Multi-Level Optimal Power Flow Solver in Large Distribution Networks

Xinyang Zhou*, Yue Chen*, Zhiyuan Liu†, Changhong Zhao‡, and Lijun Chen†

*Power System Engineering Center, National Renewable Energy Laboratory, Golden, USA

†College of Engineering and Applied Science, University of Colorado, Boulder, USA

‡Department of Information Engineering, the Chinese University of Hong Kong, HK

Emails: {*xinyang.zhou, yue.chen*}*@nrel.gov*, {*zhiyuan.liu, lijun.chen*}*@colorado.edu*, *chzhao@ie.cuhk.edu.hk*

Abstract— Solving optimal power flow (OPF) problems for large distribution networks incurs high computational complexity. We consider a large multi-phase distribution network of tree topology with a deep penetration of active devices. We divide the network into collaborating areas featuring subtree topology and subareas featuring subsubtree topology. We design a multi-level implementation of the primal-dual gradient algorithm to solve the voltage regulation OPF problems while preserving nodal voltage information and topological information within areas and subareas. Numerical results on a 4,521-node system verify that the proposed algorithm can significantly improve the computational speed without compromising any optimality.

I. INTRODUCTION

As the renewable energy generation and smart electricity devices deepen their penetration in power systems, the modem distribution networks are becoming increasingly active. This trend not only suggests that we could exploit more flexibility in dispatchable devices to improve the overall system operation, but also brings challenges to architecting faster and more efficient distribution network operation paradigms that can take advantage of these active devices. Optimal power flow (OPF) is a powerful tool that can simultaneously determine the best operating points for all the dispatchable devices in the system under the constraints of operational limits by solving an optimization problem.

The computational complexity of solving OPF, however, grows considerably when the size of the system increases. For example, the gradient algorithm for solving a voltage regulation problem (e.g., Eqs. (2)) requires computational complexity that is proportional to N^2, N being the node number of the network. This means that, solving OPF for a 4,000-node system may take approximately 1,600 times longer

than solving the same problem for a 100-node system. Other algorithms could experience similar scalability issues when used for large networks. This could lead to a slow response to fast-changing future distribution networks, especially in large systems.

To address this issue, various distributed algorithms have been developed to solve OPF in a more computationally efficient manner. The authors of [2]–[5] use a central controller that is in charge of gathering, processing, and sending network-related information to coordinate distributed devices. The work in [6]–[10] does not need a central controller but require communication and computation capability among all neighbors. Also, a recent comprehensive survey on distributed optimization and control algorithms for electric power systems [11] contains more references.

Another promising line of work to enhance OPF solving efficiency proposes to divide a big network into smaller ones and thus solve smaller parts of the OPF problems in collaboration instead of the original large ones [12]–[15]. The application of such a multi-area solution method in distribution networks, however, has been limited. Recent related works include [16], [17]. This work will continue to explore this method with novel algorithm that improves the computational efficiency without compromising optimality, which will be validated numerically on large systems.

In this work, we focus on a distribution network featuring tree topology, which is known to have fractal properties: any node within a tree together with all its children nodes makes a subtree that also features tree topology and thus inherits the same properties from the original tree; such properties are then passed on to subsubtrees within subtrees, and so on. This observation motivates us to develop a framework to exploit such a fractal pattern to improve the efficiency of solving OPF in large distribution networks.

Following our previous work [1], this work continues to explore the fractal properties of distribution networks featuring tree topology, reveals the corresponding patterns in the sensitivity matrices between voltage magnitudes and nodal power injections in a multi-phases system, and proposes a multi-level solution method to solve optimal voltage regulation problems. By design, the proposed algorithm can significantly improve the efficiency of solving large, convex OPF problems

without losing optimality. Such highly efficient design is crucial for tracking the optimal operating setpoints in real time and for fast recovery from a blackout for large distribution systems with a large number of control nodes. We illustrate the numerical performance of the proposed multi-level algorithms on the primary side of a large distribution system based on the IEEE 8,500-node test feeder and the EPRI CKT7 test feeder. The results show identical convergence dynamics to the same optimal point from centralized algorithm, bi-level implementation, and tri-level implementation, but notably computational speed improvement by using multi-level algorithms.

The rest of this paper is organized as follows. Section II models the distribution system, formulates an optimal voltage regulation problem, and introduces the primal-dual gradient algorithm for solving this OPF problem. Section III divides the large distribution system into subtrees and subsubtrees and proposes an equivalent multi-level implementation of the gradient algorithm. Section IV illustrates the numerical performance of the proposed algorithm, and Section V concludes this paper.

II. SYSTEM MODEL AND OPF SOLVING

A. Network Modeling

We consider a radial multi-phase distribution system denoted by $\mathcal{G} = \{\mathcal{N} \cup \{0\}, \mathcal{E}\}$ with the set $\mathcal{N}$ collecting all n buses excluding the substation bus 0, which is modeled as a slack bus, and the set $\mathcal{E}$ collecting their connecting lines. We denote by $\mathcal{E}_i$ the set of lines describing the unique path from the substation to bus i. Denote by $\mathfrak{i} := \sqrt{-1}$ the unit imaginary number. Let a, b, c denote the three phases, and Φ_i denote the set of phase(s) of bus $i \in \mathcal{N}$. Obviously, $\Phi_i \subseteq \Phi_0 = \{a, b, c\}$ of the substation bus 0 for all $i \in \mathcal{N}$. Define a subset $\mathcal{N}^\phi \subseteq \mathcal{N}$ collecting buses that have phase ϕ. Denote by p_i^ϕ, q_i^ϕ, V_i^ϕ and v_i^ϕ the real power injection, the reactive power injection, the complex voltage phasor, and the squared voltage magnitude, respectively, of phase $\phi \in \Phi_i$ at bus $i \in \mathcal{N}$. We make the following assumptions for a linearized power flow model.

Assumption 1 *The line losses are small and ignored, the magnitudes of the three-phase voltages are approximately equal, and the phase differences among the three-phase voltages are close to $2\pi/3$, i.e., $\frac{V_i^a}{V_i^b} \approx \frac{V_i^b}{V_i^c} \approx \frac{V_i^c}{V_i^a} \approx e^{\mathfrak{i}2\pi/3}$.*

Denote by $z_{\zeta\xi}$ the impedance of line $(\zeta, \xi) \in \mathcal{E}$. $z_{\zeta\xi}$ is a complex number for a single-phase line, a 2×2 complex matrix for a two-phase line, or a 3×3 complex matrix for a three-phase line. Denote by:

$$Z_{ij}^{\varphi\phi} = \sum_{(\zeta,\xi)\in\mathcal{E}_i\cap\mathcal{E}_j} z_{\zeta\xi}^{\varphi\phi} \in \mathbb{C}$$

the summarized impedance (if $\varphi = \phi$) or the summarized mutual impedance (if $\varphi \neq \phi$) of the unique common path of buses i and bus j leading back to bus 0, and $\overline{Z}_{ij}^{\varphi\phi}$ its conjugate. For example, in Fig. 1, the common path of bus 10 and bus 27 leading back to bus 0 is line (1,2) and line (2,4). Define:

$$\boldsymbol{v} := [[v_1^\phi]_{\phi\in\Phi_1}^\top, \dots, [v_n^\phi]_{\phi\in\Phi_N}^\top]^\top \in \mathbb{R}^N$$

$$\boldsymbol{p} := [[p_1^\phi]_{\phi\in\Phi_1}^\top, \dots, [p_n^\phi]_{\phi\in\Phi_N}^\top]^\top \in \mathbb{R}^N$$

$$\boldsymbol{q} := [[q_1^\phi]_{\phi\in\Phi_1}^\top, \dots, [q_n^\phi]_{\phi\in\Phi_N}^\top]^\top \in \mathbb{R}^N$$

the vectors of the squared voltage magnitudes, the real power injection, and the reactive power injection, with $N = \sum_{i\in\mathcal{N}} |\Phi_i|$ adding up all phases at all buses. Here, $|\Phi_i|$ is the cardinality of set Φ_i. To construct a convex and computationally trackable OPF problem, we use the following *linearized power flow equation* [18], [19] based on Assumption 1:

$$\boldsymbol{v} = \mathbf{R}\boldsymbol{p} + \mathbf{X}\boldsymbol{q} + \tilde{\boldsymbol{v}}, \tag{1}$$

with a constant vector $\tilde{\boldsymbol{v}} \in \mathbb{R}^N$ and sensitivity matrices $\mathbf{R}, \mathbf{X} \in \mathbb{R}^{N \times N}$ respectively comprising the elements of:

$$\partial_{p_j^\phi} v_i^\varphi = 2\mathfrak{Re}\{\overline{Z}_{ij}^{\varphi\phi} \cdot \omega^{\varphi-\phi}\} \text{ and}$$

$$\partial_{q_j^\phi} v_i^\varphi = -2\mathfrak{Im}\{\overline{Z}_{ij}^{\varphi\phi} \cdot \omega^{\varphi-\phi}\}.$$

Here, $\omega = e^{-\mathfrak{i}2\pi/3}$, $a, b, c = 0, 1, 2$ when calculating $\varphi - \phi$, and $\mathfrak{Re}\{\cdot\}$ and $\mathfrak{Im}\{\cdot\}$ are the real and imaginary parts of a complex number, respectively.

B. Problem Formulation and Gradient Algorithm

To determine the optimal power setpoints of dispatchable devices while strictly constraining voltages to within specified bounds, we formulate a voltage regulation OPF problem for the multi-phase distribution system as follows:

$$\min_{\boldsymbol{p},\boldsymbol{q}} \quad \sum_{i\in\mathcal{N}} \sum_{\phi\in\Phi_i} C_i^\phi(p_i^\phi, q_i^\phi), \tag{2a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{v}} \leqslant \boldsymbol{v}(\boldsymbol{p},\boldsymbol{q}) \leqslant \overline{\boldsymbol{v}}, \tag{2b}$$

$$(p_i^\phi, q_i^\phi) \in \mathcal{Y}_i^\phi, \phi \in \Phi_i, \forall i \in \mathcal{N}, \tag{2c}$$

where C_i^ϕ is a jointly strongly convex function in both p_i^ϕ and q_i^ϕ for phase $\phi \in \Phi$ of bus i, $\mathcal{Y}_i^\phi$ is its convex and compact feasible set, $\boldsymbol{v}(\boldsymbol{p},\boldsymbol{q})$ represents Eq. (1), and $\underline{\boldsymbol{v}} = \{\underline{v}_i^\phi\}_{i\in\mathcal{N}}^{\phi\in\Phi_i}$ and $\overline{\boldsymbol{v}} = \{\overline{v}_i^\phi\}_{i\in\mathcal{N}}^{\phi\in\Phi_i}$ are the voltage bounds vectors. Associate dual variables $\underline{\boldsymbol{\mu}} = \{\underline{\mu}_i^\phi\}_{i\in\mathcal{N}}^{\phi\in\Phi_i}$ and $\overline{\boldsymbol{\mu}} = \{\overline{\mu}_i^\phi\}_{i\in\mathcal{N}}^{\phi\in\Phi_i}$ with the left and right sides of the constraints (2b), respectively, and we write the regularized Lagrangian of (2) as:

$$\mathcal{L}(\boldsymbol{p},\boldsymbol{q};\overline{\boldsymbol{\mu}},\underline{\boldsymbol{\mu}}) = \sum_{i\in\mathcal{N}} \sum_{\phi\in\Phi_i} C_i^\phi(p_i^\phi, q_i^\phi) + \underline{\boldsymbol{\mu}}^\top(\underline{\boldsymbol{v}} - \boldsymbol{v}(\boldsymbol{p},\boldsymbol{q}))$$
$$+ \overline{\boldsymbol{\mu}}^\top(\boldsymbol{v}(\boldsymbol{p},\boldsymbol{q}) - \overline{\boldsymbol{v}}) - \frac{\eta}{2}\|\boldsymbol{\mu}\|_2^2, \tag{3}$$

where a regularization term $-\frac{\eta}{2}\|\boldsymbol{\mu}\|_2^2$ with a small $\eta > 0$ is added to improve convergence properties. A minor bounded discrepancy between the regularized Lagrangian and the original Lagrangian is introduced because of the regularization term [20].

We implement the primal-dual gradient algorithm for solving the unique saddle point of (3) with a stepsize ϵ as:

$$p_i^\phi(t+1) = \Big[p_i^\phi(t) - \epsilon\Big(\partial_{p_i^\phi} C_i^\phi(p_i^\phi(t), q_i^\phi(t)) +$$
$$\sum_{j\in\mathcal{N}} \sum_{\varphi\in\Phi_j} \partial_{p_i^\phi} v_j^\varphi \big(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\big)\Big)\Big]_{\mathcal{Y}_i^\phi}, \tag{4a}$$

$$q_i^\phi(t+1) \;=\; \Big[q_i^\phi(t) - \epsilon\Big(\partial_{q_i^\phi}C_i^\phi(p_i^\phi(t),q_i^\phi(t)) +$$
$$\sum_{j\in\mathcal{N}}\sum_{\varphi\in\Phi_j}\partial_{q_i^\phi}v_j^\varphi\big(\overline{\mu}_j^\varphi(t)-\underline{\mu}_j^\varphi(t)\big)\Big)\Big]_{\mathcal{Y}_i^\phi}, \quad (4b)$$

$$\underline{\mu}_i^\phi(t+1) = [\underline{\mu}_i^\phi(t) + \epsilon(\underline{v}_i^\phi - v_i^\phi(t) - \eta\underline{\mu}_i^\phi(t))]_+, \quad (4c)$$

$$\overline{\mu}_i^\phi(t+1) = [\overline{\mu}_i^\phi(t) + \epsilon(v_i^\phi(t) - \overline{v}_i^\phi - \eta\overline{\mu}_i^\phi(t))]_+, \quad (4d)$$

$$\boldsymbol{v}(t+1) = \mathbf{R}\boldsymbol{p}(t+1) + \mathbf{X}\boldsymbol{q}(t+1) + \tilde{\boldsymbol{v}}, \quad (4e)$$

where (4a)–(4d) are for all $\phi \in \Phi_i$ and all $i \in \mathcal{N}$. Proof of (4) converging asymptotically to its unique saddle point of Lagrangian (3) given a small enough stepsize ϵ can be found in numerous references (e.g., [1], [21]) and is omitted here.

C. Feedback from Nonlinear Power Flow

What has been presented so far is based on the linearized power flow equation (1) assuming a lossless and balanced three-phase system. As will be shown in Section III, such linearized model is crucial to obtain an important structure in the sensitivity matrices, allowing us to design a hierarchical algorithm. However, unrealistic assumptions like this will inevitably cause modeling errors in practice. For example, when we obtain an optimal solution $(\boldsymbol{p}^*, \boldsymbol{q}^*)$ and the corresponding $\boldsymbol{v}^*$ through dynamics (4), and implement the setpoints in real world featuring nonlinear power flow, the resultant voltage magnitudes may be different than $\boldsymbol{v}^*$, putting the actual system at risk of voltage violation.

To compensate for the linearization errors and to avoid voltage violation in practice, a model-based feedback approach can be used by replacing Eq. (4e) with its nonlinear counterpart so that the decision variables are updated based on voltages from nonlinear power flow model. We refer the detailed performance characterization to [1]. This feedback approach is also applied in the numerical examples in Section IV, where nonlinear power flow is calculated each iteration.

III. MULTI-LEVEL OPF ALGORITHM

A. Motivation

Implementing Eqs. (4) for multiple iterations until convergence calls for a great amount of computation, especially for large distribution networks. Particularly, the computational complexity for

$$\sum_{j\in\mathcal{N}}\sum_{\varphi\in\Phi_j}\partial_{p_i^\phi}v_j^\varphi\big(\overline{\mu}_j^\varphi(t)-\underline{\mu}_j^\varphi(t)\big)\Big]_{\mathcal{Y}_i^\phi} \quad (5a)$$

$$\sum_{j\in\mathcal{N}}\sum_{\varphi\in\Phi_j}\partial_{q_i^\phi}v_j^\varphi\big(\overline{\mu}_j^\varphi(t)-\underline{\mu}_j^\varphi(t)\big)\Big]_{\mathcal{Y}_i^\phi} \quad (5b)$$

in Eqs. (4a)–(4b) for all $\phi \in \Phi_i$ and all $i \in \mathcal{N}$ increases quadratically as the size of the network N increases. Moreover, certain areas within a distribution network may want to preserve local nodal and topological information for security or privacy reasons.

For these reasons, we redesign the implementation of Eqs. (4) for radial distribution networks to significantly increase the computational efficiency and preserve privacy for
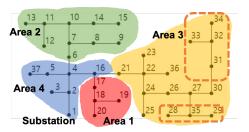

Fig. 1: Illustration of the subtree- and subsubtree-featuring structure based on IEEE 37-bus test feeder. Areas 1–3 ($\mathcal{N}_1$–$\mathcal{N}_3$) have subtree topology with their respective root buses $n_1^0 = 17$, $n_2^0 = 6$, and $n_3^0 = 21$. Area 4 is the remaining area with unclustered nodes ($\mathcal{N}_0$). The circled subareas in Areas 3 are two subsubtrees ($\mathcal{N}_{3_1}$ and $\mathcal{N}_{3_2}$).

areas with needs to protect nodal or topological information. Specifically, when we divide the large system geographically/administratively into areas featuring subtrees, we can take advantage of the network structure as well as the linearization model (1) to simplify the calculation of the computationally heavy coupling parts without compromising any performance based on the collaboration of divided areas. In this way, the computational efficiency is significantly improved while the nodal and topological information within areas are preserved from outside of the area. Next, we apply similar techniques to each area (subtree) to further boost the computational efficiency and preserve privacy within subareas (subsubtrees).

B. Multi-Level Clustering

Based on the pattern of the sensitivity matrix related to the tree/subtree structure in the distribution network, we group all nodes of distribution network $\mathcal{N}$ into two categories: K nonoverlapping subtrees indexed by $k \in \mathcal{K} := \{1,\ldots,K\}$ with $\mathcal{N}_k \in \mathcal{N}$ collecting all nodes within subtree k, and a set $\mathcal{N}_0$ collecting all the other "unclustered" nodes in $\mathcal{N}$. Here, we have $\cup_{k\in\mathcal{K}}\mathcal{N}_k \cup \mathcal{N}_0 = \mathcal{N}$ and $\mathcal{N}_j \cap \mathcal{N}_k = \varnothing, \forall j \neq k$. Denote the root node of subtree $\mathcal{N}_k$ by n_k^0.

Notice that a subtree inherits all the properties from the original tree; therefore, we can divide one subtree into "subsubtrees" for reasons that will become clear soon. We divide area $\mathcal{N}_k$ into non-overlapping subsubtrees indexed by $k_m \in \mathcal{K}_k := \{k_1, k_2, \ldots\}$ and their corresponding bus sets denoted by $\mathcal{N}_{k_m}$ together with $\mathcal{N}_{k_0}$ collecting the remaining unclustered buses.

C. Design Intuition

First, we formally define subtrees within a tree topology.

Definition 1 *A subtree of a tree consists of a node in the tree, all this node's descendants, and the connecting lines among them.*

Because the sensitivity matrices $\mathbf{R}$ and $\mathbf{X}$ are built based on $Z_{ij} = \sum_{(\zeta,\xi)\in\mathcal{E}_i\cap\mathcal{E}_j} z_{\zeta\xi}^{\varphi\phi}$ that depends on the common path of bus i and j to the substation, i.e., $\mathcal{E}_i \cap \mathcal{E}_j$, and because two

non-overlapping subtrees hold the same common path $\mathcal{E}_i \cap \mathcal{E}_j$ for any of their respective buses i and bus j, we conclude that:

$$\partial_{p_j^\phi} v_i^\varphi = \partial_{p_{n_h^0}^\phi} v_{n_k^0}^\varphi = \quad 2\Re\mathfrak{e}\{\overline{Z}_{n_k^0 n_h^0}^{\varphi\phi} \cdot \omega^{\varphi-\phi}\} \tag{6a}$$

$$\partial_{q_j^\phi} v_i^\varphi = \partial_{q_{n_h^0}^\phi} v_{n_k^0}^\varphi = -2\Im\mathfrak{m}\{\overline{Z}_{n_k^0 n_h^0}^{\varphi\phi} \cdot \omega^{\varphi-\phi}\} \tag{6b}$$

hold for any $i \in \mathcal{N}_k$ and any $j \in \mathcal{N}_h$, where $\mathcal{N}_k$ and $\mathcal{N}_h$ respectively denote the sets collecting buses of two non-overlapping subtrees indexed by $k, h \in \mathcal{K}$ with their respective root buses indexed by n_k^0 and n_h^0, with set $\mathcal{K}$ collecting all subtree indexes. See Fig. 2 for an illustration, where the white dashed line is the common path of any node in Area 1 and any node in Area 3—marked by blue triangles—to the substation. Because subtrees inherit all features from the original tree, we can apply similar results to them.

D. Methods

1) Bi-Level Implementation: Based on such intuition, we can equivalently rewrite the last term of (4a) for clustered node $i \in \mathcal{N}_k$ with phase $\phi \in \Phi_i$ as the following three parts:

$$\sum_{j\in\mathcal{N}}\sum_{\varphi\in\Phi_j} \partial_{p_i^\phi} v_j^\varphi \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\Big]_{\mathcal{Y}_i^\phi} = \tag{7a}$$

$$2\Re\mathfrak{e}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_k} \overline{Z}_{ji}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} + \tag{7b}$$

$$2\Re\mathfrak{e}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{\substack{n_h^0\in N^\varphi \\ h\in\mathcal{K}, h\neq k}} \overline{Z}_{n_h^0 n_k^0}^{\varphi\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_h} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} + \tag{7c}$$

$$2\Re\mathfrak{e}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_0} \overline{Z}_{jn_k^0}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\}, \tag{7d}$$

where (7b), (7c), and (7d), are the components of (7a) from nodes within subtree $\mathcal{N}_k$, nodes within all other subtrees $\mathcal{N}_h, h \neq k$, and nodes not in any subtree $\mathcal{N}_0$, respectively. Computation-wise, (7b) and (7d) are same as before; however, (7c) significantly simplifies the original computation by reducing a large number of repetitive computations by applying the result in (6a), and (7d) captures the influence from buses outside any divided subtree area (e.g., Area 5 in Fig. 2) whose buses are collected in set $\mathcal{N}_0$ for completeness.

The last term in (4b) can be processed similarly as:

$$\sum_{j\in\mathcal{N}}\sum_{\varphi\in\Phi_j} \partial_{q_i^\phi} v_j^\varphi \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\Big]_{\mathcal{Y}_i^\phi} = \tag{8a}$$

$$-2\Im\mathfrak{m}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_k} \overline{Z}_{ji}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} \tag{8b}$$

$$-2\Im\mathfrak{m}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{\substack{n_h^0\in N^\varphi \\ h\in\mathcal{K}, h\neq k}} \overline{Z}_{n_h^0 n_k^0}^{\varphi\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_h} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} \tag{8c}$$

$$-2\Im\mathfrak{m}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_0} \overline{Z}_{jn_k^0}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\}. \tag{8d}$$

2) Tri-Level Implementation: Consider (7b) and notice that it has a form similar to the original coupling term. Hence, the computational complexity of $\mathcal{O}(|\mathcal{N}_k|^2)$ for (7b) can be high for large $|\mathcal{N}_k|$. Fortunately, we can apply similar tricks to divide area $\mathcal{N}_k$ into subsubtrees indexed by $k_m \in \mathcal{K}_k = \{k_1, k_2, \ldots\}$ and their corresponding buses sets denoted by $\mathcal{N}_{k_m}$ together with $\mathcal{N}_{k_0}$ collecting the remaining buses. We then equivalently rewrite (7b) for $i \in \mathcal{N}_{k_m} \subset \mathcal{N}_k, m \in \mathcal{K}_k, k \in \mathcal{K}$ as:

$$2\Re\mathfrak{e}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_k} \overline{Z}_{ji}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} = \tag{9a}$$

$$2\Re\mathfrak{e}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_{k_m}} \overline{Z}_{ji}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} + \tag{9b}$$

$$2\Re\mathfrak{e}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{\substack{n_{k_m'}^0\in N^\varphi \\ k_m'\in\mathcal{K}_k, m'\neq m}} \overline{Z}_{n_{k_m'}^0 n_{k_m}^0}^{\varphi\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_{k_m'}} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} + \tag{9c}$$

$$2\Re\mathfrak{e}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_{k_0}} \overline{Z}_{jn_{k_m}^0}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\}, \tag{9d}$$

where $n_{k_m}^0$ denotes the root bus of subsubtree k_m. Now (9c) further simplifies the computation for (7b) similarly as (7c) does for (4a). Note that the nodal information within subsubtrees is aggregated without being exposed.

Then Eq. (7b) for $i \in \mathcal{N}_{k_m} \subset \mathcal{N}_k, m \in \mathcal{K}_k, k \in \mathcal{K}$ can be similarly calculated as:

$$-2\Im\mathfrak{m}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_k} \overline{Z}_{ji}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} = \tag{10a}$$

$$-2\Im\mathfrak{m}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_{k_m}} \overline{Z}_{ji}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} \tag{10b}$$

$$-2\Im\mathfrak{m}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{\substack{n_{k_m'}^0\in N^\varphi \\ k_m'\in\mathcal{K}_k, m'\neq m}} \overline{Z}_{n_{k_m'}^0 n_{k_m}^0}^{\varphi\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_{k_m'}} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\} \tag{10c}$$

$$-2\Im\mathfrak{m}\left\{ \sum_{\varphi\in\Phi_0} \omega^{\varphi-\phi} \sum_{j\in\mathcal{N}^\varphi\cap\mathcal{N}_{k_0}} \overline{Z}_{jn_{k_m}^0}^{\varphi\phi} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t)\right)\right\}. \tag{10d}$$

Eqs. (7)–(10) provide a more efficient way to carry out the most computationally heavy terms in Eqs. (4a)–(4b). While the proposed method has reduced the computational complexity, its convergence dynamics and equilibrium point stay the same because the new method is mathematically equivalent to the original primal-dual gradient algorithm. Such design gives us "free" acceleration without losing performance, even if it is not implemented in a parallel manner among areas and subareas, as will be shown in Section IV. Furthermore, considering the fractal properties of tree/subtree—i.e., subtrees have tree structures and thus can be further divided into subsubtrees where everything still holds—we can theoretically apply such layering structure as deep as we wish, if it is needed and if the network allows.

E. Privacy Preservation

In the original primal-dual gradient algorithm (4), a central controller needs to know the dual variables for all buses and the topology of the entire network to compute the coupling

One-Level Algorithm	Device #	1043												
	Time (s)	11,588.72												
Bi-Level Algorithm	Areas	Area 1				Area 2				Area 3				Area 4	Total
	Device #	357				222				310				154	1043
	Time (s)	1,050.16				501.21				834.90				256.81	2,717.25
Tri-Level Algorithm	Subareas	1_1	1_2	1_3	1_0	2_1	2_2	2_3	2_0	3_1	3_2	3_3	3_0	4_0	Total
	Device #	49	74	23	211	70	39	17	96	68	66	68	108	154	1043
	Time (s)	713.08				337.04				490.51				261.07	1,890.40

TABLE I: Detailed clustering information and computational time of all areas and subareas over 3,000 iterations for the centralized algorithm, the bi-level algorithm, and the tri-level algorithm.

parts (5); therefore, every bus must share its nodal voltage or local dual variables with the central controller every iteration, together with its topological information. Meanwhile, when computing (5) through Eqs. (7)–(8), all nodal and topological information can be preserved within each area, and only aggregated area information needs to be shared with the central controller. For example, for all buses within area k, they only need to share their aggregated dual variables $\sum_{j \in \mathcal{N}^\varphi \cap \mathcal{N}_k} \left(\overline{\mu}_j^\varphi(t) - \underline{\mu}_j^\varphi(t) \right)$ for all phases φ and the topological information related to its root buses n_k^0 to outside of area k. For buses within subareas, detailed information of subareas is similarly preserved within.
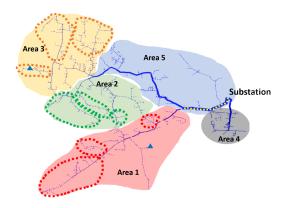

Fig. 2: Areas and subareas (within dashed lines) of the 4,521-node network used in Section IV. Areas 1–3 have subtree topology, and all the subareas feature subsubtree topology.

IV. NUMERICAL RESULTS

A. Simulation Setup

We construct a three-phase, unbalanced, 11,000-node test feeder by connecting an IEEE 8,500-node test feeder and an EPRI Ckt7 test feeder at the substation. We simplify the network by lumping the loads on the secondary side into corresponding distribution transformers, resulting in a 4,521-node network ($N = 4,521$) with a total of 1,043 controllable devices. We group all the controllable nodes into four subtree areas, and we divide Areas 1–3 further into subareas featuring subsubtrees and remaining buses; see Fig. 2 and Table I for the details.

For each controllable device at bus i of phase ϕ, we consider minimizing the cost of its deviation from its initial (the most preferred) level $(p_i^\phi(0), q_i^\phi(0))$, i.e., $C_i^\phi(p_i^\phi, q_i^\phi) = (p_i^\phi - p_i^\phi(0))^2 + (q_i^\phi - q_i^\phi(0))^2$. We set voltage bounds uniformly to 0.95 p.u. and 1.05 p.u. as the hard constraints. We set the same stepsizes for all three implementations as 3.5×10^{-4} for the primal updates and 3.5×10^{-3} for the dual updates. We disabled all voltage regulators so that they do not assist OPF with voltage regulation. As a result, a largely undervoltage scenario is observed, as plotted with blue dots in Fig. 4. We implement the primal-dual gradient algorithm to solve for the unique saddle point of the regularized Lagrangian (3) based on the centralized implementation, the bi-level implementation, and the tri-level implementation. OpenDSS is used to generate the nonlinear power flow at each iteration to replace the linear power flow in Eqs. (4).

The simulation runs with Python 3.6 on Windows 10 Enterprise Version on a laptop with Intel Core i7-7600U CPU @ 2.80GHz 2.90GHz, 8.00GB RAM.

B. Simulation Results

1) Dynamic Performance: As shown in Fig. 3, when set with the identical initial conditions and stepsizes, the centralized implementation, i.e., the straightforward implementation of Eqs. (4), the bi-level and the tri-level implementations of Eqs. (4) share exactly the same convergence dynamics and the minimum cost value at convergence, illustrating the mathematical equivalence among the three implementation.

2) Speed Improvement: Because of the computational complexity reduction achieved by our design, however, each iteration costs less time in the bi-level or the tri-level implementation than the centralized one. We record the time used to run 3,000 iterations for the three implementations in TABLE I. As shown, the time used for the centralized algorithm is 4.26 times of the time for the bi-level algorithm, which is 44% more than the time used for the tri-lvel algorithm. Note that Area 4 serving as a control group has a similar computational time for bi-level and tri-level algorithms because we do not divide it into subareas.

Such computational complexity reduction without compromising optimality gives us "free" speed improvement. Moreover, the multi-level structure also allows for more parallel and autonomous implementation among areas and subareas and potentially further speeds up the convergence.
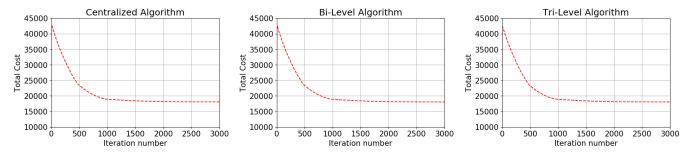
Fig. 3: The centralized algorithm (one-level algorithm), the bi-level algorithm, and the tri-level algorithms, when set to the same initialization and stepsize, exhibit identical convergence dynamics over 3,000 iterations to the same minimum cost. This is expected because the three implementations are mathematically identical.
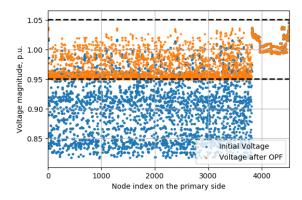

Fig. 4: The largely undervoltage system has been improved after OPF.

3) Voltage Regulation: As a result, we see that the voltage profiles are regulated to within the prescribed upper and lower bounds after the OPF is solved. Note that quite a few nodes have their voltage on the lower bound 0.95 p.u. This is because intuitively the most optimal solutions should satisfy the hard voltage constraints by incurring the minimal power setpoint deviation from the initial values for devices.

V. CONCLUSION

By exploring the fractal properties of power distribution networks of tree topology, as well as the structure of the impedance matrix, this paper extends our previous bi-level distributed OPF solver based on the primal-dual gradient algorithm to multi-level implementation. Privacy, including nodal voltages and network topology, has been preserved within areas and subareas. Numerical results on a 4,521-node large multi-phase distribution network show significant computational speed improvement of such an extension without compromising optimality.

REFERENCES

[1] X. Zhou, Z. Liu, C. Zhao, and L. Chen, "Accelerated voltage regulation in multi-phase distribution networks based on hierarchical distributed algorithm," *IEEE Trans. Power Systems*, 2019.

[2] E. Dall'Anese and A. Simonetto, "Optimal power flow pursuit," *IEEE Trans. on Smart Grid*, vol. 9, no. 2, pp. 942–952, 2018.

[3] Y. Tang, K. Dvijotham, and S. Low, "Real-time optimal power flow," *IEEE Trans. on Smart Grid*, vol. 8, no. 6, pp. 2963–2973, 2017.

[4] A. Hauswirth, A. Zanardi, S. Bolognani, F. Dörfler, and G. Hug, "Online optimization in closed loop on the power flow manifold," pp. 1–6, 2017.

[5] X. Zhou, E. Dall'Anese, and L. Chen, "Online stochastic optimization of networked distributed energy resources," *IEEE Trans. on Automatic Control*, vol. 65, no. 6, pp. 2387–2401, 2019.

[6] Q. Peng and S. H. Low, "Distributed optimal power flow algorithm for radial networks, i: Balanced single phase case," *IEEE Trans. on Smart Grid*, vol. 9, no. 1, pp. 111–121, 2016.

[7] P. Šulc, S. Backhaus, and M. Chertkov, "Optimal distributed control of reactive power via the alternating direction method of multipliers," *IEEE Transactions on Energy Conversion*, vol. 29, no. 4, pp. 968–977, 2014.

[8] S. Magnússon, G. Qu, C. Fischione, and N. Li, "Voltage control using limited communication," *IEEE Trans. on Control of Network Systems*, vol. 6, no. 3, pp. 993–1003, 2019.

[9] C. Wu, G. Hug, and S. Kar, "Smart inverter for voltage regulation: Physical and market implementation," *IEEE Trans. on Power Systems*, vol. 33, no. 6, pp. 6181–6192, 2018.

[10] M. Kraning, E. Chu, J. Lavaei, S. P. Boyd *et al.*, *Dynamic network energy management via proximal message passing*. Now Publishers, 2014.

[11] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Trans. on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[12] B. Kroposki, E. Dall'Anese, A. Bernstein, Y. Zhang, and B.-M. Hodge, "Autonomous energy grids," *Proc. of Hawaii International Conference on System Sciences*, 2018.

[13] A. J. Conejo and J. A. Aguado, "Multi-area coordinated decentralized dc optimal power flow," *IEEE Trans. on Power Systems*, vol. 13, no. 4, pp. 1272–1278, 1998.

[14] F. J. Nogales, F. J. Prieto, and A. J. Conejo, "A decomposition methodology applied to the multi-area optimal power flow problem," *Annals of operations research*, vol. 120, no. 1-4, pp. 99–116, 2003.

[15] X. Lai, L. Xie, Q. Xia, H. Zhong, and C. Kang, "Decentralized multi-area economic dispatch via dynamic multiplier-based lagrangian relaxation," *IEEE Trans. on Power Systems*, vol. 30, no. 6, pp. 3225–3233, 2014.

[16] K. Zhang, W. Shi, H. Zhu, E. Dall'Anese, and T. Basar, "Dynamic power distribution system management with a locally connected communication network," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 673–687, 2018.

[17] C.-Y. Chang, M. Colombino, J. Cortés, and E. Dall'Anese, "Saddle-flow dynamics for distributed feedback-based optimization," *IEEE Control Systems Letters*, 2019.

[18] L. Gan and S. H. Low, "Convex relaxations and linear approximation for optimal power flow in multiphase radial networks," *Proc. of Power Systems Computation Conference (PSCC)*, pp. 1–9, 2014.

[19] ——, "An online gradient algorithm for optimal power flow on radial networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 625–638, 2016.

[20] J. Koshal, A. Nedić, and U. V. Shanbhag, "Multiuser optimization: Distributed algorithms and error analysis," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 1046–1081, 2011.

[21] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.