& f(x^{t=1}) \\
& \text{s.t.} \\
& g(x^{t=1}) \leq 0 \\
& B^{t=1} - \overline{B} \leq 0 \\
& \underline{B} - B^{t=1} \leq 0 \\
& h(x^{t=1}) = 0 \\
& B^{t=1} - B^{t=0} + \Delta t P_B^{t=1} = 0
\end{aligned}
$$

$$
\begin{aligned}
L^{t=1}_{ACOPF} = \; & f(x^{t=1}) \\
& + \lambda^{t=1} g(x^{t=1}) \\
& + \mu^{t=1} h(x^{t=1}) \\
& + \lambda^{t=1}_{\overline{B}} (B^{t=1} - \overline{B}) \\
& + \lambda^{t=1}_{\underline{B}} (\underline{B} - B^{t=1}) \\
& + \mu^{t=1}_{soc} (B^{t=1} - B^{t=0} + \Delta t P_B^{t=1})
\end{aligned}
$$

$$\nabla_{B^{t=1}} L_{ACOPF} = 0 = \underbrace{\lambda^{t=1}_{\overline{B}} - \lambda^{t=1}_{\underline{B}} + \mu^{t=1}_{soc} = 0}$$

Single Period Optimality Condition (1Δ)

$$-\lambda^{t=1}_{\underline{B}} + \lambda^{t=1}_{\overline{B}} + \mu^{t=1}_{soc} \boxed{- \mu^{t=2}_{soc}} = 0$$

Multi Period Optimality Condition (1*)

DDP – Differential Dynamic Programming

Deriving DDP Formulation

- Comparing the (*) conditions with the conditions for a single time period problem, we see that **one** of the **key variables for multi-period optimality** is **missing**!

- **DDP** works by ***nudging*** the **single time-step problem** towards **multi-period optimality** by **indirectly** making the single time step problem **ensure** that the **true condition (1*)** is solved for instead of the in-built (1Δ)

$$\nabla_{\beta^{t=1}} L_{ACOPF}^{t=1} = 0 = \underbrace{\lambda_{\bar{B}}^{t=1} - \lambda_{\underline{B}}^{t=1} + \mu_{soc}^{t=1} = 0}$$

Single Period Optimality Condition (1Δ)

$$-\lambda_{B}^{t=1} + \lambda_{\bar{B}}^{t=1} + \mu_{soc}^{t=1} \boxed{- \mu_{soc}^{t=2}} = 0$$

Multi Period Optimality Condition (1*)

DDP – Differential Dynamic Programming

Deriving DDP Formulation

DDP works by ***nudging*** the **single time-step problem** towards **multi-period optimality** by **indirectly** making the single time step problem **ensure** that the **true condition (1*)** is solved for instead of the in-built (1Δ)

$$\nabla_{B^{t=1}} L^{t=1}_{ACOPF} = 0 = \lambda^{t=1}_{\overline{B}} - \lambda^{t=1}_{\underline{B}} + \mu^{t=1}_{soc} = 0$$

Single Period Optimality Condition (1Δ)

$$L^{t=1}_{ACOPF} = f(x^{t=1}) + \lambda^{t=1} g(x^{t=1}) + \mu^{t=1} h(x^{t=1}) + \lambda^{t=1}_{\overline{B}}(B^{t=1} - \overline{B}) + \lambda^{t=1}_{\underline{B}}(\underline{B} - B^{t=1}) + \mu^{t=1}_{soc}(B^{t=1} - B^{t=0} + \Delta t\, P^{t=1}_{B}) + \boxed{\mu^{t=2}_{soc}(B^{t=2} - B^{t=1} + \Delta t\, P^{t=2}_{B})}$$

$$\min \quad f(x^{t=1}) - \mu^{t=2}_{soc} B^{t=1}$$

s.t.

$$g(x^{t=1}) \leq 0$$
$$h(x^{t=1}) = 0$$
$$B^{t=1} - B^{t=0} + \Delta t\, P^{t=1}_{B} = 0$$
$$B^{t=1} \in [\underline{B}, \overline{B}]$$
$$P^{t=1}_{B} \in [-P_{B_R}, P_{B_R}]$$

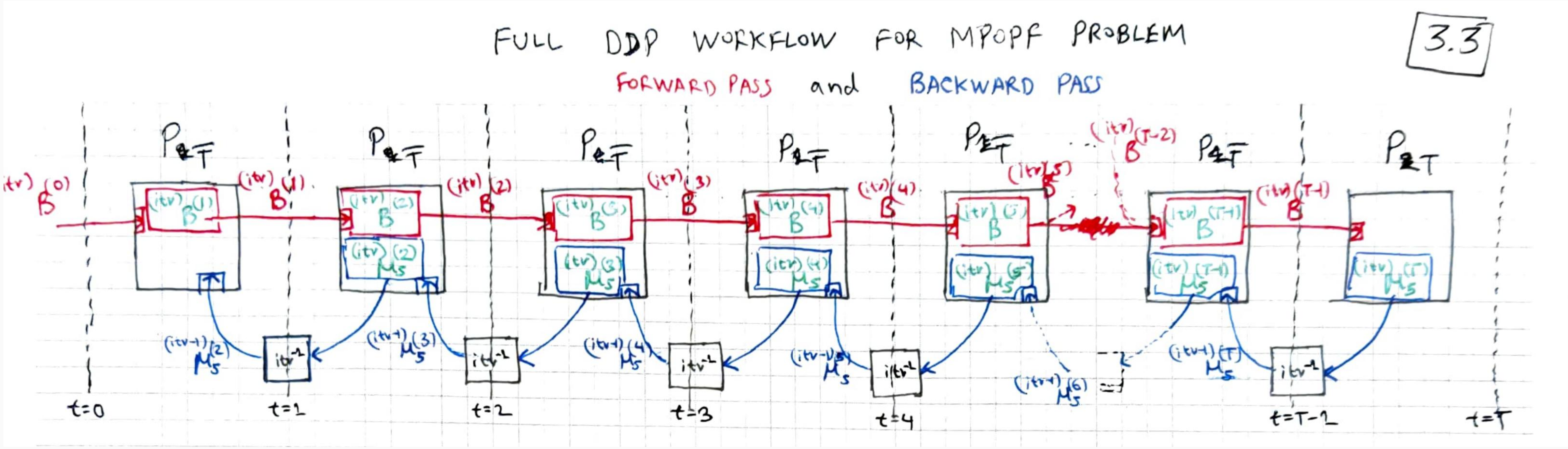
$$-\lambda^{t=1}_{\underline{B}} + \lambda^{t=1}_{\overline{B}} + \mu^{t=1}_{soc} \boxed{- \mu^{t=2}_{soc}} = 0$$

Multi Period Optimality Condition (1*)

DDP – Differential Dynamic Programming

DDP Workflow

In Forward Pass, we send the $\boldsymbol{B}^{\{t-1\}}$ solved for in the previous time-step to the current time-step t, and use the dual variable associated with the SOC trajectory $\mu_{SOC}^{\{t+1\}}$ from the next time-step. Both of these parameters (not decision variables) nudge the optimization at time-step t towards feasibility and optimality.



DDP – Differential Dynamic Programming

Forward Pass

$$
\begin{aligned}
\min \quad & C^t P^t_{Subs} && (16)\\
& + \alpha \sum_{j \in \mathscr{B}} \left\{ (1-\eta_c) P^t_{c_j} + \left(\frac{1}{\eta_d} - 1 \right) P^t_{d_j} \right\} \\
& + \sum_{j \in \mathscr{B}} \mu^{t+1}_{SOC_j} \left\{ -B^t_j \right\} \\
\text{s.t.} \\
& 0 = \sum_{(j,k) \in \mathscr{L}} \left\{ P^t_{jk} \right\} - \left\{ P^t_{ij} - r_{ij} l^t_{ij} \right\} - (P^t_{d_j} - P^t_{c_j}) - ({p_D}_j{}^t) + {p_L}_j{}^t && (17)\\
& 0 = \sum_{(j,k) \in \mathscr{L}} \left\{ Q^t_{jk} \right\} - \left\{ Q^t_{ij} - x_{ij} l^t_{ij} \right\} - (q^t_{D_j}) - (q^t_{B_j}) + {q_L}_j{}^t && (18)\\
& 0 = v^t_i - v^t_j - 2(r_{ij} P^t_{ij} + x_{ij} Q^t_{ij}) + \left\{ r^2_{ij} + x^2_{ij} \right\} l^t_{ij} && (19)\\
& 0 = (P^t_{ij})^2 + (Q^t_{ij})^2 - v^t_i l^t_{ij} && (20)\\
& 0 = B^t_j - \left\{ B^{t-1}_j + \Delta t \eta_c P^t_{c_j} - \Delta t \frac{1}{\eta_d} P^t_{d_j} \right\} && (21)\\
& 0 \geq {B_{min}}_j{}^t - {B_j}^t && (22)\\
& 0 \geq {B_j}^t - {B_{Max}}_j{}^t && (23)\\
& 0 \geq \text{All other inequality constraints} && (24)
\end{aligned}
$$

Apart from $\boldsymbol{B}^{\{t-1\}}$, $\mu_{SOC}^{\{t+1\}}$

Rest of the optimization problem of a single time-step t is just the usual set of network constraints

DDP – Differential Dynamic Programming

Backward Pass

$$\min \quad C^t P^t_{Subs} \tag{16}$$
$$+ \alpha \sum_{j \in \mathscr{B}} \left\{ (1-\eta_c) P^t_{c_j} + \left(\frac{1}{\eta_d} - 1 \right) P^t_{d_j} \right\}$$
$$+ \sum_{j \in \mathscr{B}} \mu^{t+1}_{SOC_j} \left\{ -B^t_j \right\}$$

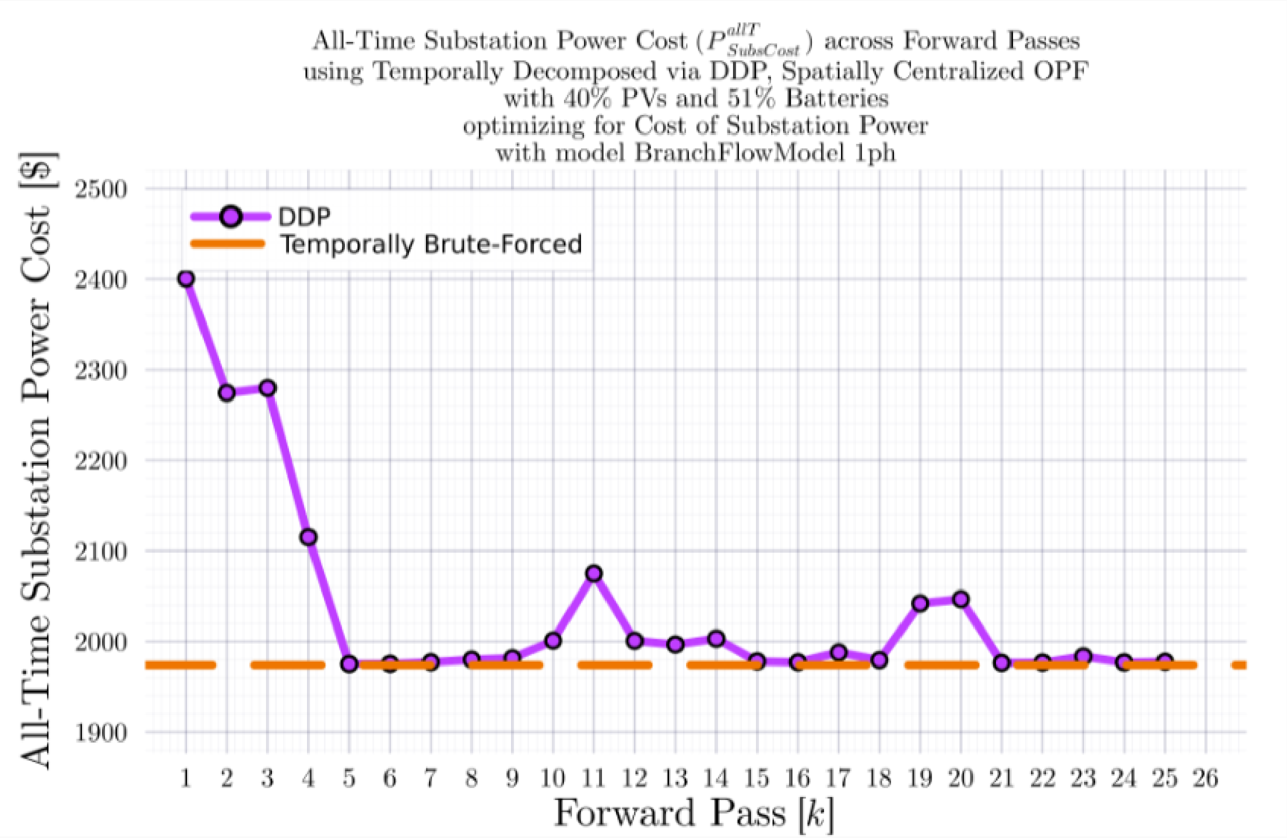
s.t.

$$0 = \sum_{(j,k) \in \mathscr{L}} \{P^t_{jk}\} - \{P^t_{ij} - r_{ij} l^t_{ij}\} - (P^t_{d_j} - P^t_{c_j}) - ({p_D}_j{}^t) + {p_L}_j{}^t \tag{17}$$
$$0 = \sum_{(j,k) \in \mathscr{L}} \{Q^t_{jk}\} - \{Q^t_{ij} - x_{ij} l^t_{ij}\} - (q^t_{D_j}) - (q^t_{B_j}) + {q_L}_j{}^t \tag{18}$$
$$0 = v^t_i - v^t_j - 2(r_{ij} P^t_{ij} + x_{ij} Q^t_{ij}) + \{r^2_{ij} + x^2_{ij}\} l^t_{ij} \tag{19}$$
$$0 = (P^t_{ij})^2 + (Q^t_{ij})^2 - v^t_i l^t_{ij} \tag{20}$$
$$0 = B^t_j - \left\{ B^{t-1}_j + \Delta t \eta_c P^t_{c_j} - \Delta t \frac{1}{\eta_d} P^t_{d_j} \right\} \tag{21}$$
$$0 \geq {B_{min}}_j{}^t - {B_j}^t \tag{22}$$
$$0 \geq {B_j}^t - {B_{Max}}_j{}^t \tag{23}$$
$$0 \geq \text{All other inequality constraints} \tag{24}$$

The Backward Pass basically just refers to updating $\mu_{SOC}^{\{t+1\}}$ values for the next Forward Pass

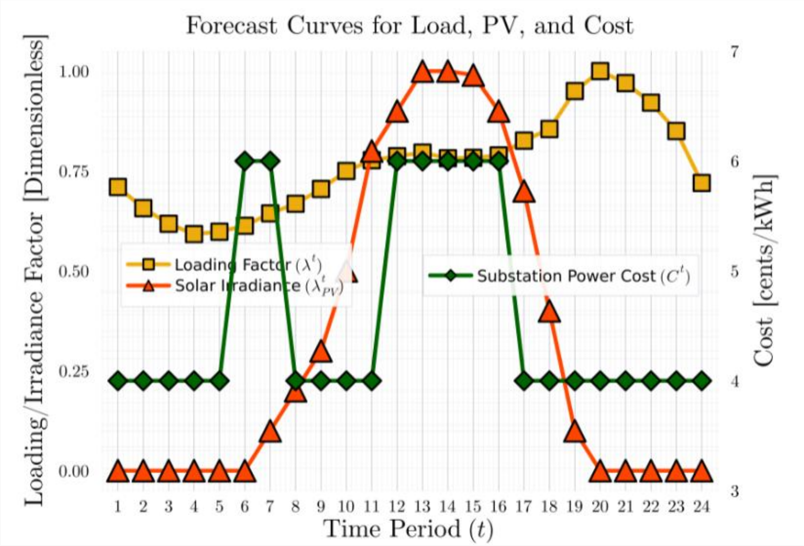
DDP – Differential Dynamic Programming

Key Results: IEEE123B_1ph BFM-NL with 40% PVs and 51% Batteries



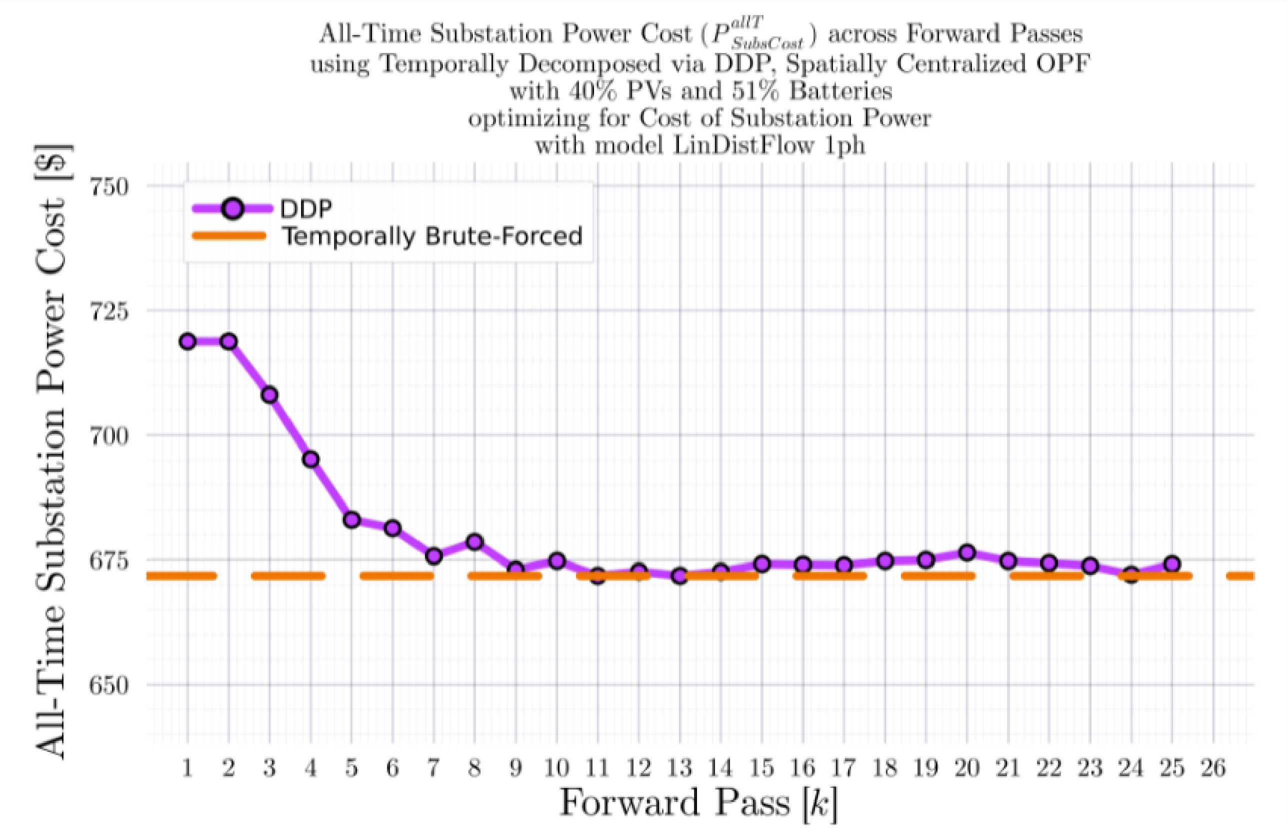
IEEE 123 1ph System – BFM-Nonlinear

- Currently while DDP iterations **'hit' near-optimal points** many times, they **never quite converge** to a fixed point.



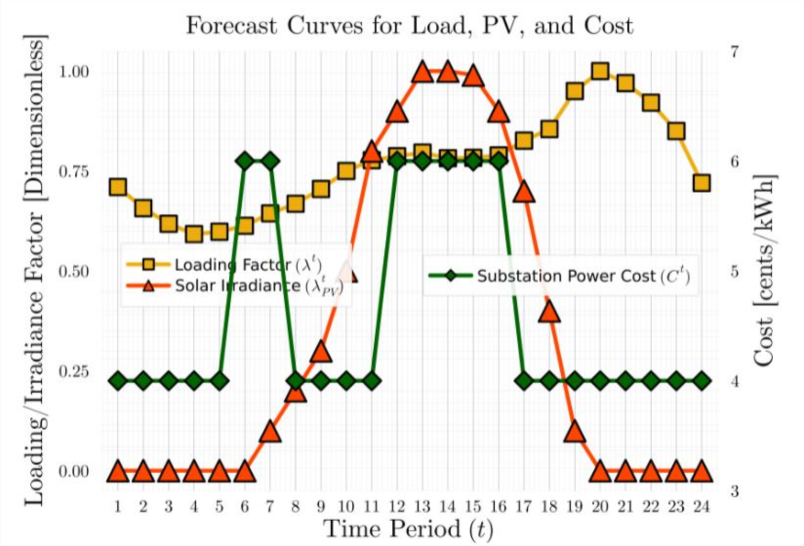
DDP – Differential Dynamic Programming

Key Results: IEEE123B_1ph LDF with 40% PVs and 51% Batteries



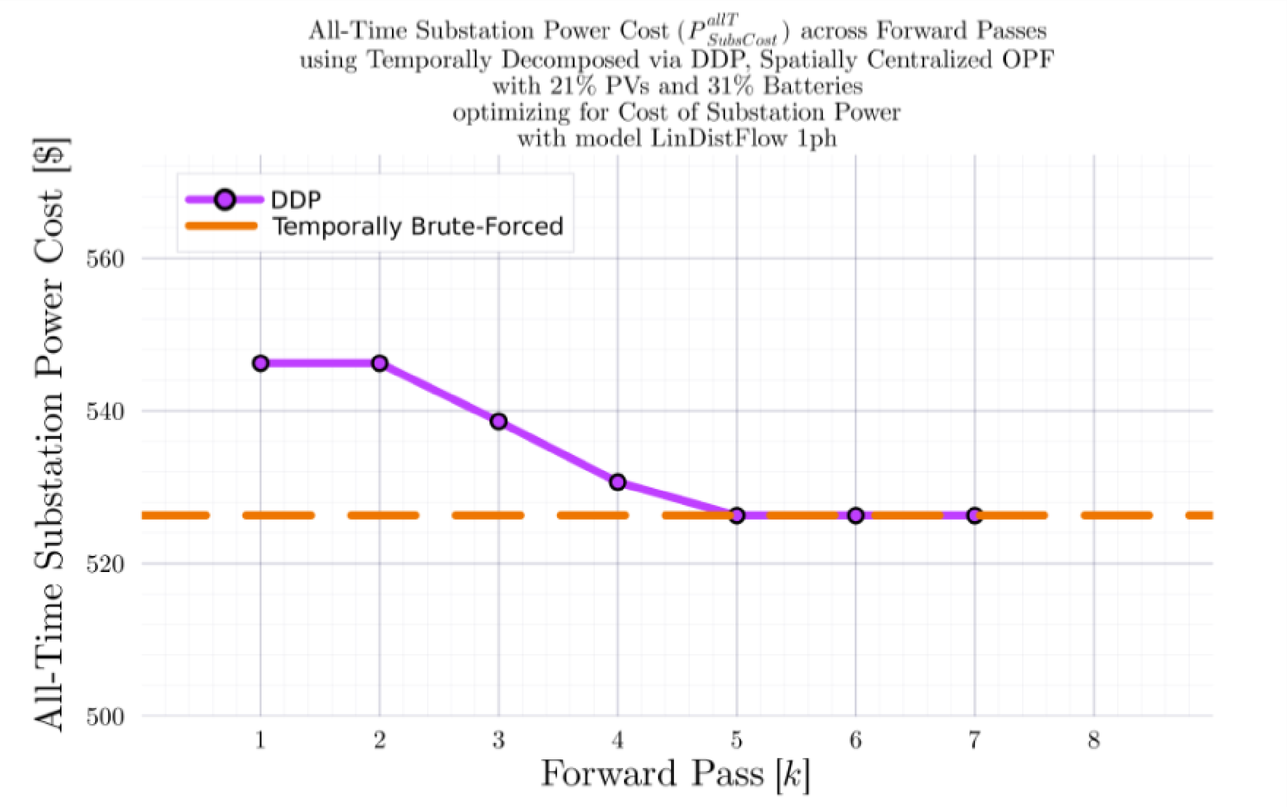
IEEE 123 1ph System - LinDistFlow

- Currently while DDP iterations **'hit' near-optimal points** many times, they **never quite converge** to a fixed point.
- Even for Linearized Problem



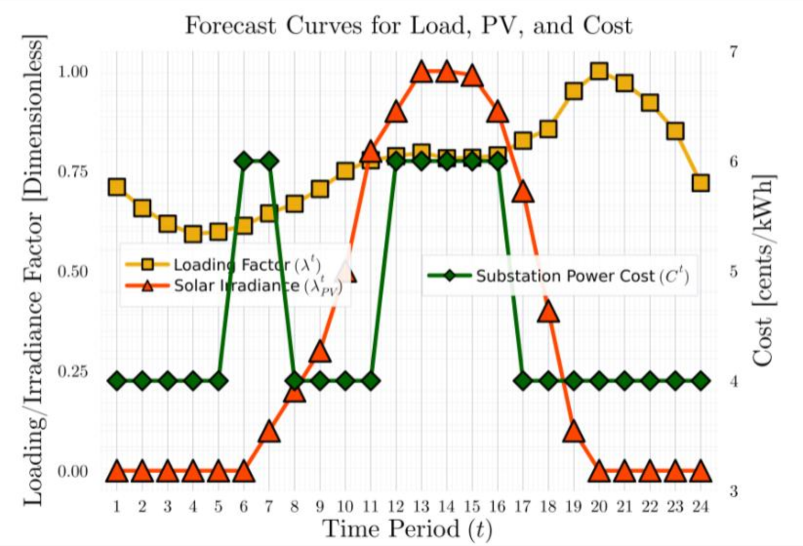
DDP – Differential Dynamic Programming

Key Results: IEEE730_1ph LDF with 21% PVs and 31% Batteries



IEEE 730 1ph System - LinDistFlow
[rahulBenchmarkOPF2022]

Certain **'catch convergence'** techniques were employed, but they too are not reliable



Publications, Future Work and Timeline

- Publications
- Perfecting Temporal Decomposition for Medium Sized Balanced Three-Phase Systems
- Temporal Decomposition for Large Sized Unbalanced Three-Phase Systems
- Concluding Research and Dissertation
- Timeline Gantt Chart

Publications, Future Work and Timeline

Current Publications and intended future publication

1. **Jha, A. R.**, Paul, S., & Dubey, A. . Spatially Distributed Multi-Period Optimal Power Flow with Battery Energy Storage Systems. 2024 56th North American Power Symposium (NAPS). IEEE. doi: 10.1109/NAPS61145.2024.10741846 [1]
2. **Jha, A. R.**, Paul, S., & Dubey, A. . Analyzing the Performance of Linear and Nonlinear Multi -Period Optimal Power Flow Models for Active Distribution Networks. 2025 IEEE North-East India International Energy Conversion Conference and Exhibition (NE-IECCE). IEEE. doi: 10.1109/NE-IECCE64154.2025.11183479 [8]

NAPS Paper

IAS Paper which can be extended to an IAS journal

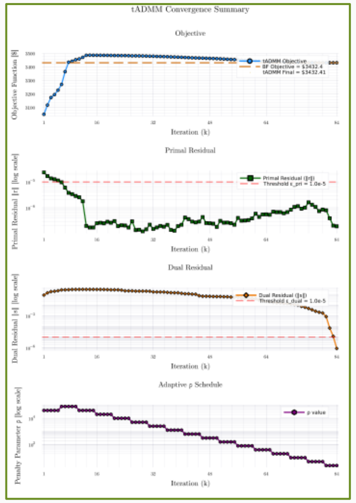
- Aiming to submit tADMM work as extension to the IAS paper by December 2025.

Future Work and Timeline

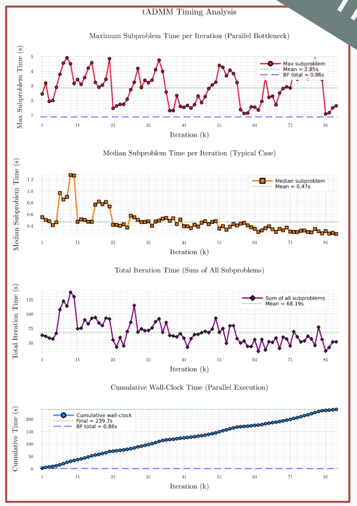
Task 1: Temporal Decomposition for Medium 1ph Systems

- Already successfully achieved: IEEE 123 Balanced-Three Phase Network + 17 PVs + 26 Batteries for $T = 96$ timesteps ($24h$ at $\Delta t = 15\text{min}$) solved to optimality using tADMM
- But current implementation is somehow slower in time compared to brute force solution, perhaps due to poorly optimized code or other reasons – currently in investigation

➢Now to December 2025



Good convergence



Slow optimization

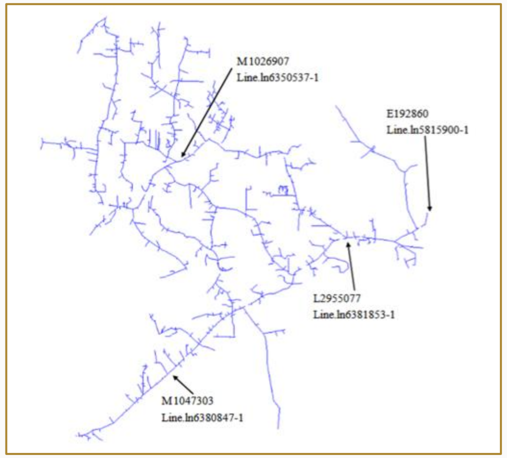
To fix!

Future Work and Timeline

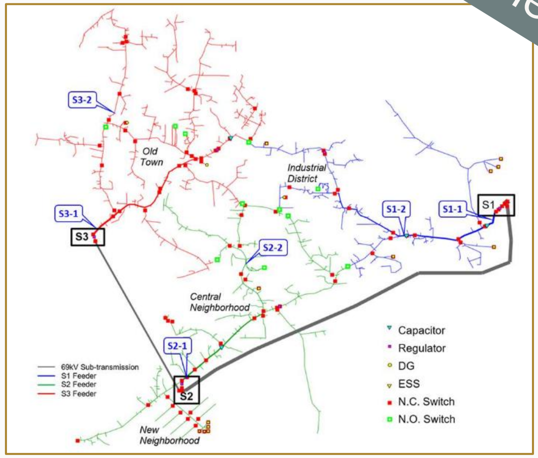
Task 2: Temporal Decomposition for Large Unbalanced 3ph Systems

- Formulate and implement tADMM for larger test systems like IEEE8500 3ph and 9500 3ph systems
  - First start with LinDistFlow3ph implementation, and if successful attempt to extend to nonlinear 3ph formulation

➤Jan 2026 to March 2026



IEEE 8500 3ph System [ieee8500]



9500 3ph System [9500Test]

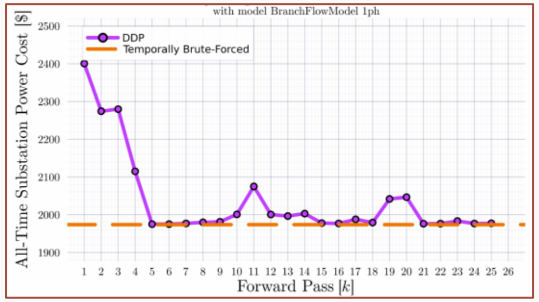
Implement

Future Work and Timeline

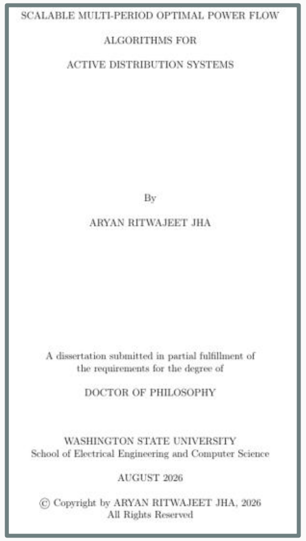
Task 3: Concluding Research and Dissertation

- Complete investigating DDP and document its performance in a paper
- Conduct a final comprehensive literature review on novel temporal decomposition methods for MPOPF
- Complete any remaining implementations or case studies

➢Feb 2026 to May 2026



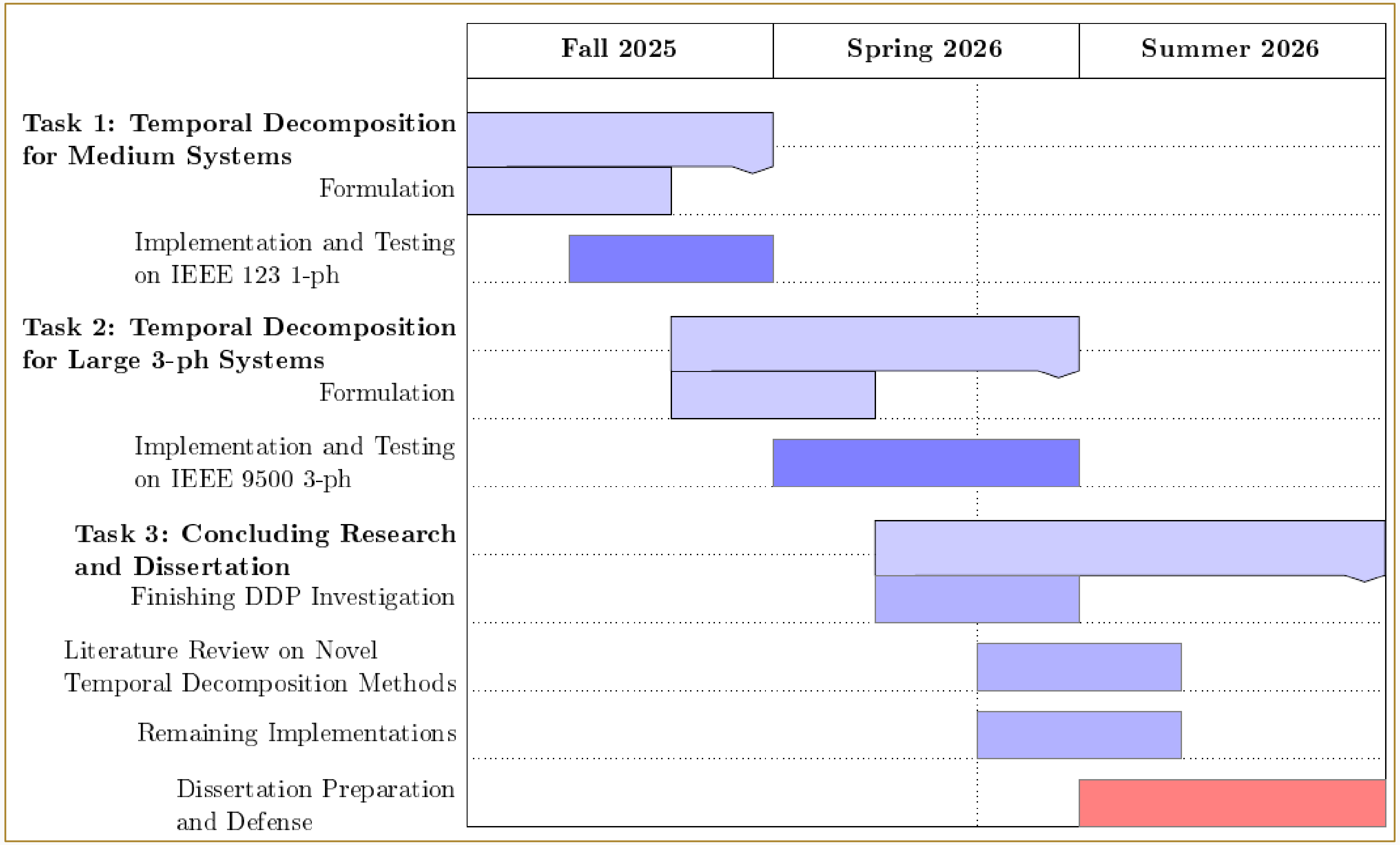
Try to finalize conclusion about DDP performance for MPOPF



Finish writing dissertation

Future Work and Timeline

Timeline as Gantt Chart



Gantt chart showing execution plan for future works

Courses

- Courses (Program of Study)

PhD Execution Pipeline at a Glance

Courses (Program of Study)

Course Number and Name	Semester	Instructor	Grade
E_E 507 Random Processes in Engineering	Fall 2022	Prof. Sandip Roy	A
E_E 521 Analysis of Power Systems	Fall 2022	Prof. Noel Schulz	A
E_E 523 Power Systems Stability	Spring 2023	Prof. Mani V. Venkatasubramanian	A
MATH 564 Convex and Nonlinear Optimization	Fall 2023	Prof. Tom Asaki	A
MATH 565 Nonsmooth Analysis and Optimization	Spring 2024	Prof. Tom Asaki	A-
CPT_S 530 Numerical Analysis[1]	Fall 2025	Prof. Alexander Panchenko	
E_E 582 Electrical Systems Modelling and Simulation[1]	Fall 2025	Prof. Seyedmilad Ebrahimi	
E_E 595 Directed Studies in Electrical Engineering[1]	Fall 2025	Prof. Rahul K. Gupta	

On track to be fulfilled by end of current semester (Fall 2025)

[1] currently taking this semester

References

References

1. [9500Test] Introducing the 9500 Node Distribution Test System to Support Advanced Power Applications. (2023, February 03). Retrieved from https://www.pnnl.gov/publications/introducing-9500-node-distribution-test-system-support-advanced-power-applications
2. [ieee8500] Arritt, R. F., & Dugan, R. C. . The IEEE 8500-node test feeder. IEEE PES T&D 2010. IEEE. doi: 10.1109/TDC.2010.5484381

- [1] Sadnan, R., & Dubey, A. (2021). Distributed Optimization Using Reduced Network Equivalents for Radial Power Distribution Systems. IEEE Trans. Power Syst., 36(4), 3645–3656. doi: 10.1109/TPWRS.2020.3049135

- [1A] Farivar, M., & Low, S. H. (2012). Branch Flow Model: Relaxations and Convexification (Parts I, II). arXiv, 1204.4865. Retrieved from https://arxiv.org/abs/1204.4865v4

- [2] Eilyan Bitar's Papers. (2023, August 16). Coordinated Aggregation
- of Distributed Demand-Side Resources. Retrieved from https://bitar.engineering.cornell.edu/papers.html

- [3] Pandey, A., Agarwal, A., & Pileggi, L. (2020). Incremental Model Building Homotopy Approach for Solving Exact AC-Constrained Optimal Power Flow. arXiv, 2011.00587. Retrieved from https://arxiv.org/abs/2011.00587v1

References

[4] Nazir, N., & Almassalkhi, M. (2021). Guaranteeing a Physically Realizable Battery Dispatch Without Charge-Discharge Complementarity Constraints. IEEE Trans. Smart Grid, 14(3), 2473–2476. doi: 10.1109/TSG.2021.3109805

[5] MultiPeriod-DistOPF-Benchmark. (2024, July 20). Retrieved from https://github.com/Realife-Brahmin/MultiPeriod-DistOPF-Benchmark