

Large-Scale Maintenance and Unit Commitment: A Decentralized Subgradient Approach

Paritosh Ramanan^{*†}, Murat Yildirim[‡], Nagi Gebraeel[†] and Edmond Chow^{*}

Abstract—Unit Commitment (UC) is a fundamental problem in power system operations. When coupled with generation maintenance, the joint optimization problem poses significant computational challenges due to coupling constraints linking maintenance and UC decisions. Obviously, these challenges grow with the size of the network. With the introduction of sensors for monitoring generator health and condition-based maintenance (CBM), these challenges have been magnified. ADMM-based decentralized methods have shown promise in solving large-scale UC problems, especially in vertically integrated power systems. However, in their current form, these methods fail to deliver similar computational performance and scalability when considering the joint UC and CBM problem.

This paper provides a novel decentralized optimization framework for solving large-scale, joint UC and CBM problems. Our approach relies on the novel use of the subgradient method to temporally decouple various subproblems of the ADMM-based formulation of the joint problem along the maintenance horizon. By effectively utilizing multithreading, our decentralized subgradient approach delivers superior computational performance and eliminates the need to move sensor data thereby alleviating privacy and security concerns. Using experiments on large scale test cases, we show that our framework can provide a speedup of upto 50x as compared to various state of the art benchmarks without compromising on solution quality.

Index Terms—Sensor driven prognosis, Joint operations and condition based maintenance, Decentralized and multithreaded optimization, Vertically integrated power systems.

NOMENCLATURE

Sets:

| | |
|---|--|
| $\mathcal{N}_r, G_r, \mathcal{U}_r, \mathcal{V}_r, \mathcal{I}_r$ | Neighboring regions, generators, foreign, boundary & internal buses of region r . |
| $G_r^b, \mathcal{U}_r^b, \mathcal{V}_r^b, \mathcal{I}_r^b$ | Generators, boundary, foreign & internal buses connected to bus $b \in \mathcal{U}_r \cup \mathcal{I}_r$. |
| \mathcal{R} | The set of all regions |
| \mathcal{B}_r | $\mathcal{U}_r \cup \mathcal{V}_r$, Boundary, foreign buses of r . |
| \mathcal{N}_r^b | Neighboring regions connected to bus $b \in \mathcal{U}_r$. |
| \mathcal{B}_r^b | $\mathcal{U}_r^b \cup \mathcal{V}_r^b \cup \mathcal{I}_r^b$, Neighboring buses of b . |
| M | Maintenance planning horizon |
| T | Operational planning horizon |

Decision Variables (at $t \in T$):

| | |
|----------------------------|---|
| y_t^g | The electricity dispatch of generator g |
| $x_t^g \in \{0, 1\}$ | The commitment decision variable of g |
| $z_t^g \in \{0, 1\}$ | The maintenance variable of generator g |
| ψ_t^b | Demand Curtailment at bus b |
| θ_t^b | The phase angle at bus b |
| $\tilde{\theta}_t^{b,r'}$ | The phase angle of bus b where $b \in \mathcal{U}_{r'}$ and $r' \in \mathcal{N}_r$. |
| $\pi_{U_t}^g, \pi_{D_t}^g$ | The up and down variable of generator g |
| $p_{r,t}$ | Production difference at region r |
| f_t^{uv} | Power flow from bus u to v such that $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r^u$ |
| α_g | The Lagrangian multiplier for maintenance cardinality constraint of generator g |
| λ_t^b | The Lagrangian multiplier with respect to phase angles of bus b where $b \in \mathcal{U}_r \cup \mathcal{V}_r$. |
| ϕ_t^{uv} | The Lagrangian multiplier with respect to flow from bus u to bus v where $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r$ for any region r |
| $\eta_{r,t}$ | The Lagrangian multiplier with respect to production target at region r |

Constants:

| | |
|--------------------------|---|
| d^g, c^g, S_U^g, S_D^g | The dispatch, commitment, start-up and shut-down cost of generator g |
| μ_U^g, μ_D^g, R^g | Minimum up time, down time and ramp-up ramp-down constant for g |
| δ_t^b | The demand at bus b at $t \in T$ |
| F_{max}^{uv} | Maximum capacity of line connecting buses u and v such that $u \in \mathcal{U}_r$ and $v \in \mathcal{V}_r^u$ |
| ρ_θ, ρ_f | Penalty parameter for phase angles, flows |
| Γ^{uv} | Phase angle conversion for line uv |

I. INTRODUCTION

Power network operations has traditionally been represented by the well-studied UC problem [1]. The UC problem optimizes generator commitment and production decisions subject to network topology, transmission, and generation constraints. Maintenance of generation assets like generators and turbines has received a fair share of attention in the literature [2], [3]. However, most of the research has focused on periodic and

^{*}School of Computational Science and Engineering, Georgia Institute of technology, Atlanta, GA, USA 30332

[†]College of Engineering, Wayne State University, Detroit, MI, USA 48202

[‡]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of technology, Atlanta, GA, USA 30332.

paritoshpr@gatech.edu, murat@wayne.edu, nagi.gebraeel@isye.gatech.edu, echow@cc.gatech.edu

This research is based upon work supported by the U.S. Department of Energy, Office of Fossil Energy (National Energy Technology Laboratory), under Award Number DOE-FE0031288.

calendar-based maintenance schedules where the goal is to optimize maintenance intervals to minimize maintenance cost [3]. Few research efforts have tried to integrate maintenance with network operations. Some noteworthy examples include [4], [5], [6]. These problems typically focus on a vertically integrated power system setting involving periodic maintenance schemes. Today, the rise of digital frameworks like the Internet-of-Things (IoT) is transforming the power generation industry. It has also highlighted the importance of leveraging IoT technologies like condition monitoring to advance maintenance management, i.e., condition-based maintenance (CBM). As a result, recent works have studied the integration of condition based maintenance (CBM) with operations and demonstrated significant cost savings by incorporating sensor-based predictive analytics [7], [8]. This paper provides a novel decentralized optimization framework to address the computational challenges in solving the joint UC and CBM optimization problem for large scale power networks.

The joint UC and CBM problem is significantly more complex than the standard UC problem because it involves maintenance and commitment coupling constraints. Moreover, the planning horizon for the joint problem tends to be in the order of months due to the timeline of maintenance decisions, whereas for UC it is typically 24 hours. Given the long planning horizons, the number of binary variables related to maintenance decisions are orders of magnitude greater than the conventional UC problem setting.

From a computational standpoint, decentralized solution methodologies, such as alternating direction method of multipliers (ADMM), have been widely used to improve computational performance of large scale mixed integer planning problems like UC [9], [10], [11], [12]. These solution methodologies rely on decomposing the power network topology into multiple autonomous regions. The methods iteratively apply ADMM by dualizing network flow constraints corresponding to transmission lines between regions. However, conventional ADMM-based decentralization approaches are not sufficient to solve large scale instances of the joint UC and CBM problem. This is because the local regional subproblems themselves become orders of magnitude more complex than the corresponding regional subproblems found in decentralized UC. Therefore, a direct adaptation of ADMM-based decentralization is not sufficient to address the computational complexities of the joint problem.

In addition to computational benefits, an added capability of decentralized methods also rests in its ability to significantly reduce or eliminate the need to move data [9], [11], [12]. The digitization of the grid due to an influx of IoT enabled sensors, has led to important questions on the security, privacy and handling of sensor data. In fact, the U.S. Department of Homeland Security's Industrial Control Systems (ICS) Cyber Emergency Response Team indicates that the energy sector accounts for nearly 35% of all ICS-related incidents in the United States[13]. Moreover, within power systems, estimating operating conditions of generation equipment followed by communication between multiple subsystems have been deemed essential system functions likely at risk to cyber attacks [14]. Needless to say, the reliance of CBM

methodologies on IoT driven sensor data makes it a perfect candidate for cyber attacks and potential misuse of sensor datasets by malicious parties. The need to frequently transmit highly resolved sensor data from a vast number of assets to a centralized database amplifies the risks of potential data leaks by providing a greater attack surface for intruders [14]. As a result, data manipulation attacks could take place which could lead to inaccurate estimates of asset health [15], [16] directly affecting maintenance policies and the overall stability of the grid. By preserving sensor data privacy and eliminating the need for its transmission, decentralized methods significantly reduce the risks of data leaks leading to enhanced security measures.

While decentralization has several benefits, the unique aspects of our joint maintenance and UC problem present numerous computational complications. The basis of our approach rests on the efficient decomposition of large scale instance of the joint UC and CBM optimization problem along the temporal and regional domains. Our approach relies on the novel use of the subgradient method to temporally decouple various regional subproblems of the ADMM-based problem formulation along the maintenance time horizon. The regional decomposition is solved in a decentralized fashion while the temporal decomposition is leveraged for multithreading.

Instead of directly solving each subproblem locally as proposed in [9],[11],[17] the introduction of the subgradient method enables us to deftly leverage multithreading to transform the complex joint problem formulation into a highly scalable, computationally efficient one. Next, we iteratively employ ADMM [18] to balance power flow between neighboring regions by dualizing the flow constraints corresponding to respective tie lines. Our specific contributions in this paper are summarized as follows:

- We develop a computationally efficient, decentralized, multithreaded subgradient method for solving large instances of the joint UC and CBM optimization problem. Existing ADMM-based decentralization models [17], [11], alone are not sufficient to solve such large scale instances. Therefore, our method relies on exploiting the block diagonal structure obtained by dualizing the maintenance cardinality constraints. This paves the way for applying the subgradient method to the local subproblems which can effectively leverage multithreading. Our method therefore, transforms the joint formulation into a massively parallelizable solution framework.
- Using our decentralized approach, we eliminate the need to move any critical sensor data required for maintenance scheduling at every region. This respects data residency requirements and significantly reduces cybersecurity threats that can arise from leaked IoT data.
- For evaluation purposes, we develop a High Performance Computing (HPC) implementation of our decentralized framework based on a hybrid Message Passing Interface (MPI) and OpenMP frameworks. We assign every region to an MPI process and multiple OpenMP threads to every region to achieve a scalable and efficient implementation.

The complexity of the joint problem is evidenced by the

fact that the solution to the IEEE 3012 bus case over a planning horizon of 3 months required approximately 48 hours by state-of-the-art commercial solvers. On the other hand, our multithreaded, decentralized solution paradigm delivered a speedup of up to 50 in some cases by effectively leveraging the computational power of 1200 threads orchestrated on a state-of-the-art supercomputer. The nature and scale of compute employed in this paper is in stark contrast to existing large-scale, decentralized UC implementations [11], [12], [17] which only required using up to 120 threads for the same problem instances.

In Section II, we review relevant literature pertaining to decentralized optimization of power systems. Details of the decentralized CBM problem formulation are discussed in Section III. In Section IV, we develop the decentralized joint CBM and operations algorithm that preserves regional data privacy. In Section V, we present our experimental results followed by conclusion and future work in Section VI.

II. RELATED WORK

Large scale problems in power systems are characterized by long planning horizons and large network sizes. While distributed and decentralized algorithms have both been applied towards solving large scale power system problems, there is a subtle but profound difference between the two categories. Distributed algorithms like those based on cutting planes are typically characterized by the presence of a master process that coordinates the entire computational progress [10]. On the other hand, decentralized algorithms, while being parallel and distributed in nature, function even in the absence of a master process [11]. As a result, decentralized approaches yield two important benefits that are not possible otherwise in generic distributed approaches relying on a master process. Due to the presence only of peer-to-peer message exchanges, decentralized approaches exhibit greater scalability [12]. Further, unlike their distributed counterparts, decentralized methods are based on a topology decomposition of the power network into multiple regions that are representative of utilities or their subsidiaries thereof [9]. Each region retains full ownership of local data and only exchanges flow information pertaining to shared transmission lines with neighbors [9], [11].

Another important differentiator between generic distributed algorithms and decentralized approaches is concerning the implementation aspect. As a consequence of their reliance only on peer-to-peer message exchanges, a decentralized algorithm can be implemented on a geographically distributed computing architecture [10], [11]. On the other hand, decompositions in distributed algorithms are designed mainly for computational convenience with the intent of unleashing massive parallelization potential [19], [20]. Typically, subproblems in distributed algorithms emanate from temporal decompositions for deterministic optimization cases [8] and scenario driven decompositions for stochastic cases [19], [20] without representing any real world entity. Therefore, such approaches can only function on a specialized computational system like an HPC cluster that offers significant parallelization potential but cannot handle geographically distributed problem instances [10]. Owing to such tight limitation of the implementation aspect, distributed

approaches force end users to transfer relevant information to an HPC cluster leading to data privacy concerns [12].

Distributed solutions pertaining to operations have conventionally revolved around stochastic UC [19], [20] with scenario based decompositions. The work done in [19] solves stochastic unit commitment problem for a fleet of sustainable energy sources where the imposed demand is time varying. The authors in [20] solve a unit commitment problem using an incremental subgradient method to progress the dual variable while simultaneously recovering primal feasibility. Lately, decentralized approaches for solving deterministic UC are gaining popularity for their enhanced scalability and privacy preserving aspects as stated above [9], [10], [11], [12].

On the other hand, as mentioned in Section I, efforts that have focused on integrating maintenance and network operations are limited and based on the cutting plane method. Pertinent examples include [4], [5], [6] where models for optimizing networks operations integrated periodic maintenance of assets and transmission lines. In [21], the authors integrated reliability information, namely generator failure rates, in a simplified operations problem. The authors in [7], [8] have proposed a sensor data driven, joint optimization of CBM with operations which represents the current state of the art. However, this approach does not scale well with larger instances on account of limited potential for parallelization. Therefore, a decentralized technique to solve the joint problem would be highly beneficial in order to achieve scalability as well as reduce computational time significantly. However, decentralized solution paradigms specifically tailored for the joint maintenance problem have largely remained unexplored.

III. PROBLEM FORMULATION

We propose a decentralized formulation based on regional decomposition with multiple regional subproblems. From a practical standpoint, each region may denote a subsidiary of the utility company in a vertically integrated market. Consider the sample network in Figure 1 consisting of 3 regions with boundary and foreign bus categorization defined as follows:

- Region 1: $\mathcal{U}_1 = \{B, C\}$, $\mathcal{V}_1 = \{G, E\}$
- Region 2: $\mathcal{U}_2 = \{E\}$, $\mathcal{V}_2 = \{F, C\}$
- Region 3: $\mathcal{U}_3 = \{G, F\}$, $\mathcal{V}_3 = \{B, E\}$

Every region is comprised of local (critical) generators and buses subject to its own operational constraints. Every critical generator is instrumented with sensors for monitoring health/degradation. Sensor data in each region is streamed to respective local databases where predictive analytic algorithms are used to predict the remaining operational life of the generator. As mentioned earlier, the focus of this paper is not to develop accurate predictive degradation models but rather to address the computational challenges in large scale joint optimization of operations and maintenance in the presence of IoT-enabled generation assets.

A. Degradation-based Predictive Modeling

Predictive analytics is a crucial aspect of our approach. Degradation-based sensor data from IoT-enabled generators are incorporated into the regional operations and maintenance scheduling problems. We leverage contemporary degradation

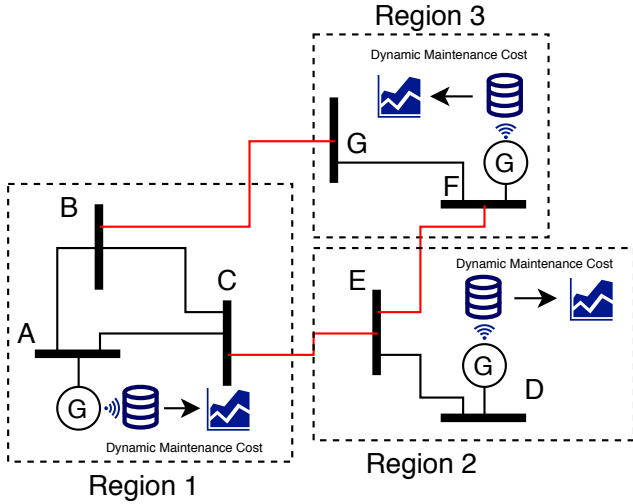


Fig. 1: Partition of Network topology into regions with full data privacy.

models developed by [22] to compute predictions of generator remaining lifetimes. The authors in [22] proposed a Bayesian framework that utilizes real-time degradation signals from partially degraded assets to predict and continuously update residual life distributions (RLDs). This framework was later extended in [23] by computing optimal replacement/maintenance and spare parts ordering policies driven by real-time RLD predictions. The authors used the RLDs to calculate a convex maintenance cost function to identify the optimal time to perform maintenance. The cost function represents the trade-off between the cost associated with the risk of unexpected failure and the opportunity cost associated with performing premature (or unnecessary) preventive maintenance. As new sensor data is observed, an asset's RLD and the corresponding cost function are updated through a Bayesian framework. [7], [8] adopted this approach in their joint optimization problem and successfully integrated these dynamically evolving expected maintenance cost functions. Considering a maintenance planning horizon M , the expected maintenance cost function at maintenance interval $m \in M$ is expressed in Equation 1.

$$\omega_m^g = \kappa \cdot \frac{\omega_p^g P(\tau_g > m) + \omega_f^g P(\tau_g \leq m)}{\int_0^m P(\tau_g > z) dz + m_o} \quad (1)$$

In Equation 1, ω_m^g denotes the maintenance cost at epoch m for generator g of age m_o . κ represents the maintenance criticality coefficient, denoting the relative importance of maintenance with respect to operations. ω_p^g and ω_f^g are the costs for preventive maintenance, and unexpected failure for generator g , respectively while τ_g is the remaining life of generator g . In other words, the function ω_m^g translates τ_g (the RLD of generator g) into a degradation-based maintenance cost function over time.

B. Decentralized Joint Maintenance and Operations

We note that the maintenance planning horizon set M comprises of multiple smaller operational planning horizons

T_m such that,

$$T = \bigcup_{m=1}^{|M|} T_m, \text{ where, } T_m = \left\{ t | t \in \left[\frac{m|T|}{|M|} \dots \frac{(m+1)|T|}{|M|} \right] \right\}$$

Therefore, the regional subproblem seeks to minimize the objective cost as represented by Problem (2).

$$\begin{aligned} \mathcal{L}_r(\Delta) = & \sum_{t \in T} \sum_{g \in G_r} \left[c_t^g x_t^g + d_t^g y_t^g + S_U^g \pi_{U_t}^g + S_D^g \pi_{D_t}^g \right] \\ & + \sum_{t \in T} \left[E_t(\bar{\theta}, \phi) + H_t(\bar{F}, \lambda) + I_t(\bar{p}, \eta) \right] \quad (2) \\ & + \nu \sum_{t \in T} \sum_{u \in \mathcal{U}_r \cup \mathcal{I}_r} \psi_t^u + \sum_{g \in G_r} \sum_{m \in M} \omega_m^g z_m^g \end{aligned}$$

where $\Delta = \{\bar{\theta}, \bar{F}, \bar{p}, \lambda, \phi, \eta\}$ represents the consensus phase angle, flow, regional production target and their respective lagrangian multipliers. Further,

$$E_t(\bar{\theta}, \lambda) = \sum_{b \in \mathcal{B}_r} \left[\lambda_t^b |\theta_t^b - \bar{\theta}_t^b| + \frac{\rho_\theta}{2} (\theta_t^b - \bar{\theta}_t^b)^2 \right]$$

$$H_t(\bar{F}, \phi) = \sum_{u \in \mathcal{U}_r} \sum_{v \in \mathcal{V}_r^u} \left[\phi_t^{uv} |F_t^{uv} - \bar{F}_{t,k}^{uv}| + \frac{\rho_f}{2} (F_t^{uv} - \bar{F}_{t,k}^{uv})^2 \right]$$

$$I_t(\bar{p}, \eta) = \eta_{r,t} (p_{r,t} - \bar{p}_{r,t}) + \frac{\rho_p}{2} (p_{r,t} - \bar{p}_{r,t})$$

The objective function represented by Problem (2), consists of a commitment, operations and maintenance cost components. In addition, it also consists of ADMM penalty terms imposed on balancing flow estimates among neighboring regions. Flow estimates are iteratively balanced across transmission lines common with neighboring regions as well as with respect to the phase angles at their respective boundary buses. We attempt to reduce overall demand violation by determining a local, customized production target $\bar{p}_{r,t}$ based on global demand violation. Computing a customized production target for every region has been shown to yield smooth convergence in large scale decentralized methods especially from the operational standpoint [11].

In our decentralized formulation represented by Problem (2), the operational decision including commitment and production are decoupled at linking intervals for every maintenance epoch [5], [6], [7], [8], [24]. The decoupling of minimum up/down, ramping constraints at linking intervals is a common practice meant to decompose the computationally challenging joint problem into multiple short-term UC subproblems as illustrated in [5], [6], [24]. Therefore, for a particular maintenance epoch $m \in M$, let \mathcal{Q}_m^r denote the set of inequalities (3a)-(3h), $\forall t \in T_m$.

$$x_t^g \leq 1 - z_m^g, \quad \forall g \in G_r \quad (3a)$$

$$P_{min}^g x_t^g \leq y_t^g \leq P_{max}^g x_t^g, \quad \forall g \in G_r \quad (3b)$$

$$-\pi_{Dt}^g \leq x_t^g - x_{t-1}^g \leq \pi_{Ut}^g, \quad \forall g \in G_r \quad (3c)$$

$$-R^g \leq y_t^g - y_{t-1}^g \leq R^g, \quad \forall g \in G_r \quad (3d)$$

$$\Gamma^{uv} (\theta_t^u - \theta_t^v) = f_t^{uv}, \quad \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r^u \quad (3e)$$

$$-F_{max}^{uv} \leq \Gamma^{uv} (\theta_t^u - \theta_t^v) \leq F_{max}^{uv}, \quad \forall u \in \mathcal{U}_r \cup \mathcal{I}_r, \forall v \in \mathcal{B}_r^u \quad (3f)$$

$$\sum_{\forall g \in G_r^u} y_t^g - \delta_t^u + \psi_t^u = \sum_{\forall v \in \mathcal{B}_r^u} [\Gamma^{uv}(\theta_t^u - \theta_t^v)], \forall u \in \mathcal{U}_r \cup \mathcal{I}_r \quad (3g)$$

$$\sum_{\forall i \in U_t} \pi_{U_i}^g \leq x_t^g \leq 1 - \sum_{\forall i \in D_t} \pi_{D_i}^g, \forall g \in G_r, \quad (3h)$$

$$U_t = [t - \mu_U^g + 1, t], D_t = [t - \mu_D^g + 1, t]$$

Constraint (3a) ensures that a generator that has been placed under maintenance must not have any production. Constraint (3b) enforces production limits at each generator while Constraints (3c) and (3h) enforce minimum up and down-time for each generator. Constraint (3d) enforces their respective ramping limitations. Equation (3e) establishes the linear relationship between flow and their respective phase angles. Constraint (3f) enforces transmission line capacity constraints. Equation (3g) balances the demand at each bus with local generation and network flow. Equations (3e)-(3g) enforce network flow constraints globally. Therefore, the joint CBM and operations problem can be represented by Problem (4).

$$\min \mathcal{L}_r(\Delta) \quad (4a)$$

$$\text{s.t.} \quad \sum_{m \in M} z_m^g = 1, \quad \forall g \in G_r \quad (4b)$$

$$\mathbf{x}, \mathbf{y}, \mathbf{z}, \boldsymbol{\pi}_U, \boldsymbol{\pi}_D, \boldsymbol{\theta}, \mathbf{f} \in \mathcal{Q}_m^r, \quad \forall m \in M \quad (4c)$$

In Problem (4) Constraint (4b) represents the maintenance cardinality constraint. Specifically, Constraint (4b) ensures that maintenance must be performed on each generator exactly once during the maintenance planning horizon. We also note that our formulation represented by Problem (4) does not make any assumptions on the length of the maintenance epochs. Therefore, the same decentralized, subgradient based approach proposed in this section is capable of handling different values of lengths of maintenance epochs (like 2 weeks, 3 weeks etc.).

We estimate two important consensus quantities, i.e. *intermediate flow* denoted by \bar{F}_t^{uv} and *intermediate phase angles* denoted by $\bar{\theta}_t^b$ based on Equations (5),(6) respectively. These intermediate values are calculated based on the phase angle estimates received from neighboring regions $\forall b \in \mathcal{B}_r$ and $\forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r$ for phase angles and flows respectively applied $\forall t \in T$.

$$\bar{\theta}_t^b = \frac{\theta_t^b + \sum_{r' \in \mathcal{N}_r^b} \bar{\theta}_t^{b,r'}}{|\mathcal{N}_r^b| + 1}, \quad b \in \mathcal{B}^r \quad (5)$$

$$\bar{F}_t^{uv} = \frac{\Gamma^{uv}(\bar{\theta}_t^{u,r'} - \bar{\theta}_t^{v,r'}) + \Gamma^{uv}(\theta_t^u - \theta_t^v)}{2}, \quad u \in \mathcal{U}_r, v \in \mathcal{V}_r, r' \in \mathcal{N}_r^u \quad (6)$$

We update the Lagrangian multipliers λ, ϕ based on Equations (7),(8) respectively.

$$\lambda_t^b = \lambda_t^b + \rho_\theta(\theta_t^b - \bar{\theta}_t^b), \quad \forall b \in \mathcal{B}_r, \forall t \in T \quad (7)$$

$$\phi_t^{uv} = \phi_t^{uv} + \rho_f(F_t^{uv} - \bar{F}_t^{uv}), \quad \forall u \in \mathcal{U}_r, \forall v \in \mathcal{V}_r, \forall t \in T \quad (8)$$

We estimate the customized production target for region r

based on Equation (9).

$$\bar{p}_{r,t} = \sum_{\forall g \in G_r} y_t^g + \frac{\sum_{\forall r \in \mathcal{R}} \varphi_{r,t}}{|\mathcal{R}|} \quad (9)$$

In Equation (9), $\varphi_{r,t} = \left| \sum_{\forall u \in \mathcal{U}_r \cup \mathcal{I}_r} (\delta_t^u - \psi_t^u) - \sum_{\forall g \in G_r} y_t^g \right|$ represents the local violation $\forall t \in T$. We update the associated Lagrangian η based on Equation (10).

$$\eta_{r,t} = p_{r,t} + \rho_p(p_{r,t} - \bar{p}_{r,t}) \quad (10)$$

Therefore, the joint optimization model given by Problem (4) describes a Mixed-Integer Quadratic Problem (MIQP) which solves for the maintenance and operations in a decentralized manner. The Lagrangian terms in the model serve as *penalties* for deviating from a position of balance. Convergence occurs when the dualized flow terms become small enough so that the optimization problem given by (4) become mathematically equivalent to that of a *centralized* problem as described in [8]. We are now in a position to state Lemma 1.

Lemma 1. Let $\mathcal{L}_r(\Delta)$ represent the objective of Problem (4) subject to constraints (4b),(4c), then

$$\sum_{m \in M} \min_{\mathcal{Q}_m^r} \mathcal{L}_r^m(\Delta, \alpha) \leq \min \mathcal{L}_r(\Delta)$$

where, α_g represents the dual variable with respect to Constraint (4b)

Proof. Dualizing Equation (4b) with the dual variable α_g , we obtain the Lagrangian relaxation $\hat{\mathcal{L}}_r$

$$\hat{\mathcal{L}}_r = \mathcal{L}_r(\Delta) + \sum_{g \in G_r} \alpha_g (1 - \sum_{m \in M} z_m^g) = \sum_{m \in M} \mathcal{L}_r^m(\Delta, \alpha) \quad (11)$$

where,

$$\begin{aligned} \mathcal{L}_r^m(\Delta, \alpha) = & \sum_{t \in T_m} \left[E_t(\bar{\boldsymbol{\theta}}, \boldsymbol{\phi}) + H_t(\bar{\mathbf{F}}, \boldsymbol{\lambda}) + I_t(\bar{\mathbf{p}}, \boldsymbol{\eta}) + \nu \sum_{u \in \mathcal{U}_r \cup \mathcal{I}_r} \psi_t^u \right] \\ & + \sum_{g \in G_r} \left[\sum_{t \in T_m} \left[c_t^g x_t^g + d_t^g y_t^g + S_U^g \pi_{U_t}^g + S_D^g \pi_{D_t}^g \right] + (\omega_m^g) z_m^g \right] \\ & + \sum_{g \in G_r} \alpha_g \left(\frac{1}{|M|} - z_m^g \right) \end{aligned}$$

Let $\bar{\boldsymbol{\vartheta}} = \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \boldsymbol{\pi}_U, \boldsymbol{\pi}_D, \boldsymbol{\theta}, \mathbf{f}\}$ be the optimal solution to $\mathcal{L}_r(\Delta)$ (i.e. Problem (4)). With this solution fixed, $\bar{\mathbf{z}}$ ensures $(1 - \sum_{m \in M} z_m^g) = 0 \quad \forall g \in G_r$, which would enforce

i) $\sum_{g \in G_r} \alpha_g (1 - \sum_{m \in M} z_m^g) = 0$, and ii) $\hat{\mathcal{L}}_r|_{\bar{\boldsymbol{\vartheta}}} = \mathcal{L}_r(\Delta)|_{\bar{\boldsymbol{\vartheta}}}$.

Let $\boldsymbol{\vartheta}'$ and \mathcal{F} be the optimal solution, and the set of feasible solutions for $\hat{\mathcal{L}}_r$, respectively. Note that the solution $\bar{\boldsymbol{\vartheta}}$ is still feasible for the lagrangian relaxation (i.e. $\bar{\boldsymbol{\vartheta}} \in \mathcal{F}$), it is trivial to conclude $\hat{\mathcal{L}}_r|_{\boldsymbol{\vartheta}'} = \min_{\boldsymbol{\vartheta} \in \mathcal{F}} \hat{\mathcal{L}}_r|_{\boldsymbol{\vartheta}} \leq \hat{\mathcal{L}}_r|_{\bar{\boldsymbol{\vartheta}}}$. Since, $\hat{\mathcal{L}}_r|_{\boldsymbol{\vartheta}'} \leq \hat{\mathcal{L}}_r|_{\bar{\boldsymbol{\vartheta}}} = \mathcal{L}_r(\Delta)|_{\bar{\boldsymbol{\vartheta}}}$, we conclude:

$$\min_{\bigcup_{m \in M} \mathcal{Q}_m^r} \hat{\mathcal{L}}_r = \sum_{m \in M} \min_{\mathcal{Q}_m^r} \mathcal{L}_r^m(\Delta, \alpha) \leq \min \mathcal{L}_r(\Delta) \quad (13)$$

We note that minimizing $\hat{\mathcal{L}}_r$ is equivalent to minimizing each

of the individual terms \mathcal{L}_r^m since their respective constraint sets \mathcal{Q}_m^r 's are disjoint $\forall m \in M$. \square

As a result of Lemma 1, in order to perform one solve of Problem (4) it suffices to iteratively solve \mathcal{L}_r^m in a multithreaded parallel fashion $\forall m \in M$ followed by an update of α_g using the subgradient method outlined in [25].

IV. ALGORITHM DESIGN FOR JOINT DECENTRALIZED MAINTENANCE AND OPERATIONS

We divide the joint, decentralized optimization algorithm into four distinct parts. Each component pertains to the local multithreaded solver, peer-to-peer communication scheme, decentralized optimization frameworks based on convex relaxation and the subgradient methods respectively.

A. Local Multithreaded Optimization Solver

```

function MTOPT( $\alpha, \mathcal{L}_r(\Delta), \mathcal{Q}^r$ )
  for  $m = 1, 2, \dots, |M|$  using multithreading do
    solve  $\min_{\mathcal{Q}_m^r} \mathcal{L}_r^m(\Delta, \alpha)$  using Lemma 1
  end for
  return  $\{x, y, z, \theta, f, \varphi\}$ 
end function

```

The entire decentralized joint problem relies on a local optimization solver represented by function `MTOPT`. Owing to its completely decoupled nature along the maintenance planning horizon as detailed in Lemma 1, we apply multithreading to solve Problem (4) in parallel to boost computational efficiency. Therefore, given an objective function $\mathcal{L}_r(\Delta)$, a Lagrangian estimate α of Constraint (4b) and a constraint set \mathcal{Q}^r , function `MTOPT` applies M threads to solve one iteration of the local joint problem.

B. Peer to peer Communication

```

function COMMUNICATE( $k, \Delta^{k-1}, \theta^k, f^k, \varphi^k$ )
  send  $\theta^{b,k}$  to all regions  $r', \forall r' \in \mathcal{N}_r, b \in \mathcal{B}_{r'}$ 
  receive  $\tilde{\theta}^{b,k}$  from all regions  $r', \forall r' \in \mathcal{N}_r, b \in \mathcal{B}_{r'}$ 
  send  $\varphi^k$  and receive  $\varphi_{r'}^k$  to and from all regions  $r' \in \mathcal{R}$ 
  compute  $\Delta^k$  based on Equations (5)-(10)
  if  $\|f^k - \bar{f}^k\| < \epsilon$  and  $\|f^k - \bar{f}^{k-1}\| < \epsilon$  then
    set local convergence to true
    if local convergence is true  $\forall r \in \mathcal{R}$  then  $\Omega \leftarrow 1$ 
  end if
  return  $\{\Delta^k, \Omega\}$ 
end function

```

Communication between neighboring regions occurs according to the steps represented in function `Communicate` which is invoked after every local solve. At every round k , estimates of the phase θ^k , flow f^k , local violation φ^k , their corresponding consensus values embodied in Δ^{k-1} are taken as inputs. The phase angle estimates are sent to neighbors and are used to compute fresh estimates of the intermediate values Δ^k . The local load violation $\varphi^{r,k}$ is also communicated to all regions and is used to compute the customized local production target. A local convergence check follows based on the consensus estimates of the flow. The global convergence status is denoted by Ω . In case of local convergence of all regions, the problem is deemed to have globally converged and Ω is set to 1.

C. Decentralized Optimization with Fixed Maintenance

```

function DECENTFIXEDOPT( $\Delta, \mathcal{Q}^r$ )
   $k \leftarrow 0, \Delta_0 \leftarrow \Delta, \alpha \leftarrow 0$ 
  set global convergence value  $\Omega \leftarrow 0$ 
  while  $\Omega \neq 1$  do
     $k \leftarrow k + 1$ 
     $\{x^k, y^k, z^k, \theta^k, f^k, \varphi^k\} \leftarrow$  MTOPT( $\alpha, \mathcal{L}_r(\Delta^{k-1}), \mathcal{Q}^r$ )
     $\{\Delta^k, \Omega\} \leftarrow$  COMMUNICATE( $k, \Delta^{k-1}, \theta^k, f^k, \varphi^k$ )
  end while
  return  $\{x^k, y^k, z^k, \Delta^k\}$ 
end function

```

For solving the joint problem with the maintenance decisions fixed, we follow the sequence of steps represented in function `DecentFixedOpt`. Since maintenance decisions are fixed, the Lagrangian α is also fixed to 0. A multithreaded local solve followed by a peer to peer communication forms one round of the decentralized joint framework. The sequence of computation followed by communication occurs synchronously among all regions until global convergence is achieved.

D. Decentralized Optimization with Local Subgradients

```

function DECENTSGOPT( $\Delta, \mathcal{Q}^r$ )
   $k \leftarrow 0, \Delta_0 \leftarrow \Delta, \alpha^0 \leftarrow 0$ 
  set global convergence value  $\Omega \leftarrow 0$ 
  while  $\Omega \neq 1$  do
     $k \leftarrow k + 1, j \leftarrow 0, \Delta^{sg} \leftarrow \Delta^{k-1}$ 
    while Constraint (4b) not satisfied do
       $\{x^j, y^j, z^j, \theta^j, f^j, \varphi^j\} \leftarrow$  MTOPT( $\alpha^j, \mathcal{L}_r(\Delta^{sg}), \mathcal{Q}^r$ )
       $\mathcal{L}^j = c_r^T x^j + d_r^T y^j + \omega_r^T z^j$ 
       $\alpha_g^{j+1} = \alpha_g^j + \sigma^j (1 - \sum_{m \in M} z_m^{g,j}), \forall g \in G_r$ 
      update  $\sigma^{j+1}$  based on Equation (15)
       $\{x^k, y^k, z^k, \theta^k, f^k, \varphi^k\} \leftarrow \{x^j, y^j, z^j, \theta^j, f^j, \varphi^j\}$ 
    end while
     $\{\Delta^k, \Omega\} \leftarrow$  COMMUNICATE( $k, \Delta^{k-1}, \theta^k, f^k, \varphi^k$ )
  end while
  return  $\{x^k, y^k, z^k, \Delta^k\}$ 
end function

```

When the maintenance decisions are released, we utilize the sequence of steps detailed in function `DecentSGOpt`. Specifically, the framework outlined in [25] is utilized to update the Lagrangian value α_g which is iteratively updated using the Equation 14

$$\alpha_g^{j+1} = \alpha_g^j + \sigma^j (1 - \sum_{m \in M} z_m^{g,j}), \forall g \in G_r \quad (14)$$

The step size for the subgradient method is given by Equation (15) where $\mathcal{L}^j, \mathcal{L}_{UB}$ represent the joint operational and maintenance objective costs of the j^{th} local subgradient iteration and its upper bound respectively.

$$\sigma_j = \frac{|\mathcal{L}_{UB} - \mathcal{L}^j|}{\sum_{g \in G_r} \left[1 - \sum_{m \in M} z_m^{g,j} \right]^2} \quad (15)$$

E. Decentralized Multithreaded Joint Maintenance Algorithm

Based on the four main functions mentioned before, we now present the decentralized multithreaded (DMT) joint optimization solution framework in Algorithm 1. We begin by initializ-

Algorithm 1 Decentralized Multithreaded(DMT) Algorithm for Joint CBM and Operations

$\{x, y, z, \Delta\}_{FMRC} \leftarrow \text{DECENTFIXEDOPT}(\Delta_0, Q_{FMRC}^r)$
 $\{x, y, z, \Delta\}_{FMBC} \leftarrow \text{DECENTFIXEDOPT}(\Delta_{FMRC}, Q_{FMBC}^r)$
 compute \mathcal{L}_{UB} based on $\{x, y, z\}_{FMBC}$
 $\{x, y, z, \Delta\} \leftarrow \text{DECENTSGOPT}(\Delta_{FMBC}, Q_{BMBC}^r)$

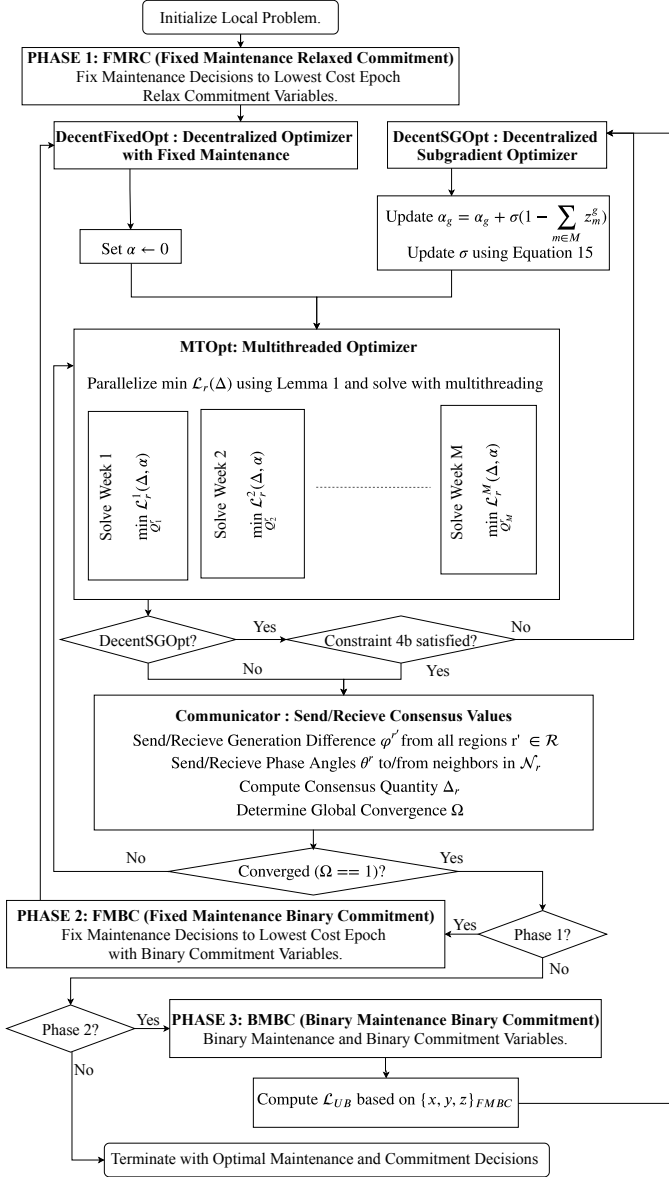


Fig. 2: Flowchart depicting the DMT algorithm at region r

ing the intermediate values and delineating the three different constraint sets Q_{FMRC}^r , Q_{FMBC}^r and Q_{BMBC}^r . Set Q_{FMRC}^r consists of maintenance decisions fixed to the lowest epoch of the maintenance cost function with relaxed commitment variables. As a result, Q_{FMRC}^r is convex in nature. Further, Q_{FMBC}^r represents the constraint set with fixed maintenance and binary commitment variables. It is easy to see that an optimal solution of Problem (4) subject to Q_{FMBC}^r is a feasible solution and also an upper bound of the joint problem as well. Moreover, a solution to Problem (4) with respect to Q_{FMBC}^r is purely operations oriented. Finally, Q_{BMBC}^r

represents a constraint set with released maintenance variables and binary commitment variables.

Initially, Algorithm 1 globally converges with respect to Q_{FMRC}^r . Next, the resulting intermediate values are used as input to converge with respect to Q_{FMBC}^r . Based on the optimal solution obtained in the previous step, an upper bound of the joint solution is determined that in turn forms the basis of the subgradient method. Warm starting with the help of the intermediate values of the previous step, the local solves based on the subgradient method globally converge to yield the joint maintenance and operations solution.

A flowchart of the DMT algorithm has been provided in Figure 2. The flowchart reflects the program logic of Algorithm 1 at a regional level. The process starts with an initialized local problem, where the maintenance variables would be fixed, and relaxed commitment decisions would be optimized using the Phase 1 (Fixed Maintenance Relaxed Commitment) formulation. The balance of consensus quantities achieved among neighbor regions in Phase 1 is used to warm start Phase 2 (Fixed Maintenance Binary Commitment), wherein the binary decisions are obtained with respect to the maintenance decisions fixed in Phase 1. Upon converging with respect to Phase 2, an upper bound on the joint CBM and operations problem is obtained regionally. Next, the subgradient method is instantiated across all regions which in turn takes advantage of the consensus achieved thus far. The subgradient method in Phase 3 is applied iteratively until Constraint (4b) is locally met to complete one regional iteration. These regional iterations are performed by all regions until global convergence is achieved.

While Figure 2 presents the regional program flow, Figure 3 depicts the ADMM based coordination employed in Algorithm 1 based on the sample power network illustrated in Figure 1. Broadly, there are three main steps concerning the consensus process. Under Step 1, the regions perform multithreaded, local computation using the three different phases illustrated in Figure 2. After local computation, in Step 2, the phase angles and generation difference values are exchanged with neighbors and all other regions respectively. In Step 3, the received values from other regions are utilized to compute the respective Δ values which are used as an input in the subsequent iteration of DMT.

V. EXPERIMENTAL RESULTS

For our experiments, we used Gurobi 7.5 [26] for solving the MIQP problem represented by (3) on each node. All our experiments were run with 128GB of memory on a state of the art super computer [27]. We evaluate our model on the IEEE 3012 bus case with data derived from MATPOWER library [28]. In order to model the regions, we utilize the region decompositions as well as the data provided in [9] for the IEEE 3012 bus case for 150 generators. In all our experiments, the penalty terms ρ_θ , ρ_f and ρ_p are fixed at a value of 1.

A. Computational Architecture

For our experiments, we use an implementation that is based on the Message Passing Interface (MPI)[29] framework for decentralized computation and OpenMP for multithreading. The computational architecture for the sample power network

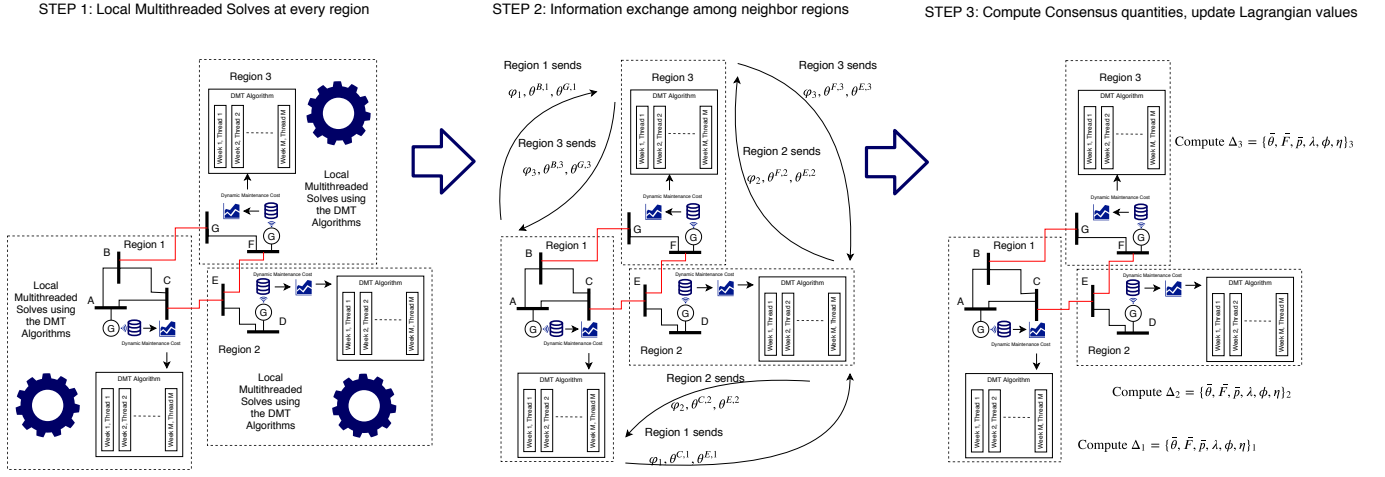


Fig. 3: Schematic describing the DMT algorithm in the context of peer-to-peer consensus of phase angle and flow values

depicted in Figure 1 is presented in Figure 4. The architecture follows a hybrid MPI-OpenMP architecture where each region is assigned one MPI process and every maintenance epoch for the regional subproblem is assigned to one OpenMP thread. Using a distributed memory framework like MPI for decentralization helps us evaluate the algorithm in an environment close to the *real-world*, where each region may represent various utilities. Our multithreaded decentralized computational architecture and software framework can be easily extended to a geographically distributed computational environment.

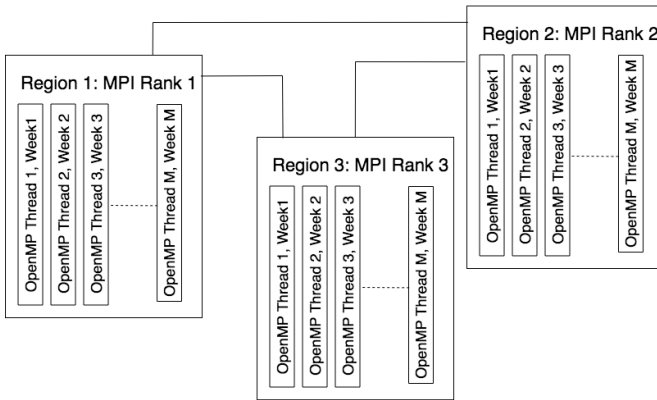


Fig. 4: Hybrid MPI OpenMP Computational Architecture for solving the Joint CBM and Operations problem

B. Degradation Modeling

In order to obtain the sensor data necessary to derive the maintenance cost of the assets, we rely on vibration data acquired from a rotating machinery apparatus. Using this experimental setup, condition monitoring is employed to estimate the degradation of the rolling element bearing present in the rotating machine apparatus. We chose roller bearings because they represent an integral component of every rotating machinery including generators and different types of turbines [30], [31]. Vibration signals from the bearings are used to represent generator degradation. We follow an experimental

setup outlined in [22] that traces the degradation of bearings from their new state until their failure.

C. Benchmark Models

For all our experiments, we consider a planning horizon of 12 weeks. To highlight the benefits of our decentralized multithreaded approach, we resolve the problem under different operational complexity cases. More specifically, the operational complexity in our setting is quantified by the commitment decisions per generator per day (CGD). In order to evaluate the computational benefits of our framework, we consider four different benchmark models.

- 1) *Centralized Benchmark*: Our centralized benchmark comprises a direct solve of the large scale problem formulation by Gurobi without any decomposition schemes. We use the centralized benchmark to rate the performance of the decentralized algorithm by measuring the relative optimality gap. We let γ denote the total optimal objective value comprised of operations and CBM components. γ_{decent} and γ_c represent the optimal objective for the decentralized joint method and its centralized counterpart, respectively. Solving a centralized version of the joint problem could be significantly challenging in itself. In our experiments, the centralized version did not converge within 6 hours for even the simplest operational complexity of 4 CGD. Therefore, our centralized benchmark formulation considers its relaxation without transmission line constraints. By relaxing both the integrality constraints of the commitment decisions, and the transmission line constraints in the centralized benchmark, we report the performance of the other models as a function of a conservative lower bound for the centralized problem. This means that the optimality gaps reported by the decentralized solution may actually be lower. We also note that an exact theoretical rate of convergence cannot be provided owing to the mixed integer and consequently NP-hard nature of the joint problem as indicated in [17]. We do show that our method provides a solution quality that consistently provides an optimality gap of approximately 1%.

- 2) *Two Stage APMII Benchmark*: The Two Stage APMII Benchmark is based on a modified Benders decomposition strategy introduced in [8]. The joint formulation is decomposed into a maintenance-driven master problem as well as weekly operational subproblems that are captured through Benders and integer-type cuts. Driven inherently by a Benders decomposition, this two stage method represents the current state of the art for solving the joint CBM and operations problem. Cutting plane methods (such as this benchmark) are inherently limited by the complexity of the weekly subproblems in parallel as well as the growing size of the master problem.
- 3) *Decentralized Single Threaded (DST) Benchmark*: This benchmark is based on the current state-of-the-art decentralized methods highlighted in [9], [10], [11]. It employs a region based decomposition of the power network topology. An ADMM based technique is used to iteratively balance the electricity flow among the regions themselves, which iteratively leads to convergence to the optimal cost. The DST benchmark employs only a single thread at every region and as a result suffers significant performance degradation compared to DMT.
- 4) *Decentralized Solver Driven Multithreading (SMT) Benchmark*: This benchmark augments DST with solver (i.e. Gurobi) driven multithreading. In this strategy, we employ the same region driven decomposition with ADMM based flow balance. However, instead of every region being given a single thread, multiple threads are assigned to the optimization solver itself at every region. Consequently, the mechanics of multithreading are completely handled by the solver without any added decomposition strategies from the formulation perspective. Therefore, such a strategy represents the most stringent, state-of-the art benchmark for comparing the decentralized multithreading scheme proposed in this paper.

D. Computational Results and Observations

TABLE I: Centralized Results Objective Costs (USD 10^4)

| CGD | Variables | Ops | CBM | DC | Obj |
|-----|-----------|--------|-------|------|--------|
| 4 | 100800 | 641.70 | 69.80 | 2.66 | 714.16 |
| 8 | 201600 | 626.66 | 69.81 | 5.12 | 701.60 |
| 12 | 403200 | 624.44 | 69.84 | 7.58 | 701.86 |

We present our results in terms of the calculated speedup offered by the DMT algorithm as compared to the corresponding benchmarking method. The speedup is calculated as the ratio of the solution times incurred by the benchmark with respect to those of the DMT.

1) Centralized Benchmark Comparison

Table I depicts the operational, CBM, demand curtailment and total costs of the centralized algorithm with varying number of CGD values. We use the centralized results for benchmarking the solution quality of our decentralized, multithreaded algorithm. Table II depicts the operational (Ops), CBM, Demand Curtailment (DC) and Total Objective cost with respect to increasing CGD values with varying number of regions. Further, Table III shows the computational per-

formance of the decentralized algorithm with respect to the optimality gap, the computational time incurred as well as the total number of threads used. Specifically, we show the results with varying CGD and region cases.

We observe that with increasing complexity of the problem as indicated by higher CGD values, the computational time keeps increasing. Further, Table II shows that our decentralized approach yields highly stable solutions with respect to optimal maintenance decisions as well as operational decisions. On the other hand Table III yields numerous interesting insights concerning the computational performance of the decentralized algorithm. First, we observe that with increasing number of regions, the time incurred by our algorithm does not increase by a significant amount. In fact, our approach is seen to be highly stable with increasing number of regions thereby highlighting scalability. Moreover, we can also note that with increasing complexity of the problem denoted by the CGD value, the time incurred increases approximately in a linear fashion thereby demonstrating computational efficiency. Lastly, we observe that in terms of quality of solution as indicated by the optimality gap, our algorithm retains its stability with increasing number of regions across varying CGD values.

Table IV presents the performance in terms of the speedup offered by DMT in comparison with the centralized method. We observe that the DMT algorithm consistently solves the joint problem at least three times as fast as the centralized counterpart. In fact, it can be seen from Table IV, that the DMT can be 50 times faster than the conventional centralized solution method.

2) Two Stage APMII Benchmark Comparison

Table V represents the results relating the Two Stage APMII benchmark with respect to varying CGD. For these benchmark experiments, we provided a maximum time limit of 12 hours. The weekly operational subproblems in our benchmark implementation were also parallelized using MPI. We note that none of the experiments pertaining to APMII converged within the specified time limit. Therefore, we present only the projected speedup values, which evidently represents the lower bound on the speedup offered by the DMT algorithm.

Table V clearly demonstrates that the DMT can provide a speedup by a factor of 22 in certain cases. The speedup values largely remain stable across increasing number of regions however, they decrease slightly with increasing CGD complexity. Even at its slowest, DMT is still approximately at least twice as fast as the APMII method. Therefore, results in Table V indicate that the DMT algorithm outperforms state-of-the-art Two Stage APMII method in all the cases that we have studied.

3) Decentralized Single Threaded (DST) Benchmark

For comparisons with respect to DST, convergence was very slow on account of no multithreading support available to the solver at the regional level. We restricted our benchmark experiments to the 4 CGD case across all regional decompositions and the 100 region case with varying computational complexity depicted in Tables VII, VIII respectively.

Table VII demonstrates that the DMT algorithm can be faster by up to a factor of 18 with the 4 CGD case. While

TABLE II: Final Objective Costs (USD 10^4) of Decentralized Joint CBM and Operations framework

| No. Of Regions | 4 CGD | | | | 8 CGD | | | | 12 CGD | | | |
|----------------|--------|-------|-------|--------|--------|-------|-------|--------|--------|-------|-------|--------|
| | Ops | CBM | DC | Obj | Ops | CBM | DC | Obj | Ops | CBM | DC | Obj |
| 50 | 641.16 | 69.82 | 6.85 | 717.82 | 627.89 | 69.80 | 6.96 | 704.65 | 626.12 | 69.86 | 9.96 | 705.95 |
| 60 | 641.05 | 69.82 | 8.79 | 719.66 | 627.03 | 69.83 | 8.96 | 705.82 | 625.70 | 69.80 | 8.43 | 703.94 |
| 75 | 640.32 | 69.84 | 10.48 | 720.65 | 626.82 | 69.87 | 10.72 | 707.40 | 626.78 | 69.81 | 8.34 | 704.93 |
| 80 | 638.32 | 69.83 | 12.37 | 720.52 | 625.13 | 69.81 | 12.24 | 707.18 | 624.93 | 69.83 | 11.27 | 706.03 |
| 90 | 639.25 | 69.84 | 12.52 | 721.61 | 623.68 | 69.82 | 13.17 | 706.67 | 623.64 | 70.01 | 18.82 | 712.47 |
| 100 | 637.93 | 69.85 | 14.34 | 722.12 | 622.80 | 69.91 | 18.05 | 710.76 | 620.61 | 69.88 | 20.07 | 710.56 |

TABLE III: Computational Performance of Decentralized Joint CBM and Operations framework

| No. Of Regions | No. Of Threads | 4 CGD | | 8 CGD | | 12 CGD | |
|----------------|----------------|------------|-------------|------------|-------------|------------|------------|
| | | Opt Gap(%) | Time (mins) | Opt Gap(%) | Time (mins) | Opt Gap(%) | Time(mins) |
| 50 | 600 | 0.51 | 33.77 | 0.43 | 52.39 | 0.58 | 122.74 |
| 60 | 720 | 0.77 | 44.58 | 0.59 | 53.54 | 0.29 | 170.89 |
| 75 | 900 | 0.90 | 49.8 | 0.82 | 219.17 | 0.43 | 111.50 |
| 80 | 960 | 0.89 | 31.49 | 0.79 | 50.98 | 0.59 | 56.50 |
| 90 | 1080 | 1.04 | 33.24 | 0.71 | 35.02 | 1.51 | 63.06 |
| 100 | 1200 | 1.11 | 41.37 | 1.2 | 39.88 | 1.23 | 368.98 |

TABLE IV: Centralized Solution Benchmarking

| CGD | Time (mins) | DMT Speedup | | | | | |
|-----|-------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | 50 Regions | 60 Regions | 75 Regions | 80 Regions | 90 Regions | 100 Regions |
| 4 | 120.85 | 3.58 | 2.71 | 2.43 | 3.84 | 3.64 | 2.92 |
| 8 | 611.43 | 11.67 | 11.42 | 2.79 | 11.99 | 17.46 | 15.33 |
| 12 | 2846.17 | 23.19 | 16.66 | 25.53 | 50.38 | 45.14 | 7.71 |

TABLE V: Two Stage APMII Benchmark

| CGD | Convergence Status ¹ | Obj | (% Gap from centralized) | Lower Bound on DMT Speedup | | | | | |
|-----|---------------------------------|--------|--------------------------|----------------------------|---------------|--------------|--------------|--------------|--------------|
| | | | | 50 Regions | 60 Regions | 75 Regions | 80 Regions | 90 Regions | 100 Regions |
| 4 | NC | 712.37 | 0.23 | 21.32 | 16.15 | 14.46 | 22.86 | 21.66 | 17.40 |
| 8 | NC | 699.93 | 0.22 | 13.74 | 113.45 | 3.29 | 14.12 | 20.56 | 18.06 |
| 12 | NC | 699.62 | 0.26 | 5.87 | 4.21 | 6.46 | 12.74 | 11.42 | 1.95 |

TABLE VI: Decentralized Solver Driven Multithreading (SMT) Benchmark

| No. Of Regions | 4 CGD | | | 8 CGD | | | 12 CGD | | |
|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | Opt Gap (%) | Time (mins) | DMT Speedup | Opt Gap (%) | Time (mins) | DMT Speedup | Opt Gap (%) | Time (mins) | DMT Speedup |
| 50 | 0.38 | 197.62 | 5.85 | 0.31 | 306.11 | 5.84 | 0.29 | 278.46 | 2.27 |
| 60 | 0.36 | 113.87 | 2.55 | 0.50 | 155.77 | 2.91 | 0.18 | 301.56 | 1.76 |
| 75 | 0.43 | 181.06 | 3.64 | - | 210.05 | * | - | 276.25 | * |
| 80 | 0.41 | 106.87 | 3.39 | - | 244.62 | * | 0.22 | 194.45 | 3.44 |
| 90 | 0.39 | 83.51 | 2.51 | - | 199.73 | * | - | 203.27 | * |
| 100 | - | 182.28 | * | - | 228.17 | * | - | 209.23 | * |

TABLE VII: DST Benchmark with respect to CGD=4

| Regions | Opt Gap (%) | Time(mins) | DMT Speedup |
|---------|-------------|------------|--------------|
| 50 | 0.33 | 594.32 | 17.60 |
| 60 | 0.30 | 943.80 | 21.17 |
| 75 | 0.46 | 391.69 | 7.87 |
| 80 | 0.40 | 260.46 | 8.27 |
| 90 | 0.37 | 197.03 | 5.93 |
| 100 | 0.50 | 107.61 | 2.60 |

TABLE VIII: Performance Comparison of DST with 100 Region Decomposition

| CGD | Opt Gap (%) | Time(mins) | DMT Speedup |
|-----|-------------|------------|--------------|
| 4 | 0.50 | 107.61 | 2.60 |
| 8 | 0.49 | 1,967.09 | 49.33 |
| 12 | 0.39 | 2,776.94 | 7.53 |

the speedup of DMT decreases with increasing number of regions, it is still faster than DST at least by a factor of 2.6

for the 100 region case. On the other hand, analysis of results in Table VIII reveal that DMT can be 50 times faster than

¹A convergence status of NC refers to Non Convergence

DST while still delivering a healthy speedup across all degrees of complexity. Tables VII and VIII show the degradation of performance with increasing complexity as well as regions as evidenced by convergence times stretching up to 48 hours (for the 12 CGD, 100 regions case).

4) Decentralized Solver Driven Multithreading Benchmark

Table VI represents the results obtained by allowing the solver to leverage multithreading itself. In this case, in order to compare with the DMT technique, we allocate 12 threads to the solver per region while keeping the same experimental settings. Table VI depicts the optimality gap, the time as well as the speedup offered by DMT in comparison with the benchmark method. Rows marked * represent instances where the solver ran out of memory along with the time taken to reach the out of memory exception.

Table VI reveals several interesting trends. First, we can see that even with a state-of-the-art solver like Gurobi, the DMT method delivers consistent speedup values of up to a factor of 6. Further, out of memory exceptions increase with the rise in complexity. This can be directly attributed to the increasing number of binary variables encountered with increasing CGD values. As a result, more memory is consumed by the multithreading schemes on commercial solvers, leading to a poor space complexity and scalability.

VI. CONCLUSION AND FUTURE WORK

In this paper we develop a decentralized, multithreaded framework for the joint CBM and operational problem designed for large scale power systems. Our solution involves, decomposing a given power network topology into multiple regions and using ADMM to formulate a joint optimization model. Such a formulation based on decentralization, allows preservation of regional data privacy. We solve the optimization model in a decentralized manner wherein each region holds its own local subproblem and cooperates with its neighbors by exchanging flow estimates. We further leverage multithreading at every regional subproblem to bolster the computational efficiency of our solution.

We demonstrate the convergence of our algorithm based on experiments on the large scale IEEE 3012 bus case incorporating varying degrees of region decompositions. We benchmark our methodology against four different computational paradigms i.e., centralized, two stage cutting plane, decentralized single threaded as well as decentralized solver driven multithreaded methods. Our results show that our decentralized multithreaded algorithm consistently outperforms all four benchmarking methods offering as high as 50x speedups with a stable solution quality as well. The results demonstrate that our decentralized algorithm can provide good solution quality, scalability and efficiency with full privacy of sensor data while being robust to varying problem complexities. As part of our future work, we intend to explore applications of the proposed framework into coordination mechanisms for different participants within deregulated electricity markets, and computational algorithms that can harness further efficiency through asynchronous model of operation as well as integrating differential privacy for consensus.

ACKNOWLEDGEMENT

This paper is dedicated to Sudha Venkat Ramanan, a constant source of inspiration and whose love, support and understanding played a major role in making this research happen.

REFERENCES

- [1] N. P. Padhy, "Unit commitment-a bibliographical survey," *IEEE Transactions on Power Systems*, vol. 19, pp. 1196–1205, May 2004.
- [2] R. Billinton, "Composite system maintenance coordination in a deregulated environment," *IEEE Transactions on Power Systems*, vol. 20, pp. 485–492, Feb 2005.
- [3] C. Feng and X. Wang, "A competitive mechanism of unit maintenance scheduling in a deregulated environment," *IEEE transactions on power systems*, vol. 25, no. 1, pp. 351–359, 2010.
- [4] M. Marwali and S. Shahidehpour, "Integrated generation and transmission maintenance scheduling with network constraints," *IEEE Transactions on Power Systems*, vol. 13, no. 3, pp. 1063–1068, 1998.
- [5] Y. Fu, M. Shahidehpour, and Z. Li, "Security-constrained optimal coordination of generation and transmission maintenance outage scheduling," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1302–1313, 2007.
- [6] Y. Fu, Z. Li, M. Shahidehpour, T. Zheng, and E. Litvinov, "Coordination of midterm outage scheduling with short-term security-constrained unit commitment," *IEEE Transactions on Power Systems*, vol. 24, no. 4, pp. 1818–1830, 2009.
- [7] M. Yildirim, X. A. Sun, and N. Z. Gebraeel, "Sensor-driven condition-based generator maintenance scheduling part i: Maintenance problem," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1–10, 2016.
- [8] M. Yildirim, X. A. Sun, and N. Z. Gebraeel, "Sensor-driven condition-based generator maintenance scheduling part ii: Incorporating operations," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1–9, 2016.
- [9] M. J. Feizollahi, M. Costley, S. Ahmed, and S. Grijalva, "Large-scale decentralized unit commitment," *International Journal of Electrical Power & Energy Systems*, vol. 73, pp. 97 – 106, 2015.
- [10] P. Ramanan, M. Yildirim, E. Chow, and N. Gebraeel, "Asynchronous decentralized framework for unit commitment in power systems," *Procedia Computer Science*, vol. 108, pp. 665–674, 2017.
- [11] P. Ramanan, M. Yildirim, E. Chow, and N. Gebraeel, "An asynchronous, decentralized solution framework for the large scale unit commitment problem," *IEEE Transactions on Power Systems*, vol. 34, pp. 3677–3686, Sep. 2019.
- [12] Á. S. Xavier, F. Qiu, and S. S. Dey, "Decomposable formulation of transmission constraints for decentralized power systems optimization," *arXiv preprint arXiv:2001.07771*, 2020.
- [13] B. J. Walker, "Multiyear plan for energy sector cybersecurity," *Office of Electricity Delivery and Energy Analysis, U.S. Department of Energy*, March 2018.
- [14] C. Glenn, D. Sterbentz, and A. Wright, "Cyber threat and vulnerability analysis of the us electric sector," *US DOE Office of Energy Policy and Systems Analysis (EPSA)*, December 2016.
- [15] D. Li, K. Paynabar, and N. Gebraeel, "A degradation-based detection framework against covert cyberattacks on scada systems," *IIEE Transactions*, vol. 53, no. 7, pp. 812–829, 2021.
- [16] D. Li, N. Gebraeel, and K. Paynabar, "Detection and differentiation of replay attack and equipment faults in scada systems," *IEEE Transactions on Automation Science and Engineering*, 2020.
- [17] R. Zhang and J. T. Kwok, "Asynchronous distributed admm for consensus optimization," in *ICML*, pp. 1701–1709, 2014.
- [18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, Jan. 2011.
- [19] A. Papavasiliou, S. S. Oren, and B. Rountree, "Applying high performance computing to transmission-constrained stochastic unit commitment for renewable energy integration," *IEEE Transactions on Power Systems*, vol. 30, pp. 1109–1120, May 2015.
- [20] I. Aravena and A. Papavasiliou, "A distributed asynchronous algorithm for the two-stage stochastic unit commitment problem," in *2015 IEEE Power Energy Society General Meeting*, pp. 1–5, July 2015.
- [21] A. Abiri-Jahromi, M. Fotuhi-Firuzabad, and M. Parvania, "Optimized midterm preventive maintenance outage scheduling of thermal gener-

- ating units,” *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1354–1365, 2012.
- [22] N. Z. GEBRAEEL, M. A. LAWLEY, R. LI, and J. K. RYAN, “Residual-life distributions from component degradation signals: A bayesian approach,” *IIE Transactions*, vol. 37, no. 6, pp. 543–557, 2005.
- [23] A. H. Elwany and N. Z. Gebraeel, “Sensor-driven prognostic models for equipment replacement and spare parts inventory,” *IIE Transactions*, vol. 40, no. 7, pp. 629–639, 2008.
- [24] L. Wu, M. Shahidehpour, and Y. Fu, “Security-constrained generation and transmission outage scheduling with uncertainties,” *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1674–1685, 2010.
- [25] M. L. Fisher, “The lagrangian relaxation method for solving integer programming problems,” *Management science*, vol. 50, no. 12_supplement, pp. 1861–1871, 2004.
- [26] I. Gurobi Optimization, “Gurobi optimizer reference manual,” 2017. https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/7.5/refman.pdf.
- [27] PACE, *Partnership for an Advanced Computing Environment (PACE)*, 2017. <http://www.pace.gatech.edu>.
- [28] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, “Matpower: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Transactions on Power Systems*, vol. 26, pp. 12–19, Feb 2011.
- [29] L. Dalcín, R. Paz, M. Storti, and J. D’Elía, “Mpi for python: Performance improvements and mpi-2 extensions,” *Journal of Parallel and Distributed Computing*, vol. 68, no. 5, pp. 655 – 662, 2008.
- [30] A. Sugarman, “Condition monitoring of electrical equipment in nuclear power plants,” *IEEE Transactions on Energy Conversion*, vol. EC-1, pp. 1–8, Sep. 1986.
- [31] T. Harris, *Rolling bearing analysis*. John Wiley and Sons, Inc.(United States), 1991.

APPENDIX: IMPLEMENTATION NOTES

The experiments were performed on the Hive supercomputer at the Georgia Institute of Technology, Atlanta, GA, USA. The codebase used in the experiments consisted of C++ as well as Python based optimization modules that we developed for Gurobi 7.5. Python 2.7 was used for implementation including the creation of the joint CBM and operations optimization models. To improve computational efficiency, the solver modules were written in C++ with Cython wrappers for easy and fast access from Python.

There were numerous reasons for adopting a mixed C++ and Python oriented framework. Support for multithreading frameworks like OpenMP is largely absent in Python. For solving the same optimization model, there was a 10x reduction in compute time in case of a C++ implementation as compared to a Python based one.

Paritosh Ramanan Paritosh Ramanan is a Postdoctoral Fellow with the Georgia Institute of Technology in Atlanta, Georgia. He got his PhD in Computational Science and Engineering from the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology in Atlanta, Georgia in 2020. Prior to his PhD he earned a Masters in Computer Science from Georgia State University in Atlanta, Georgia in 2015 and obtained his Bachelors in Information Systems from Birla Institute of Technology and Science (BITS) Pilani, Goa Campus in 2013. His research focuses on developing decentralized algorithms for improved computational performance of large scale optimization problems through the use of parallel and distributed computing paradigms.

Murat Yildirim Dr. Murat Yildirim is an Assistant Professor in the Department of Industrial and Systems Engineering at Wayne State University. Prior to joining Wayne State, he worked as a postdoctoral fellow at the Georgia Institute of Technology (2016-2018), and obtained a Ph.D. degree in Industrial Engineering, and B.Sc. degrees in Electrical and Industrial Engineering from the same institution. Dr. Yildirim’s research interest lies in advancing the integration of mathematical programming and data analytics in large scale energy systems. Specifically, he focuses on the modeling and the computational challenges arising from the integration of real-time sensor inferences into large-scale mixed integer programs (MIPs) used for optimizing and controlling networked systems.

Nagi Gebraeel Dr. Nagi Gebraeel is the Georgia Power Early Career Professor and Professor in the Stewart School of Industrial and Systems Engineering at Georgia Tech. His research interests lie at the intersection of industrial predictive analytics and decision optimization models for large scale power generation applications. Dr. Gebraeel serves as an associate director at Georgia Tech’s Strategic Energy Institute and the director of the Analytics and Prognostics Systems laboratory at Georgia Tech’s Manufacturing Institute. Dr. Gebraeel was the former president of the Institute of Industrial Engineers (IIE) Quality and Reliability Engineering Division, and is currently a member of the Institute for Operations Research and the Management Sciences (INFORMS).

Edmond Chow Edmond Chow is an Associate Professor in the School of Computational Science and Engineering at Georgia Institute of Technology. He previously held positions at D. E. Shaw Research and Lawrence Livermore National Laboratory. His research is in developing numerical methods specialized for high-performance computers and applying these methods to enable the solution of large-scale physical simulation problems in science and engineering. Dr. Chow received the 2009 ACM Gordon Bell prize and a 2002 Presidential Early Career Award for Scientists and Engineers (PECASE). He is a Fellow of the Society for Industrial and Applied Mathematics.