A Scheduled-Asynchronous Distributed Optimization Algorithm for the Optimal Power Flow Problem

Chin-Yao Chang Jorge Cortés Sonia Martínez

***Abstract*— Optimal power flow (OPF) problems are non-convex and large-scale optimization problems. Finding an optimal solution for the OPF problem in real time is challenging and important in various applications. Recent studies show that a wide class of OPF problems have an exact semidefinite programming (SDP) convex relaxation. However, only few works have considered distributed algorithms to solve these. In this paper, we propose a scheduled-asynchronous algorithm with this objective. The proposed algorithm follows an ADMM-like iteration for every edge in the electrical network and is asynchronous in the sense that the agents do not simultaneously update their local variables, but only do so when they have received fresh information from all of their neighbors. In addition, if the electrical network topology is bipartite, the proposed algorithm has a convergence rate of $O(1/n)$, where n is the iteration per agent. The asynchronous property and fast convergence rate make the proposed algorithm suitable for the OPF problem. Simulation studies demonstrate that the proposed algorithm is scalable with the number of buses and robust to network effects including delays and packet drops.**

I. Introduction

The OPF problem is known to be a non-convex optimization problem that minimizes electricity generation cost subject to voltage and power flow constraints. Due to its large scale and non-convex nature, it is challenging to solve this problem in general, and, therefore, solutions are typically found off-line for centralized planning. However, the integration of renewable energy resources into distribution networks results into a larger operational uncertainty. In such cases, networks with better robustness and tolerance to intermittency are needed, for which the real-time solution to the OPF problems would be beneficial. Motivated by the increasing need of real time OPF solvers, in this paper we develop a distributed algorithm to solve OPF problems.

Literature review: Finding a global optimum for the OPF problem is challenging, so most existing algorithms only guarantee a local optimum, see e.g. [1]–[3]. The recent work [4] shows that the SDP convex relaxation on the OPF problems is exact for many OPF examples and a global optimal solution can be obtained. Conditions on the exact convex relaxation have been further established in [5], [6]. For cases such that SDP does not provide a feasible solution, a near global optimal solution can still be obtained [7], [8]. Though centralized SDP algorithms have been well studied, only few works have considered distributed algorithms to solve the SDP problem associated with the OPF problem, which may lead to the near real-time global optima. The work [9] proposes gradient-based primal and dual algorithm method for this purpose. The algorithm exhibits fast convergence in the presented simulation studies and its complexity only grows linearly with respect to the problem size. The drawback of the algorithm is that only simple voltage magnitude constraints and linear objective functions are considered. How the constraints on the active/reactive power or nonlinear objective functions affect the algorithm is unclear. In [10], the network is first divided into several regions and the alternating direction method of multipliers (ADMM) is used, thereby relying on synchronization between the agents of the regions.

The OPF problem can be seen as a separable optimization problem because every constraint and objective function only relates to the voltage at one bus and its neighboring buses [9], [11]. Plenty of distributed algorithms are available for separable optimization, but they may not be a good fit for distributed OPF. ADMM has a convergence rate $O(1/n)$ with no assumption on strong convexity of the cost function [12], while the implementation requires clock synchronization between the agents. Gradient-based methods are amenable to asynchronous implementation [13], but many of those have a rate of convergence $O(\log(n)/n)$ or slower [14]. This highlights the need for a distributed algorithm with combined fast convergence and asynchronous properties.

Statement of contributions: We develop a scheduled-asynchronous algorithm to solve the OPF problem in a distributed way. The OPF problem is first formulated as a distributed optimization problem with every bus (node) being an agent in the network. Every agent has a cost function and local constraints with additional equality constraints on neighbors' variables to ensure that every copy of the variables coincide. We apply the SDP convex relaxation on the distributed OPF problem and assume that the conditions for the exact convex relaxation hold. We propose that every pair of connected buses do alternate minimizations in such a way that any two buses sub-network follows an ADMM-like iteration. In this way, every bus only performs a local

C.-Y. Chang, Jorge Cortés, and Sonia Martínez are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA, USA. Email: {chc433, cortes, soniamd}@ucsd.edu

minimization when all neighboring buses have updated their variables after the bus's previous minimization. The algorithm does not require clock synchronization between agents, while it requires subtle ordering of iterations. Under mild conditions, the proposed algorithm converges for general SDP-relaxed OPF problems. Furthermore, if the network is bipartite, the algorithm reduces to ADMM and has a $O(1/n)$ convergence rate. One important special case of a bipartite graph is the tree-network topology. Considering that distribution networks typically have tree-network topology, the proposed algorithm is especially suitable for the OPF problem in distribution networks. For reasons of space, proofs are omitted and will appear elsewhere.

Organization: Section II introduces the notations used in this paper. In Section III, a standard OPF problem and its distributed formulation are introduced. In Section IV, the scheduled-asynchronous algorithm is developed. We then demonstrate the effectiveness and the robustness of the proposed algorithm through the simulations in Section V. Some concluding remarks are made in Section VI.

II. Preliminaries

This section introduces some notations and graph-theoretic concepts that will be used in the paper.

A. Notation

We denote the sets of real and complex numbers by $\mathbb{R}$ and $\mathbb{C}$, respectively. Let $\mathbb{S}_+$ and $\mathcal{H}^n$ be the set of positive semidefinite matrices and n-dimensional Hermitian matrices, respectively. The cardinality of a set $\mathcal{N}$ is denoted as $|\mathcal{N}|$. For a complex number $a \in \mathbb{C}$, we let $|a|$ and $\angle a$ denote the complex modulus and angle of a. The 2-norm of a complex vector $v \in \mathbb{C}^n$ is denoted as $\|v\|$. For a complex matrix $A \in \mathbb{C}^{n\times n}$, let A^* be the conjugate transpose. Denote $\mathbf{Tr}\{A\}$ as the trace of A.

B. Graph Theory

We review basic concepts from graph theory following [15]. We denote a graph as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} \subseteq \mathbb{N}$ is the set of the vertices and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges. In undirected graphs, a pair of nodes is undirected, that is, $\{i, k\} = \{k, i\} \in \mathcal{E}$. The local neighborhood of a node k in an undirected graph is $\mathcal{N}_k := \{l \in \mathcal{N} \mid \{l, k\} \in \mathcal{E}\} \cup \{k\}$. In directed graphs, a pair of vertices $\{i, k\}$ in $\mathcal{E}$ is ordered, in such a way that, given $\{i, k\}$ (or $i \to k$,) i is distinguished as a *tail* and k distinguished as the *head* node, respectively. A path in an undirected graph is a sequence of edges which connects a sequence of vertices. A directed path is a sequence of edges connecting a sequence of vertices with an additional restriction that the edges are all oriented in the same direction. A directed cycle is a directed path (with at least one edge) whose first and last vertices are the same. An orientation of an undirected graph is an assignment of exactly one direction to each of the edges. The orientation is *acyclic* if the assignment does not form any directed cycle.

III. Problem Formulation

This section introduces our problem of interest. We begin with a general matrix formulation of the optimal power flow (OPF) problem over an electrical network. We then follow the exposition in [11] to rewrite the OPF problem as a combination of several smaller-scale interconnected subproblems, which are further convexified by dropping the rank constraints. The resulting SDP problem establishes the foundation for our distributed algorithm design.

Consider an electrical network with generation $\mathcal{N}_G$ buses, and edge set $\mathcal{E}$. Let $\mathcal{N} = \mathcal{N}_G \cup \mathcal{N}_L$ and denote its cardinality by N. We denote the phasor voltage at bus i by $V_i = E_i e^{j\theta_i}$, where $E_i \in \mathbb{R}$ and $\theta_i \in [-\pi, \pi)$ are the voltage magnitude and phase angle, respectively. The active and reactive power injections at bus k are given by the power flow equations [16]

$$\begin{aligned} P_k &= \mathbf{Tr}\{Y_k V V^*\} + P_{D_k}, \\ Q_k &= \mathbf{Tr}\{\overline{Y}_k V V^*\} + Q_{D_k}, \end{aligned}$$

where $V \in \mathbb{C}^N$ is the collection of voltage of all buses, $Y_k, \overline{Y}_k \in \mathcal{H}^N$ are derived from the admittance matrix of the electrical network, and $P_{D_k}, Q_{D_k} \in \mathbb{R}$ are the active and reactive power demands at bus k.

The OPF problem involves the following box constraints

$$\begin{aligned} &\underline{V}_k^2 \le |V_k|^2 \le \overline{V}_k^2, \ \forall k \in \mathcal{N} \\ &\underline{P}_k \le P_k \le \overline{P}_k, \ \ \underline{Q}_k \le Q_k \le \overline{Q}_k, \\ &|V_i - V_k|^2 \le \overline{V}_{ik}, \ \forall \{i, k\} \in \mathcal{E}, \end{aligned} \tag{1}$$

where $\overline{V}_{ik}$ is the upper bound of the voltage difference between buses i, k, $\underline{V}_k$ and $\overline{V}_k$ are the lower and upper bounds of the voltage magnitude respectively. All $\underline{P}_k, \underline{Q}_k, \overline{P}_k, \overline{Q}_k$ are defined similarly. The objective function for the OPF problem is given as a quadratic function of the active power injections as follows

$$\sum_{k\in\mathcal{N}_G} c_{k2} P_k^2 + c_{k1} P_k, \tag{2}$$

where $c_{k2} > 0$, $c_{k1} \in \mathbb{R}$. Define the decision variable as $W = VV^*$. The OPF problem is formulated in the following

(P1) $$\min_{a, W} \sum_{k\in\mathcal{N}_G} a_k,$$

subject to

$$\begin{bmatrix} c_{k1}(\mathbf{Tr}\{Y_k W\} + P_{D_k}) - a_k & \star \\ \sqrt{c_{k2}}(\mathbf{Tr}\{Y_k W\} + P_{D_k}) & -1 \end{bmatrix} \preceq 0,\ \forall k \in \mathcal{N}_G, \tag{3a}$$

$$\underline{P}_k \le \mathbf{Tr}\{Y_k W\} + P_{D_k} \le \overline{P}_k,\ \forall k \in \mathcal{N}, \tag{3b}$$

$$\underline{Q}_k \le \mathbf{Tr}\{\overline{Y}_k W\} + Q_{D_k} \le \overline{Q}_k, \tag{3c}$$

$$\underline{V}_k^2 \le \mathbf{Tr}\{M_k W\} \le \overline{V}_k^2, \tag{3d}$$

$$\mathbf{Tr}\{M_{ik} W\} \le \overline{V}_{ik},\ \forall \{i,k\} \in \mathcal{E}, \tag{3e}$$

$$W \succeq 0, \quad \text{rank}(W) = 1, \tag{3f}$$

where $\star$ denotes the complex conjugate of the off-diagonal elements, $a \in \mathbb{R}^{|\mathcal{N}_G|}$, $M_k, M_{ik} \in \mathcal{H}^N$ are defined so that $\mathbf{Tr}\{M_k W\} = |V_k|^2$ and $\mathbf{Tr}\{M_{ik} W\} = |V_i - V_k|^2$. Note that Eq. (3a) together with the introduction of the new variable a allows us to write a linear objective function and move the quadratic dependence to the constraints. Constraints (3b-3e) come from Eq. (1). The combined constraints $W \succeq 0$ and $\text{rank}(W) = 1$ in (3f) correspond to writing the voltage as a matrix variable.

The OPF problem above can be reformulated in a distributed way as follows. We start from the observation that every constraint except Eq. (3f) in **(P1)** is either related to the power injection at one bus or the voltage difference between connected buses. In addition, the power injection at one bus is only related to the voltage of the buses it is connected to. The property is embedded in the non-zero entries of the admittance matrix. We can therefore rewrite constraints (3a-3e) by defining new variables $W_k \in \mathcal{H}^{|\mathcal{N}_k|}$ for all $k \in \mathcal{N}$, where $W_k := \hat{V}_k \hat{V}_k^*$ and $\hat{V}_k \in \mathbb{C}^{|\mathcal{N}_k|}$ is the collection of voltages of bus $i \in \mathcal{N}_k$. In this way, we obtain

$$\begin{bmatrix} c_{k1}(\mathbf{Tr}\{Y_{k,r} W_k\} + P_{D_k}) - a_k & \star \\ \sqrt{c_{k2}}(\mathbf{Tr}\{Y_{k,r} W_k\} + P_{D_k}) & -1 \end{bmatrix} \preceq 0,\ \forall k \in \mathcal{N}_G, \tag{4a}$$

$$\underline{P}_k \le \mathbf{Tr}\{Y_{k,r} W_k\} + P_{D_k} \le \overline{P}_k,\ \forall k \in \mathcal{N}, \tag{4b}$$

$$\underline{Q}_k \le \mathbf{Tr}\{\overline{Y}_{k,r} W_k\} + Q_{D_k} \le \overline{Q}_k, \tag{4c}$$

$$\underline{V}_k^2 \le \mathbf{Tr}\{M_{k,r} W_k\} \le \overline{V}_k^2, \tag{4d}$$

$$\mathbf{Tr}\{M_{ik,r} W_k\} \le \overline{V}_{ik},\ \forall \{i,k\} \in \mathcal{E}. \tag{4e}$$

Here, $M_{k,r}$ is the principal submatrix of M_k obtained by dropping the rows and columns associated with the buses in $\mathcal{N} \setminus \mathcal{N}_k$. The matrices $M_{ik,r}$, $Y_{k,r}$, $\overline{Y}_{k,r}$ are defined similarly.

Instead of solving **(P1)**, we view W_i, $i \in \mathcal{N}$, as a new state variable and consider the following distributed convex optimization problem

$$\textbf{(P2)} \qquad \min_{a, W_1, W_2, \ldots, W_N} \sum_{k \in \mathcal{N}_G} a_k, \tag{5}$$

subject to

$$\begin{aligned} &\text{Eq. (4) holds}, \quad W_k \succeq 0, \quad \forall k \in \mathcal{N}, \\ &W_k(\hat{k}, \hat{k}) = W_i(\hat{k}, \hat{k}),\ \ \forall \{i,k\} \in \mathcal{E}, \\ &W_k(\hat{i}, \hat{k}) = W_i(\hat{i}, \hat{k}), \end{aligned}$$

where $\hat{k}$ denotes the row (or column) of W_i associated with bus k. Notice that if the additional non-convex rank constraints, $\text{rank}(W_i^{opt}) = 1$, $\forall i \in \mathcal{N}$, are imposed on **(P2)**, **(P2)** is equivalent to **(P1)** [9], [11]. Conditions for **(P2)** to have a rank one optimum can be found in [11]. We will assume the conditions in the reference hold so that a solution to **(P2)** is a solution **(P1)** as well. In the rest of the paper, every bus is considered as an agent and the bidirectional communication between connected buses are assumed. We consider distributed algorithms such that every agent interchanges information with its neighbors about its local variables to solve **(P2)**. The focus of this manuscript is on developing the distributed algorithm.

IV. Scheduled-Asynchronous Distributed Algorithm

In this section, we propose a scheduled-asynchronous block-wise distributed optimization algorithm to tackle problem **(P2)**. For convenience of exposition, we rewrite this optimization problem in the following form

$$\begin{aligned} &\min_{X_i \in \mathcal{X}_i, i=1,\ldots,N} \sum_{i \in \mathcal{N}} f_i(X_i) \\ &\text{s.t. } G_{ik}(X_i, X_k) = 0,\ \forall \{i,k\} \in \mathcal{E}, \end{aligned} \tag{6}$$

where $X_i = \{a_i, W_i\}$, $\mathcal{X}_i$ is the constraint set of X_i including (4) and the constraint $W_i \succeq 0$, and G_{ik} is a compact-form, functional representation of the linear constraints $W_k(\hat{k}, \hat{k}) = W_i(\hat{k}, \hat{k})$, $W_k(\hat{i}, \hat{k}) = W_i(\hat{i}, \hat{k})$, given as

$$G_{ik}(X_i, X_k) = \begin{bmatrix} \mathbf{Tr}\{B_{1,ki} W_k - B_{2,ik} W_i\} \\ \mathbf{Tr}\{B_{2,ki} W_k - B_{1,ik} W_i\} \\ \mathbf{Tr}\{B_{3,ki} W_k - B_{4,ki} W_i\} \\ \mathbf{Tr}\{B_{4,ki} W_k - B_{3,ki} W_i\} \end{bmatrix}, \text{ where}$$

$$B_{1,ki}(l,m) = \begin{cases} 1, & \text{if } l = m = \hat{k}, \\ 0, & \text{otherwise}, \end{cases}$$

$$B_{2,ik}(l,m) = \begin{cases} 1, & \text{if } l = m = \hat{k}, \\ 0, & \text{otherwise} \end{cases}$$

$$B_{3,ki}(l,m) = \begin{cases} 1, & \text{if } (l,m) = (\hat{k}, \hat{i}) \text{ or } (l,m) = (\hat{i}, \hat{k}), \\ 0, & \text{otherwise} \end{cases}$$

$$B_{4,ki}(l,m) = \begin{cases} -j, & \text{if } (l,m) = (\hat{k}, \hat{i}), \\ j, & \text{if } (l,m) = (\hat{i}, \hat{k}), \\ 0, & \text{otherwise} \end{cases}$$

A. Design Rationale

To motivate our algorithm design, we start by considering the optimization in **(P2)** on a two-bus network, $N = 2$. In this case, the problem exactly corresponds to the standard form for the ADMM algorithm [17],

$$\min_{X_i \in \mathcal{X}_i} f_1(X_1) + f_2(X_2)$$
$$\text{s.t. } G_{12}(X_1, X_2) = 0.$$

An iteration of ADMM is given as

$$X_1^{t^+} = \text{argmin}_{X_1 \in \mathcal{X}_1} f_1(X_1) \tag{7a}$$
$$+ (p_{12}^t)^\top (G_{12}(X_1, X_2^t)) + \frac{\rho}{2}\|G_{12}(X_1, X_2^t)\|^2,$$
$$X_2^{t^+} = \text{argmin}_{X_2 \in \mathcal{X}_2} f_2(X_2) \tag{7b}$$
$$+ (p_{12}^t)^\top (G_{12}(X_1^{t^+}, X_2)) + \frac{\rho}{2}\|G_{12}(X_1^{t^+}, X_2)\|^2,$$
$$p_{12}^{t^+} = p_{12}^t + \rho G_{12}(X_1^{t^+}, X_2^{t^+}), \tag{7c}$$

where the superscript t is the time at which the update occurs, t^+ is the time for the next round of the optimization, ρ is a given constant, and $p_{12}^t \in \mathbb{R}^4$ is the collection of Lagrange multipliers associated with edge (line) $\{1, 2\}$.

For general networks, $N > 2$, the optimization can be understood as a combination of a number of two-bus sub-problems. This viewpoint inspires us to propose the following design: whenever bus i receives the updated X_k^t from all its neighboring nodes $k \in \mathcal{N}_i$, it solves the following optimization

$$X_i^{t^+} = \text{argmin}_{X_i \in \mathcal{X}_i} f_i(X_i) \tag{8}$$
$$+ \sum_{\{i,k\} \in \hat{\mathcal{E}}} \Big((p_{ik}^t)^\top G_{ik}(X_i, X_k^t) + \frac{\rho_{ik}}{2}\|G_{ik}(X_i, X_k^t)\|^2 \Big)$$
$$+ \sum_{\{k,i\} \in \hat{\mathcal{E}}} \Big((p_{ik}^t)^\top G_{ik}(X_k^{t^+}, X_i) + \frac{\rho_{ik}}{2}\|G_{ik}(X_k^{t^+}, X_i)\|^2 \Big),$$

where $\hat{\mathcal{E}}$ is an orientation of $\mathcal{E}$ that defines the ordering of optimization of terminal nodes for every edge. The terminal nodes of each edge take turns in performing the optimization in (8). Notice that, under the proposed design, the difference in the number of iterations made between two connected nodes is at most one due to the alternating execution (the node that does (7a) goes first and hence may have executed one more iteration than the other node at any given time). Such ordering is encoded by the orientated graph $\hat{\mathcal{G}} = (\mathcal{N}, \hat{\mathcal{E}})$. The tail node of each edge does the first step (7a) and the head node does the second step (7b). Each node i updates the Lagrange multiplier according to

$$\text{Head node: } p_{ik}^{t^+} = p_{ik}^t + \rho_{ik} G_{ik}(X_k^{t^+}, X_i^{t^+}), \tag{9a}$$
$$\text{Tail node: } p_{ik}^{t^+} = p_{ik}^t + \rho_{ik} G_{ik}(X_i^{t^+}, X_k^{t^+}). \tag{9b}$$

Notice that Eq. (9) generates the same $p_{ik}^{t^+}$ for any connected nodes i and k. Updating $p_{ik}^{t^+}$ locally at the terminal nodes can reduce the communication burden and enhances robustness. Let $p^t = \{p_{ik}^t, \{i,k\} \in \mathcal{E}\} \in \mathbb{R}^{4|\mathcal{E}|}$ for convenience. Algorithm 1 presents formally the proposed strategy.

Algorithm 1

Initialize:
  $X^0 \in \prod_{i \in \mathcal{N}} \mathcal{X}_i$, $p^0 = 0$, $\gamma_l^0 = 2\epsilon, \forall l \in \mathcal{N}$
Requires: an acyclic orientation of $\mathcal{G}$: $\hat{\mathcal{G}}$
For local variable X_i **at bus** i**,**
while (received X_k^t, γ_k^t from bus $k \in \mathcal{N}_i$) **and** ($\exists l \in \mathcal{N}_i$ s.t. $\gamma_l^t > \epsilon$) **do**
  If i is the tail node of $\{i,k\} \in \hat{\mathcal{E}}$,
    compute p_{ik}^t and then solve opt. (8)
  else if i is the head node of $\{i,k\} \in \hat{\mathcal{E}}$,
    solve opt. (8) and then update $p_{ik}^{t^+}$
  end
  For all $k \in \mathcal{N}_i$, compute
    $\gamma_i^{t^+} = \sum_{\{i,k\} \in \mathcal{E}} \|G_{ik}(X_i^{t^+}, X_k^{t^+})\|^2$
  Send $X_i^{t^+}$ and $\gamma_i^{t^+}$ to all $k \in \mathcal{N}_i$
end while

In Algorithm 1, each bus i only does optimization when it received updates $k \in \mathcal{N}_i$ which comes after its last iteration. The implementation of Algorithm 1 does not require synchronization between nodes while involves subtle ordering. We therefore refer Algorithm 1 as scheduled-asynchronous distributed algorithm. Note that we use the global time index t to time-stamp all the iterations in Algorithm 1 only for convenience. Every agent tracks the number of iterations locally, but in general does not know the global time.

The following result establishes that Algorithm 1 converges under mild conditions.

Theorem IV.1. ***(Convergence of Algorithm 1).*** *If the following conditions hold*

1) *Cost functions f_i, $i \in \mathcal{N}$, are convex,*
2) ***(P2)** is feasible and the Slater conditions holds,*
3) *The optimum of **(P2)** has rank$(W_i^\star) = 1$, $\forall i \in \mathcal{N}$,*
4) *The orientation $\hat{\mathcal{G}}$ is acyclic,*

then the sequence (X^t, p^t) generated by Algorithm 1 converges to the optimal primal-dual pair $(X^\star, p^\star)$ as $t \to \infty$.

We want to remark that condition 1) in Theorem IV.1 holds for most OPF problems. Conditions 2)-3) are the assumptions for exact SDP convex relaxation of the OPF problem. Condition 4) prevents the "locked" situation from happening, which we discuss in the following section.

B. Directed Graph Design and Convergence Rate

The orientation $\hat{\mathcal{G}}$ given to the electrical network graph must be free of cycles. Otherwise, given the meaning

encoded by the orientation of each edge, if a cycle was present, then the algorithm would get stuck: every node at a cycle would be waiting for the update from a neighboring node in the cycle. The network can determine an acyclic orientation in a distributed way as follows. Assign every node with a number $k \in \mathbb{N}$ so that any connected two nodes in $\mathcal{G}$ have a different number. Then, for each edge in $\mathcal{E}$, designate the node with the smallest number to be the tail node and the other node to be the head node. It can be proven that the resulting graph $\hat{\mathcal{G}}$ is acyclic, see [18] for details. Other than the acyclic requirement, it is desired to minimize the diameter of the acyclic orientation because the averaging time for two consecutive iterations for every node heavily depends on the diameter. Therefore, though finding an optimal orientation is NP-hard, we are motivated to develop a way to find such orientation.

We formally define the optimization problem that minimizes the diameter of the orientation as follows. Let ω be an acyclic orientation of $\mathcal{G}$, and Ω collect all possible acyclic orientations. The directed graph derived from $\mathcal{G}$ and $\omega \in \Omega$ is written as $\mathcal{G}_\omega$. We write the optimization of interest in the following

$$\omega^\star = \arg\min_{\omega\in\Omega} \Big(\max_{h\in\mathcal{P}_\omega} |h| \Big), \tag{10}$$

where $\mathcal{P}_\omega$ is the set of path in $\mathcal{G}_\omega$, and $|h|$ is the number of arc in the path h. The optimization (10) is directly related to a classical problem of finding the chromatic number of an undirected graph [19],

$$\mathcal{C}(\mathcal{G}) = 1 + \min_{\omega\in\Omega} \max_{h\in\mathcal{P}_\omega} |h|. \tag{11}$$

We explain how Eq. (11) holds as follows. If $\mathcal{C}(\mathcal{G}) = \overline{m}$, then $\mathcal{N}$ is partitioned into $\overline{m}$ subsets given as $\mathcal{N} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{\overline{m}}$. We can assign numbers $\{1, 2, ..., \overline{m}\}$ to all the nodes according to $\mathcal{C}(\mathcal{G})$. The smallest number is first assigned to $\mathcal{C}_1$. We then choose the second smallest number to $\mathcal{C}_2$ and repeat the procedure. Then, for each edge in $\mathcal{E}$, designate the node with the smallest number to be the tail node and the other node to be the head node. The resulting graph $\hat{\mathcal{G}}$ is acyclic because there exists at least one arc in a reverse direction with the rest of arcs in every cycle of $\mathcal{G}$. Furthermore, the diameter is bounded by $\overline{m} - 1$ because there exists no path between nodes with the same color (ζ) and the distance between same color nodes is at most $\overline{m}$.

Finding $\mathcal{C}(\mathcal{G})$ is again NP-hard and challenging to solve. There are only algorithms that approximate the solution, see for example [20]. Fortunately, the following assumption holds for most electrical networks [21], which makes finding $\mathcal{C}(\mathcal{G})$ easier

Assumption 1. ***(Planar network topology).*** *Electrical networks have simple planar network topology.*

The chromatic number of planar graphs is upper bounded by four [22]. Furthermore, there exists a quadratic time algorithm to find the four coloring of the planar graph [23]. We can assume that every agent (or bus) registers to a centralized entity. The centralized entity then use the algorithm in [23] to assign a number (or color) to every agent. In many cases, the number of unplugged agents are relatively small compared to N. The acyclic graph defined by centralized entity is then near optimal.

TABLE I

NUMBER OF ITERATIONS NEEDED FOR $\gamma_l < 10^{-4}, \forall l \in \mathcal{N}$

	N. edges	Iter./bus	Iter./bus 5% drop	Iter./bus 10% drop
6 bus	11	50	62	84
14 bus	20	100	139	184
30 bus	41	158	376	398

Both finding the chromatic number and the analysis of the convergence rate of Algorithm 1 can be simplified by considering certain graph topologies. If the electrical network is bipartite, then the chromatic number is trivially two. Furthermore, optimization (6) can be rewritten as

$$\begin{aligned} \min_{X\in\mathcal{X}} & \sum_{i\in\mathcal{N}_a} f_i(X_i) + \sum_{i\in\mathcal{N}_b} f_i(X_i) \\ \text{s.t. } & G_{ab}(X_a, X_b) = 0, \end{aligned} \tag{12}$$

where $\mathcal{N}_a \cup \mathcal{N}_b = \mathcal{N}$ forms a partition of $\mathcal{N}$, $X_a = \{X_i, i \in \mathcal{N}_a\}$, $X_b = \{X_i, i \in \mathcal{N}_b\}$, and G_{ab} is derived from G_{ik}. By assigning nodes in $\mathcal{N}_a$ with smaller numbers compared to those in $\mathcal{N}_b$, the schedule-asynchronous distributed algorithm generates an alternating update sequence between $\mathcal{N}_a$ and $\mathcal{N}_b$ that resembles ADMM. Without assuming strong convexity of f_i, ADMM has a convergence rate of $O(1/n)$ [12]. The proof of the convergence rate for optimization (12) follows similar steps as in the literature [12].

Remark IV.2. ***(Bipartite graphs in distribution networks).*** A graph is bipartite if and only if it does not has an odd cycle. A tree network topology is a special case of bipartite graph. In fact, many existing distribution networks have a tree network topology or only with few cycles, which may remain bipartite. Hence, the proposed algorithm gets $O(1/n)$ convergence rate and short idling time in many applications. □

V. SIMULATIONS

We validate the scheduled-asynchronous algorithm over the six bus testbed in [24], IEEE 14, and 30 bus testbeds. The cost functions are selected such that the OPF problems have rank-one optima. We also choose an acyclic orientation for every test case. The stopping criteria is that $\gamma_l \leq 10^{-4}$ for all $l \in \mathcal{N}$. The algorithm parameter ρ is chosen to be 700. The simulation results are showed in Table I.

We observe that the number of iteration per node grows only linearly with respected to the number of edges in the electrical network. The property shows a sub-linear rate of convergence. To test the algorithm robustness, we simulate the operation scenario in which, if one node waits for more than a certain threshold time to hear from its neighbors, it precedes with its optimization by using the state of the previous step of neighboring nodes. The 5% and 10% of unsuccessful line communications are simulated. The algorithm remains convergent at the expense of an increase in the number of iterations per node. This increase in the number of iterations can be considered the price to pay to bring the variables back to optimality under this type of disturbance.

VI. Conclusions

In this paper, we have proposed a scheduled-asynchronous distributed algorithm for OPF applications. The OPF problem is first rewritten as a combination of non-convex subproblems. With the fact that SDP convex relaxation can provide a global or near global optimal solution in most existing electrical networks, we convexify those subproblems by dropping the rank constraints. The proposed algorithm is shown to be suitable for the SDP convexified OPF problem, especially for distributed networks with tree network topology. The novel distributed algorithm includes an ADMM-like iteration between connected lines without the need of clock synchronization between buses. Simulation studies show that the complexity at one node only grows linearly with the size of the network. Future work will seek to improve the guarantee on the convergence rate for general network topologies and develop a rigorous analysis on the robustness properties over various network effects such as delays and packet dropouts.

Acknowledgments

This research was supported by ARPA-e Cooperative Agreement DE-AR0000695.

References

[1] J. Momoh, M. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993. Part I: Nonlinear and quadratic programming approaches," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 96–104, 1999.

[2] V. Ajjarapu and C. Christy, "The continuation power flow: a tool for steady state voltage stability analysis," *IEEE Transactions on Power Systems*, vol. 7, no. 1, pp. 416–423, 1992.

[3] M. Abido, "Optimal power flow using particle swarm optimization," *International Journal of Electrical Power & Energy Systems*, vol. 24, no. 7, pp. 563–571, 2002.

[4] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 92–107, 2012.

[5] B. Zhang, "Control and optimization of power systems with renewables: Voltage regulation and generator dispatch," Ph.D. dissertation, University of California, Berkeley, 2013.

[6] D. K. Molzahn, "Application of semidefinite optimization techniques to problems in electric power systems," Ph.D. dissertation, University of Wisconsin-Madison, 2013.

[7] D. K. Molzahn, C. Josz, I. A. Hiskens, and P. Panciatici, "A Laplacian-based approach for finding near globally optimal solutions to OPF problems," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 305–315, 2017.

[8] R. Madani, M. Ashraphijuo, and J. Lavaei, "Promises of conic relaxation for contingency-constrained optimal power flow problem," in *Allerton Conf. on Communications, Control and Computing*, 2014, pp. 1064–1071.

[9] A. Lam, B. Zhang, and N. T. David, "Distributed algorithms for optimal power flow problem," in *IEEE Conf. on Decision and Control*, 2012, pp. 430–437.

[10] E. Dall'Anese, H. Zhu, and G. B. Giannakis, "Distributed optimal power flow for smart microgrids," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1464–1475, 2013.

[11] C. Y. Chang and W. Zhang, "On near and exact optimal power flow solutions for microgrid applications," in *IEEE Conf. on Decision and Control*, Las Vegas, NV, 2016.

[12] B. He and X. Yuan, "On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers," *Numerische Mathematik*, vol. 130, no. 3, pp. 567–577, 2015.

[13] A. Cherukuri, E. Mallada, S. Low, and J. Cortés, "The role of convexity in saddle-point dynamics: Lyapunov function and robustness," *IEEE Transactions on Automatic Control*, 2016, submitted.

[14] D. Jakovetić, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014.

[15] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009, electronically available at http://coordinationbook.info.

[16] S. Bose, D. F. Gayme, S. H. Low, and K. M. Chandy, "Optimal power flow over tree networks," in *Allerton Conf. on Communications, Control and Computing*, 2011, pp. 1342–1348.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, 2011.

[18] R. Deming, "Acyclic orientations of a graph and chromatic and independence numbers," *Journal of Combinatorial Theory, Series B*, vol. 26, no. 1, pp. 101–110, 1979.

[19] R. Figueiredo, V. Barbosa, N. Maculan, and C. D. Souza, "Acyclic orientations with path constraints," *RAIRO-Operations Research*, vol. 42, no. 4, pp. 455–467, 2008.

[20] D. Corneil and B. Graham, "An algorithm for determining the chromatic number of a graph," *SIAM Journal on Computing*, vol. 2, no. 4, pp. 311–318, 1973.

[21] K. C. Sou, "A branch-decomposition approach to power network design," in *American Control Conference*, 2016, pp. 6483–6488.

[22] N. Robertson, D. Sanders, P. Seymour, and R. Thomas, "The four-colour theorem," *Journal of Combinatorial Theory, Series B*, vol. 70, no. 1, pp. 2–44, 1997.

[23] ——, "Efficiently four-coloring planar graphs," in *Proceedings of the twenty-eighth annual ACM symposium on theory of computing*, 1996, pp. 571–575.

[24] A. J. Wood and B. Wollenberg, *Power generation operation and control-2nd edition*. New York, USA: John Wiley and Sons, 1996, vol. 3.