Scheduled-Asynchronous Distributed Algorithm for Optimal Power Flow

Chin-Yao Chang Jorge Cortés Sonia Martínez

***Abstract*—Optimal power flow (OPF) problems are non-convex and large-scale optimization problems with important applications in power networks. This paper proposes the scheduled-asynchronous algorithm to solve a distributed semidefinite programming (SDP) formulation of the OPF problem. In this formulation, every agent seeks to solve a local optimization with its own cost function, physical constraints on its nodal power injection, voltage, and power flow of the lines it is connected to, and decision constraints on variables shared with neighbors to ensure consistency of the obtained solution. In the scheduled-asynchronous algorithm, every pair of connected nodes in the electrical network update their local variables in an alternating fashion. This strategy is asynchronous, in the sense that no clock synchronization is required, and relies on an orientation of the electrical network that prescribes the precise ordering of node updates. We establish the asymptotic convergence properties to the primal-dual optimizer when the orientation is acyclic. Given the dependence of the convergence rate on the network orientation, we also develop a distributed graph coloring algorithm that finds an orientation with diameter at most five for electrical networks with geometric degree distribution. Simulations illustrate our results on various IEEE bus test cases.**

I. Introduction

The optimal power flow (OPF) problem seeks to minimize the cost of electricity generation subject to voltage and power flow constraints. Finding a solution to the OPF problem is challenging due to its large-scale and non-convex nature and, therefore, it is typically solved off-line for centralized planning of power networks. However, recent technological advances involving the integration of renewable distributed energy resources introduce higher operational uncertainty in managing the electrical grid, motivating the need for real-time methods to solve OPF problems. Ideally, such methods should enjoy robustness against disturbances and tolerance against intermittent engagement of energy resources. An additional consideration further justifying the need for such methods is the increase of plug-and-play devices in distribution networks and the ensuing uncertain overall system configuration. Motivated by these considerations, this paper introduces a provably-correct distributed algorithm to solve a convexified OPF problem over a power network.

Literature review: Finding a global optimum of the OPF problem is challenging due to its non-convexity, so most existing algorithms only guarantee a local optimum, see e.g., [2]–[4]. A commonly used approach in the literature [5], [6] to convexify the problem is the use of DC power flow equations. An alternative route for the convexification of the OPF problem is employing semi-definite programming (SDP). The work [7] shows that the SDP convex relaxation on the OPF problem is exact for many networks, a fact that allows to find a global solution. Conditions on the exact convex relaxation have been further established in [8], [9]. Several recent works [10], [11] further develop SDP-based algorithms for a near-global optimal solution for OPF problems where the SDP convex relaxation does not provide a feasible solution. Relatively few works consider solving the OPF problem in a distributed way. The paper [12] proposes a distributed approach to the DC-OPF problem, where the electrical network is decomposed into several regions and each region solves its regional DC-OPF problem while iteratively matching its tie-line powers with the connected regions. The work [13] considers an SDP-based convexification of the OPF problem and then proposes gradient-based primal-dual algorithm which displays fast convergence to the optimum in simulation studies. The algorithm design is limited to voltage magnitude constraints and linear objective functions and does not incorporate constraints on the active/reactive power. The work [14] also considers a SDP-based relaxation of the OPF problem, partitions a so-called weakly-meshed network into several areas so that the "macro" graph describing the interconnected areas has a tree topology, and applies the alternating direction method of multipliers (ADMM) to solve the distributed optimization associated with the macro graph. In general, gradient-based methods are amenable to asynchronous implementations, but have a rate of convergence $O(\log(n)/n)$ or slower [15]. In comparison, ADMM is faster with a $O(1/n)$ convergence rate [16], while requiring a synchronous implementation. Drawing connections with the increasing body of work on gossiping in network systems [17]–[19], our algorithm design prescribes pairwise updates between neighboring agents to avoid the requirement of clock synchronization while enjoying similar convergence properties as ADMM.

Statement of contributions: Our starting point for the algorithm design is the formulation of a distributed version of the OPF problem, where each bus (node) plays the role of an agent with computation and communication capabilities. Every node creates local copies of the voltage of the neighboring nodes which are physically connected to it. Each node poses an optimization with its local cost function, constraints of its nodal power injection and voltage, and constraints associated with the power flow of the connected lines. On top of

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

A preliminary version of this work appeared as [1] at the 2017 American Control Conference.

those local physical constraints, every node also has equality constraints sharing to its neighbors to ensure that every copy of the variables coincides. We then consider the convexification of this distributed OPF formulation using semi-definite programming (SDP). Our first contribution is the synthesis of the scheduled-asynchronous algorithm to solve the distributed OPF formulation in a distributed way. In our design, every pair of connected buses solves their local optimization problem in an ADMM-like, alternating fashion. To establish the order of such alternating iterations across the network, every bus only updates its variables when all the neighboring buses have finished an update later than its last update. This logic does not require any clock synchronization between the agents, in contrast to ADMM. Our second contribution is the convergence analysis of the proposed scheduled-asynchronous algorithm. Under reasonable assumptions on the data defining the optimization problem and the requirement than the graph orientation is acyclic (to avoid "locked" situations where every bus is waiting for an update from at least one of its neighbors), we employ the LaSalle Invariance Principle to establish the convergence to the primal-dual solution. Given the dependence of the algorithm convergence on the orientation of the network graph, our third contribution concerns the optimal selection of this orientation. In fact, the time need for all agents to finish at least one update is directly related to the diameter of the oriented network graph, and hence it is desirable to minimize it. In general, finding an orientation that minimizes the diameter is equivalent to finding the chromatic number of the graph, which is NP-hard. We design a distributed graph coloring algorithm that finds an orientation with diameter at most five for electrical networks with geometric degree distribution. The distributed nature of this strategy makes it naturally robust to changes in the network topology. Simulations on various IEEE bus test cases of the combined graph coloring and scheduled-asynchronous algorithms illustrate the performance and robustness of the proposed design.

Organization: Section II presents basic concepts and notation. Section III introduces the OPF problem and its distributed formulation. Section IV presents the scheduled-asynchronous algorithm and analyzes its convergence properties. Section V introduces a distributed graph coloring algorithm to obtain an acyclic orientation of small diameter. Section VI illustrates the effectiveness of the proposed algorithm in benchmark IEEE test cases. Finally, we gather our conclusions in Section VII.

II. Preliminaries

This section introduces basic notation and concepts from graph theory and optimization.

A. Notation

We denote by $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{C}$ the sets of positive integers, reals and complex numbers, respectively. We denote by $|\mathcal{N}|$ the cardinality of the set $\mathcal{N}$. For a complex number $a \in \mathbb{C}$, we let $|a|$ and $\angle a$ be the complex modulus and angle of a. The 2-norm of a complex vector $v \in \mathbb{C}^n$ is written as $\|v\|$. Let $\mathbb{S}_+ \subset \mathbb{C}^{n\times n}$ and $\mathcal{H}^n \subset \mathbb{S}_+$ be the set of positive semidefinite and n-dimensional Hermitian matrices, respectively. For $A \in \mathbb{C}^{n\times n}$, we let A^* be its conjugate transpose and $\mathbf{Tr}\{A\}$ be its trace. For $A, B \in \mathcal{H}^n$, we denote their inner product by $\langle A, B\rangle = \mathbf{Tr}\{AB\}$. We use $\bigtriangledown F$ to denote the gradient of the scalar function F.

B. Graph Theory

We review basic notions of graph theory following [20]. A graph is a pair $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} \subseteq \mathbb{N}$ is its set of vertices or nodes and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is its set of edges. A *loop* is an edge that connects a vertex to itself. Two nodes $i, k \in \mathcal{N}$ are *connected* if $\{i, k\} \in \mathcal{E}$. The graph is *undirected* if $\{i, k\} = \{k, i\} \in \mathcal{E}$. The *local neighborhood* of a node k in an undirected graph is $\mathcal{N}_k := \{l \in \mathcal{N} \mid \{l, k\} \in \mathcal{E}\} \cup \{k\}$. The *degree* of a node k is $|\mathcal{N}_k| - 1$. In a directed graph, each pair of vertices in $\mathcal{E}$ is ordered such that the corresponding edge $\{i, k\}$ has a direction, with i and k distinguished as the *tail* and *head* nodes, respectively. Node i is a *source* (resp. *sink*) if it is the tail (resp. head) node of all the edges it belongs to. Node k is an *out-neighbor* of node i if $\{i, k\} \in \mathcal{E}$. The *out-degree* of node i is defined as $|\{k \in \mathcal{N} \,|\{i, k\} \in \mathcal{E}\}|$. A *path* in a (directed or undirected) graph is a sequence of vertices such that any two consecutive nodes correspond to an edge of the graph. The *length* of a path is the number of its corresponding edges. The *diameter* of a graph is the maximum length of the shortest path connecting any two graph vertices in the graph. A *cycle* is a path whose first and last vertices are the same. A graph is *acyclic* if it contains no cycles. An *orientation* of an undirected graph is an assignment of exactly one direction to each of its edges. A graph orientation is acyclic if the resulting directed graph is acyclic. A graph is *bipartite* if the set of its vertices can be decomposed into two disjoint subsets such that, within each one, no two vertices are connected.

A *simple graph* is a graph with no loops nor multiple edges connecting any pair of two vertices. A *planar graph* is a graph that can be drawn on the plane in a way that its edges intersect only at their endpoints. A *vertex-induced subgraph* of $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, written as $\mathcal{G}_s[\mathcal{N}_s]$, is a subgraph of $\mathcal{G}$ with the set of nodes $\mathcal{N}_s \subseteq \mathcal{N}$ and set of edges $\mathcal{E}_s = \mathcal{E} \cap (\mathcal{N}_s \times \mathcal{N}_s)$. A *chordal graph* is a graph that does not contain an induced cycle of length greater than four. *Graph coloring* consists of assigning a color to every node in the graph in such a way that any pair of connected nodes have different colors. The smallest number of colors needed to color a graph G is called its chromatic number.

C. Strong Duality of Convex Optimization

Here, we review some fundamental concepts in convex optimization following [21]. Consider a convex optimization problem of the form

$$\min_x f_0(x), \quad \text{s.t. } Ax = b,\ f_i(x) \le 0,\ i = 1, \dots, m, \qquad (1)$$

where $f_0, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex functions, $A \in \mathbb{R}^{n\times r}$, $b \in \mathbb{R}^r$, and $Ax = b$ defines affine equality constraints. The dual problem of optimization (1) is given as

$$\max_{\lambda\geq 0,\mu} \Big(\min_x f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \mu^\top (Ax - b)\Big), \quad (2)$$

where $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^r$ are known as Lagrange multipliers. Let $p^\star$ and $d^\star$ be the optimal value of the primal and dual problems, respectively. Strong duality holds if $p^\star = d^\star$. Under strong duality, the Karush-Kuhn-Tucker (KKT) conditions are a necessary and sufficient characterization of the optimality of the primal-dual solution $(x^\star, \lambda^\star, \mu^\star)$,

$$\begin{cases} 0 \in \bigtriangledown f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star \bigtriangledown f_i(x^\star) + (\mu^\star)^\top A x^\star, \\ \lambda_i^\star f_i(x^\star) = 0, \quad \forall i = 1, \ldots, m, \\ (\mu^\star)^\top (Ax^\star - b) = 0, \\ Ax^\star = b, \quad f_i(x^\star) \leq 0, \quad \forall i = 1, \ldots, m, \\ \lambda_i^\star \geq 0, \quad \forall i = 1, \ldots, m. \end{cases}$$

These conditions correspond to stationarity, complementary slackness, and primal and dual feasibility, respectively. The (refined) Slater's condition holds if there exists $x \in \mathbb{R}^n$ with

$$Ax = b \text{ and } f_i(x) < 0, \quad \forall i = 1, \ldots, m.$$

Slater's condition implies that strong duality holds.

III. Problem Formulation

This section introduces the problem of interest. We begin with a general formulation of the optimal power flow (OPF) problem over an electrical network. We then consider its convex relaxation and rewrite it as the combination of several smaller-scale interconnected convex subproblems. The resulting semidefinite programming (SDP) problem is the starting point for our distributed algorithm design.

Consider an electrical network graph with generation buses $\mathcal{N}_G$, load buses $\mathcal{N}_L$, and electrical interconnections described by an undirected 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. When convenient, we let $V = \{V_i \mid i \in \mathcal{N}\}$ denote the collection of voltages at all buses. The active and reactive power injections at bus i are given by the power flow equations [22]

$$P_i = \mathbf{Tr}\{Y_i V V^*\} + P_{D_i},$$
$$Q_i = \mathbf{Tr}\{\overline{Y}_i V V^*\} + Q_{D_i},$$

where $P_{D_i}, Q_{D_i} \in \mathbb{R}$ are the active and reactive power demands[1] at bus i, and $Y_i, \overline{Y}_i \in \mathcal{H}^N$ are derived from the admittance matrix $\mathbf{Y} \in \mathbb{C}^{N\times N}$ as follows

$$Y_i = \frac{(e_i e_i^\top \mathbf{Y})^* + e_i e_i^\top \mathbf{Y}}{2}, \quad (3a)$$

$$\overline{Y}_i = \frac{(e_i e_i^\top \mathbf{Y})^* - e_i e_i^\top \mathbf{Y}}{2j}. \quad (3b)$$

[1]Some buses may have generation and load simultaneously. For buses with only generators, P_{D_i}, Q_{D_i} are both zero.

Here $\{e_i\}_{i=1,\ldots,N}$ denotes the canonical basis of $\mathbb{R}^N$. The OPF problem also involves the following box constraints

$$\begin{aligned} &\underline{V}_i^2 \leq |V_i|^2 \leq \overline{V}_i^2, \ \forall i \in \mathcal{N}, \\ &\underline{P}_i \leq P_i \leq \overline{P}_i, \ \ \underline{Q}_i \leq Q_i \leq \overline{Q}_i, \ \forall i \in \mathcal{N}, \\ &|V_i - V_k|^2 \leq \overline{V}_{ik}, \ \forall \{i,k\} \in \mathcal{E}, \end{aligned} \quad (4)$$

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

$$\sum_{k\in\mathcal{N}_G} c_{i2} P_i^2 + c_{i1} P_i, \quad (5)$$

where $c_{i2} \geq 0$, and $c_{i1} \in \mathbb{R}$. Using $W = VV^* \in \mathcal{H}^N$ as the decision variable, the OPF problem is formulated as follows

$$\textbf{(P1)} \ \min_W \sum_{i\in\mathcal{N}_G} c_{i2}(\mathbf{Tr}\{Y_i W\} + P_{D_i})^2 + c_{i1}(\mathbf{Tr}\{Y_i W\} + P_{D_i}),$$

subject to

$$\underline{P}_i \leq \mathbf{Tr}\{Y_i W\} + P_{D_i} \leq \overline{P}_i, \ \forall i \in \mathcal{N}, \quad (6a)$$
$$\underline{Q}_i \leq \mathbf{Tr}\{\overline{Y}_i W\} + Q_{D_i} \leq \overline{Q}_i, \ \forall i \in \mathcal{N}, \quad (6b)$$
$$\underline{V}_i^2 \leq \mathbf{Tr}\{M_i W\} \leq \overline{V}_i^2, \ \forall i \in \mathcal{N}, \quad (6c)$$
$$\mathbf{Tr}\{M_{ik} W\} \leq \overline{V}_{ik}, \ \forall \{i,k\} \in \mathcal{E}, \quad (6d)$$
$$W \succeq 0, \quad \mathrm{rank}(W) = 1, \quad (6e)$$

where $M_i, M_{ik} \in \mathcal{H}^N$ are defined so that $\mathbf{Tr}\{M_i W\} = |V_i|^2$ and $\mathbf{Tr}\{M_{ik} W\} = |V_i - V_k|^2$. Constraints (6a-6d) come from Eq. (4). The combined constraints $W \succeq 0$ and $\mathrm{rank}(W) = 1$ in (6e) correspond to writing the voltage as a matrix variable. The elimination of the rank constraint gives rise to the convex relaxation of the OPF problem.

Following the exposition in [23], we next reformulate the OPF problem in a distributed way as follows. We start from the observation that every constraint except Eq. (6e) 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. This property is embedded in the structure of the non-zero entries of the admittance matrix in (3). As a consequence, we can rewrite the constraints (6a-6d) in terms of variables $W_i \in \mathcal{H}^{N_i}$ for all $i \in \mathcal{N}$, where W_i is quadratic in the voltage variables corresponding to the local neighborhood $\mathcal{N}_i$ in the electrical network graph, and $N_i = |\mathcal{N}_i|$. In this way, we obtain

$$\underline{P}_i \leq \mathbf{Tr}\{Y_{i,r} W_i\} + P_{D_i} \leq \overline{P}_i, \ \forall i \in \mathcal{N}, \quad (7a)$$
$$\underline{Q}_i \leq \mathbf{Tr}\{\overline{Y}_{i,r} W_i\} + Q_{D_i} \leq \overline{Q}_i, \ \forall i \in \mathcal{N}, \quad (7b)$$
$$\underline{V}_i^2 \leq \mathbf{Tr}\{M_{i,r} W_i\} \leq \overline{V}_i^2, \ \forall i \in \mathcal{N}, \quad (7c)$$
$$\mathbf{Tr}\{M_{ik,r} W_k\} \leq \overline{V}_{ik}, \ \forall \{i,k\} \in \mathcal{E}. \quad (7d)$$

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

Define $W_{\mathcal{N}}$ as a shorthand notation for the set of variables, W_i for $i \in \mathcal{N}$, i.e., $W_{\mathcal{N}} := \{W_i, i \in \mathcal{N}\}$. We next consider the following distributed convex optimization problem associated with $W_{\mathcal{N}}$,

$$\textbf{(P2)} \quad \min_{W_{\mathcal{N}}} \sum_{i \in \mathcal{N}_G} c_{i2}(\mathbf{Tr}\{Y_{i,r}W_i\} + P_{D_i})^2 + c_{i1}(\mathbf{Tr}\{Y_{i,r}W_i\} + P_{D_i}), \tag{8a}$$

subject to

$$\text{Eq. (7) holds}, \tag{8b}$$
$$W_i \succeq 0, \quad \forall i \in \mathcal{N}, \tag{8c}$$
$$W_i(\hat{i},\hat{i}) = W_k(\hat{i},\hat{i}),\ \forall \{i,k\} \in \mathcal{E}, \tag{8d}$$
$$W_i(\hat{i},\hat{k}) = W_k(\hat{i},\hat{k}),\ \forall \{i,k\} \in \mathcal{E}, \tag{8e}$$

where $\hat{i}$ refers the row (or column) of the matrix associated with bus i (note that W_i and W_k might have different dimensions). For chordal graphs, **(P2)** with the additional non-convex rank constraints, $\mathrm{rank}(W_i) = 1, \forall i \in \mathcal{N}$ is equivalent to **(P1)**, see [24]. For general graphs, **(P2)** with the rank constraints does not necessarily give an optimal solution of **(P1)**, but simulations indicate [25] that **(P2)** has a low-rank solution whose value is close to the optimal value of **(P1)**. In the rest of the paper, we assume that a unique optimal solution of **(P2)** exists, and we denote it as $(W^\star_{\mathcal{N}}, p^\star)$.

Our objective in this paper is to design a distributed algorithm to solve **(P2)**. We view each bus of the electrical network as a computing agent that can communicate with any other bus which is physically connected to. By distributed, we mean that each agent only requires information from neighboring buses that share their local variables to implement the algorithm. By solving the optimization problem, we mean that each bus eventually finds its own optimal allocation (not the optimal allocation for the whole electrical network). When considered collectively, the local optimal allocation with agreement on the shared variables yields the complete optimal solution.

IV. The Scheduled-Asynchronous Algorithm

In this section, we first provide a design rationale for the scheduled-asynchronous distributed algorithm and then introduce it formally. We next proceed to characterize the algorithm convergence properties.

A. Rationale for Algorithm Design

For convenience of exposition, we start by rewriting the optimization **(P2)**. To this end, for each $i \in \mathcal{N}$, define $f_i : \mathcal{H}^{N_i} \to \mathbb{R}$ as the objective function, and let $\mathcal{W}_i \subset \mathcal{H}^{N_i}$ be the constraint set defined by (7) and the constraint $W_i \succeq 0$. Note that $\mathcal{W}_i$ is compact for every $i \in \mathcal{N}$, where the boundedness of $\mathcal{W}_i$ is the result of the bounded diagonal elements of W_i and the positive definiteness of W_i. The objective function f_i is given by (8a), for $i \in \mathcal{N}_G$, and $f_i = 0$, for $i \in \mathcal{N} \setminus \mathcal{N}_G$. To represent the equality constraints in (8d) and (8e), we introduce the functions $G_{ik} : \mathcal{H}^{N_i} \times \mathcal{H}^{N_k} \to \mathbb{R}^4$,

$$G_{ik}(W_i, W_k) = D_{ik}(W_i) + D_{ki}(W_k), \tag{9}$$

where

$$D_{ki}(W_l) = \begin{bmatrix} \mathbf{Tr}\{B_{1,ki}W_l\} \\ \mathbf{Tr}\{B_{2,ki}W_l\} \\ \mathbf{Tr}\{B_{3,ki}W_l\} \\ \mathbf{Tr}\{B_{4,ki}W_l\} \end{bmatrix}, D_{ik}(W_l) = -\begin{bmatrix} \mathbf{Tr}\{B_{2,ik}W_l\} \\ \mathbf{Tr}\{B_{1,ik}W_l\} \\ \mathbf{Tr}\{B_{3,ki}W_l\} \\ \mathbf{Tr}\{B_{4,ki}W_l\} \end{bmatrix}.$$

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

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

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

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

Note that the linear equality constraints (8d)-(8e) can be equivalently represented in compact form by $G_{ik}(W_i, W_k) = 0$, for all $\{i,k\} \in \hat{\mathcal{E}}$, where $\hat{\mathcal{G}} = (\mathcal{N}, \hat{\mathcal{E}})$ is an arbitrarily selected orientation of the original undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. With these elements in place, we rewrite the optimization **(P2)** in the following form

$$\min_{W_i \in \mathcal{W}_i, i \in \mathcal{N}} \sum_{i \in \mathcal{N}} f_i(W_i) \quad \text{s.t. } G_{ik}(W_i, W_k) = 0,\ \forall \{i,k\} \in \hat{\mathcal{E}}. \tag{10}$$

To motivate our algorithm design, we start by considering the optimization in **(P2)** for a two-bus network ($N = 2$). In this case, from the formulation (10), the problem exactly corresponds to the standard ADMM, see e.g., [26],

$$\min_{W_1 \in \mathcal{W}_1, W_2 \in \mathcal{W}_2} f_1(W_1) + f_2(W_2) \quad \text{s.t. } G_{12}(W_1, W_2) = 0.$$

The ADMM algorithm consists of the following steps

$$W_1^{t^+} = \mathrm{argmin}_{W_1 \in \mathcal{W}_1} f_1(W_1) + {p_{12}^t}^\top G_{12}(W_1, W_2^t) + \frac{\rho_{12}}{2}\|G_{12}(W_1, W_2^t)\|^2, \tag{11a}$$

$$W_2^{t^+} = \mathrm{argmin}_{W_2 \in \mathcal{W}_2} f_2(W_2) + {p_{12}^t}^\top G_{12}(W_1^{t^+}, W_2) + \frac{\rho_{12}}{2}\|G_{12}(W_1^{t^+}, W_2)\|^2, \tag{11b}$$

$$p_{12}^{t^+} = p_{12}^t + \rho_{12} G_{12}(W_1^{t^+}, W_2^{t^+}), \tag{11c}$$

where the superscript t is the time at which the update occurs, t^+ is the time for the next round of the optimization, $\rho_{12} > 0$ is a given constant scalar, and $p_{12}^t \in \mathbb{R}^4$ are the Lagrange multipliers associated with the constraint $G_{12}(\cdot) = 0$. Note that there is a natural order in performing the updates in (11), where node 1 goes first, and then node 2 uses the value obtained by 1 to perform its update.

For a network with an arbitrary number of buses, one can view the optimization as a combination of multiple two-bus sub-problems. This viewpoint inspires the following algorithm design. Once bus i receives the updated W_k^t from all its neighboring nodes $k \in \mathcal{N}_i$, it solves the following optimization

$$W_i^{t^+} = \mathrm{argmin}_{W_i \in \mathcal{W}_i} f_i(W_i) \tag{12a}$$

$$+\sum_{\{i,k\}\in\hat{\mathcal{E}}}\Big({p_{ik}^t}^\top G_{ik}(W_i,W_k^t)+\frac{\rho_{ik}}{2}\|G_{ik}(W_i,W_k^t)\|^2\Big) \quad (12b)$$

$$+\sum_{\{k,i\}\in\hat{\mathcal{E}}}\Big({p_{ik}^t}^\top G_{ki}(W_k^{t^+},W_i)+\frac{\rho_{ik}}{2}\|G_{ki}(W_k^{t^+},W_i)\|^2\Big), \quad (12c)$$

where p_{ik}^t and ρ_{ik} are non-directional, namely, $p_{ik}^t = p_{ki}^t$ and $\rho_{ik} = \rho_{ki}$. Note that, according to (12), for every edge (i.e., for every two-bus sub-problem), the tail node performs first the update, followed by the head node. In other words, the orientation $\hat{\mathcal{G}} = (\mathcal{N}, \hat{\mathcal{E}})$ encodes the natural ordering of updating by the terminal nodes present in the ADMM algorithm. 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. For each pair of connected nodes i and k such that $\{i,k\} \in \hat{\mathcal{E}}$, each of them updates the corresponding Lagrange multiplier according to

$$p_{ik}^{t^+} = p_{ik}^t + \rho_{ik} G_{ik}(W_i^{t^+}, W_k^{t^+}). \quad (13)$$

Updating $p_{ik}^{t^+}$ locally at the terminal nodes can reduce the communication burden and enhances robustness. We denote $p^t = \{p_{ik}^t, \{i,k\} \in \mathcal{E}\} \in \mathbb{R}^{4|\mathcal{E}|}$ and $\rho \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ be the diagonal matrix such that each diagonal element corresponds to ρ_{ik} of one unique link in $\mathcal{E}$. Algorithm 1 below presents formally the proposed strategy.

Algorithm 1 Scheduled-Asynchronous Algorithm

Initialize:
$W_{\mathcal{N}}^0 \in \prod_{i\in\mathcal{N}} \mathcal{W}_i$, $p^0 = 0$,
$\gamma_l^0 = 2\epsilon > 0$, $\forall l \in \mathcal{N}$
Requires: acyclic orientation $\hat{\mathcal{G}}$ of $\mathcal{G}$
For every bus i**,**
while (received W_k^t, γ_k^t from all $k \in \mathcal{N}_i$) **and** ($\exists l \in \mathcal{N}_i$ s.t. $\gamma_l^t > \epsilon$) **do**
  Update $p_{ik}^{t^+}$ by Eq. (13) for $\{i,k\} \in \hat{\mathcal{E}}$
  Update $W_i^{t^+}$ by solving optimization (12)
  Update $p_{ik}^{t^+}$ by Eq. (13) for $\{k,i\} \in \hat{\mathcal{E}}$
  Compute $\gamma_i^{t^+}$ by Eq. (14)
  Send $W_i^{t^+}$ and $\gamma_i^{t^+}$ to all $k \in \mathcal{N}_i$
end while

In Algorithm 1, each bus i only does its optimization after it has received new updates from all its neighbors $k \in \mathcal{N}_i$ since the last iteration. The stopping criteria is given by the scale of the violation of the equality constraints,

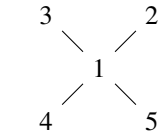
$$\gamma_i^{t^+} = \sum_{\{i,k\}\in\hat{\mathcal{E}}} \|G_{ik}(W_i^{t^+}, W_k^t)\|^2 + \sum_{\{k,i\}\in\hat{\mathcal{E}}} \|G_{ki}(W_k^{t^+}, W_i^{t^+})\|^2. \quad (14)$$

This criteria is justified by the observation that if $\gamma_i^{t^+} = 0$, $\forall i \in \mathcal{N}$, then $W_{\mathcal{N}}^{t^+}$ is the optimal solution for both (12) and (10). Using the continuity of the cost function, f_i, having $\gamma_i^{t^+}$ sufficiently small for all $i \in \mathcal{N}$ guarantees that the solution of Algorithm 1 is reasonably close to the optimum. The implementation of Algorithm 1 does not require synchronous updates between connected nodes, but involves a subtle ordering. We therefore term this strategy as the *scheduled-asynchronous* algorithm. Note that we use a 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 index.

Remark IV.1. ***(The orientation of the network graph must be acyclic).*** An important observation regarding the execution of Algorithm 1 is that 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, a deadlock would occur: every node at the cycle would be waiting for the update from a neighboring node in the cycle. There are various ways in which the network can determine an acyclic orientation in a distributed way. For instance, if every node has a unique identity $k \in \mathbb{N}$, then, for each edge in $\mathcal{E}$, one can designate the node with the smallest identity as the tail and the other node as the head. The resulting graph $\hat{\mathcal{G}}$ is acyclic [27]. We revisit this point in Section V below. □

Remark IV.2. ***(The scheduled-asynchronous algorithm as a single-valued map).*** The scheduled-asynchronous algorithm is, in general, a set-valued map due to the argmin operator in (12). However, we argue here that it can be seen as a single-valued map upon further examination of (12). Notice that, except for the constraint $W_i \succeq 0$, all the other constraints and the objective function of (12) are only related to the entries associated with the star network centered at node i. We illustrate the meaning of entries associated with a star network in an example with 5 nodes. For W_1, its diagonal, first column, and first row elements are the entries associated with the star network centered at node one.

$$W_1 = \begin{bmatrix} W_1(1,1) & W_1(1,2) & \cdots & W_1(1,5) \\ W_1(2,1) & W_1(2,2) & & W_1(2,5) \\ \vdots & & \ddots & \vdots \\ W_1(5,1) & W_1(5,2) & \cdots & W_1(5,5) \end{bmatrix}$$



We refer to the entries of W_i that are not associated with the star network centered at node i as "irrelevant", because those entries can take any value without affecting the optimal value of (12) as long as $W_i \succeq 0$ remains satisfied. Without loss of generality, Algorithm 1 can always assign zeros to the irrelevant entries. Such assignment makes W_i a Hermitian matrix associated with a star network, and [23, Proposition 3] ensures that $W_i \succeq 0$. The manipulation above makes the objective function of (12) strongly convex on the decision variables (irrelevant entries are considered as constants) due to the quadratic terms in (12b) and (12c). This observation justifies the interpretation of the scheduled-asynchronous algorithm as a single-valued map $F : \mathcal{W} \times \mathbb{R}^{4|\hat{\mathcal{E}}|} \to \mathcal{W} \times \mathbb{R}^{4|\hat{\mathcal{E}}|}$ with $\mathcal{W} := \mathcal{W}_1 \times \mathcal{W}_2 \times \cdots \times \mathcal{W}_N$. □

B. Convergence Analysis

The analysis of the convergence properties of the scheduled-asynchronous algorithm requires a careful consideration of the asynchronous updates of the nodes. In what follows and for convenience, we view the time index of the decision variables as an iteration index. For each bus $i \in \mathcal{N}$, let $t_i(n)$ be the time index at which i has exactly performed n number of the minimizations described in (12). By definition, for each

n, we have $W_i^t = W_i^{t_i(n)}$, for $t_i(n) \leq t < t_i(n+1)$. With a slight abuse of notation, in the following we use the shorthand notation W_i^n, p_i^n and r_i^n instead of the corresponding $W_i^{t_i(n)}$, $p_i^{t_i(n)}$ and $r_i^{t_i(n)}$, where $r^t = \{r_{ik}^t\ ,\ \{i,k\} \in \hat{\mathcal{E}}\} \in \mathbb{R}^{4|\hat{\mathcal{E}}|}$ and $r_{ik}^t = G_{ik}(W_i^t, W_k^t)$. With all the elements in place, we next establish the convergence properties of Algorithm 1.

Theorem IV.3. ***(Convergence of Algorithm 1).*** *Assume the following conditions hold*

1) *the cost functions f_i, $i \in \mathcal{N}$, are convex,*
2) ***(P2)** is feasible and Slater's condition holds,*
3) *the optimal Lagrange multipliers are bounded, $p^\star < \infty$,*
4) *the orientation $\hat{\mathcal{G}}$ is acyclic.*

Then, the sequence $(W_{\mathcal{N}}^n, p^n)$ generated by Algorithm 1 converges to the optimal primal-dual pair $(W_{\mathcal{N}}^\star, p^\star)$ as $n \to \infty$.

Proof. Our proof strategy consists of employing the LaSalle's Invariance Principle for discrete-time systems, [Theorem 1.19] in [20]. To this goal, we will justify that all the assumptions of the LaSalle's theorem hold. First, according to Remark IV.2, we can view F as a single-valued mapping without loss of generality. Furthermore, F is continuous due to maximum theorem and condition 1). We next consider the candidate LaSalle function

$$V(W_{\mathcal{N}}, p) := \sum_{\{i,k\}\in\hat{\mathcal{E}}} \left(\frac{\|p_{ik} - p_{ik}^\star\|^2}{\rho_{ik}} + \rho_{ik}\|D_{ki}(W_k - W_k^\star)\|^2 \right). \tag{15}$$

For notational convenience, we write $V^n = V(W_{\mathcal{N}}^n, p^n)$. Recall that we assume $p^\star$ is bounded in condition 3) and $W_{\mathcal{N}}^\star$ is also bounded because $\mathcal{W}$ is compact. It follows that $V^0 < \infty$ for any bounded initial $(W_{\mathcal{N}}^0, p^0)$.

Monotonicity of LaSalle function. We next show that V is monotonically non-increasing along the solutions of (12)-(13), specifically,

$$V^{n+1} - V^n \leq -\sum_{\{i,k\}\in\hat{\mathcal{E}}} \rho_{ik}\|r_{ik}^{n+1} - D_{ki}(W_k^{n+1} - W_k^n)\|^2. \tag{16}$$

To show this inequality, we first sum the inequalities in Lemma A.2 to obtain

$$\sum_{\{i,k\}\in\hat{\mathcal{E}}} \rho_{ik} D_{ki}^\top (W_k^{n+1} - W_k^n) D_{ik} (W_i^{n+1} - W_i^\star) \geq (p^{n+1} - p^\star)^\top r^{n+1}.$$

Using $r^{n+1} = \rho^{-1}(p^{n+1} - p^n)$ and $r^\star = 0$, we rewrite the inequality above as

$$\begin{aligned} &2\sum_{\{i,k\}\in\hat{\mathcal{E}}} \rho_{ik} D_{ki}^\top (W_k^n - W_k^{n+1}) \Big(D_{ki}(W_k^{n+1} - W_k^\star) - r_{ik}^{n+1} \Big) \\ &\quad \geq 2(p^{n+1} - p^\star)^\top \rho^{-1} (p^{n+1} - p^n). \end{aligned}$$

Using the fact that

$$\begin{aligned} &2(p^{n+1} - p^\star)^\top \rho^{-1}(p^{n+1} - p^n) = \|\sqrt{\rho}^{-1}(p^{n+1} - p^\star)\|^2 \\ &\quad + \|\sqrt{\rho}^{-1}(p^{n+1} - p^n)\|^2 - \|\sqrt{\rho}^{-1}(p^n - p^\star)\|^2, \\ &2D_{ki}^\top (W_k^n - W_k^{n+1}) D_{ki}(W_k^{n+1} - W_k^\star) = \|D_{ki}(W_k^n - W_k^\star)\|^2 \\ &\quad - \|D_{ki}(W_k^{n+1} - W_k^n)\|^2 - \|D_{ki}(W_k^{n+1} - W_k^\star)\|^2, \end{aligned}$$

where $\sqrt{\rho}$ denotes the element-wise square root of the diagonal matrix ρ, then

$$\begin{aligned} &-\sum_{\{i,k\}\in\hat{\mathcal{E}}} \rho_{ik} \Big(\|D_{ki}(W_k^{n+1} - W_k^\star)\|^2 - \|D_{ki}(W_k^n - W_k^\star)\|^2 \\ &+ \|D_{ki}(W_k^{n+1} - W_k^n)\|^2 + 2D_{ki}^\top (W_k^n - W_k^{n+1}) r_{ik}^{n+1} \\ &+ \|r_{ik}^{n+1}\|^2 \Big) \geq \|\sqrt{\rho}^{-1}(p^{n+1} - p^\star)\|^2 - \|\sqrt{\rho}^{-1}(p^n - p^\star)\|^2. \end{aligned}$$

Using the definition (15) of V, we can identify the terms V^n and V^{n+1} in the inequality above to obtain

$$\begin{aligned} &-\sum_{\{i,k\}\in\hat{\mathcal{E}}} \rho_{ik} \Big(\|D_{ki}(W_k^{n+1} - W_k^n)\|^2 + 2{r_{ik}^{n+1}}^\top \\ &\cdot D_{ki}(W_k^n - W_k^{n+1}) + \|r_{ik}^{n+1}\|^2 \Big) \geq V^{n+1} - V^n \end{aligned} \tag{17}$$

Rearranging the left-hand side leads to (16).

Bounded Trajectories. We next justify that the trajectories of Algorithm 1 are bounded. According to (A.39), the primal variable $W_{\mathcal{N}}^n$ always evolves in a compact set $\mathcal{W}$ and therefore is bounded. To show that the evolution of p^n is also bounded, we can reason by contradiction. If it were not, then the sequence $V(W^n, p^n)$ would go to infinity, and this would contradict the fact that the sublevel sets of V are invariant (which is as a consequence of (16)).

Application of LaSalle Invariance Principle. Given our discussion above, all assumptions of the LaSalle Invariance Principle [20] hold and we conclude that as $n \to \infty$, $(W_{\mathcal{N}}^n, p^n)$ converges to the largest invariant set $\mathcal{I}_I$ contained in $\mathcal{I}_0$, where

$$\mathcal{I}_0 := \{(W_{\mathcal{N}}, p)\,|\,V(F(W_{\mathcal{N}}, p)) - V(W_{\mathcal{N}}, p) = 0\}. \tag{18}$$

Our final step to establish the result is to show that $\mathcal{I}_I = \{(W_{\mathcal{N}}, p) | V(W_{\mathcal{N}}, p) = 0\}$. To this end, let $(W_{\mathcal{N}}^0, p^0)$ be an arbitrary point in $\mathcal{I}_I$. Consider the algorithm trajectory starting from $(W_{\mathcal{N}}^0, p^0)$, which by definition of the notion of invariance must remain in $\mathcal{I}_0$. The next equalities must hold because of the definition of $\mathcal{I}_0$,

$$\begin{aligned} r_{ik}^{n+1} &= D_{ki}(W_k^{n+1} - W_k^n), && \text{(19a)} \\ \iff D_{ik}(W_i^{n+1}) &= -D_{ki}(W_k^n), && \text{(19b)} \\ \iff r_{ik}^{n+1} &= D_{ik}(W_i^{n+1} - W_i^{n+2}), && \text{(19c)} \end{aligned}$$

for all $\{i,k\} \in \hat{\mathcal{E}}$ and $n \geq 0$ because the right-hand side of (16) should be zero. Let $\mathbb{K}_0$ be the set of sink nodes in $\hat{\mathcal{G}}$. The primal variable update for $i \in \mathbb{K}_0$ is given as

$$\begin{aligned} W_i^{n+1} &= \text{argmin}_{W_i \in \mathcal{W}_i} f_i(W_i) && \text{(20a)} \\ &\quad + \sum_{\{k,i\}\in\hat{\mathcal{E}}} \Big({p_{ik}^n}^\top G_{ki}(W_k^{n+1}, W_i) + \frac{\rho_{ik}}{2}\|G_{ki}(W_k^{n+1}, W_i)\|^2 \Big) \\ &= \text{argmin}_{W_i \in \mathcal{W}_i} f_i(W_i) && \text{(20b)} \\ &\quad + \sum_{\{k,i\}\in\hat{\mathcal{E}}} \Big(\frac{\rho_{ik}}{2}\|G_{ki}(W_k^{n+1}, W_i) + \frac{p_{ik}^n}{\rho_{ik}}\|^2 \Big). \end{aligned}$$

Eq. (20b) follows by completing the squares inside the sum over $\hat{\mathcal{E}}$ in (20a) (we also drop the term ${p_{ik}^n}^2/2\rho_{ik}$ because this

does not affect the argmin operation). Now we analyze how the last quadratic term in (20b) evolves as n increases

$$\begin{aligned}&\frac{\rho_{ik}}{2}\|G_{ki}(W_k^{n+1},W_i)+\frac{p_{ik}^n}{\rho_{ik}}\|^2 \\ &=\frac{\rho_{ik}}{2}\|D_{ki}(W_k^{n+1})+D_{ik}(W_i)+\frac{p_{ik}^n}{\rho_{ik}}\|^2 \\ &=\frac{\rho_{ik}}{2}\|(D_{ki}(W_k^n)-r_{ki}^n)+D_{ik}(W_i)+\frac{p_{ik}^{n-1}+\rho_{ik}r_{ki}^n}{\rho_{ik}}\|^2 \\ &=\frac{\rho_{ik}}{2}\|G_{ki}(W_k^n,W_i)+\frac{p_{ik}^{n-1}}{\rho_{ik}}\|^2.\end{aligned} \tag{21}$$

We use (19c) and (13) in the second equality in (21), with node k being the tail node as $\{k,i\}\in\hat{\mathcal{E}}$ [2]. Note that (21) holds only for $n\geq 1$ because (19c) holds only for $n\geq 0$. Due to (21), optimization (20) for node $i\in\mathbb{K}_0$ does not change with respect to the iteration number as long as $n\geq 1$. Hence,

$$D_{ik}(W_i^{n+1}-W_i^n)=0 \text{ and } r_{ki}^{n+1}=0, \tag{22}$$

as a result of Eq. (19c), for all $i\in\mathbb{K}_0$ and $\forall n\geq 1$. Notice that we have (22) only for $i\in\mathbb{K}_0$ because if $i\notin\mathbb{K}_0$, (20) includes additional terms for $\{i,k\}\in\hat{\mathcal{E}}$, for which results similar to (21) are not available.

Next consider the subgraph of $\hat{\mathcal{G}}$ induced by the set of vertices $\mathcal{N}\setminus\mathbb{K}_0$, denoted as $\hat{\mathcal{G}}[\mathcal{N}\setminus\mathbb{K}_0]$. The graph $\hat{\mathcal{G}}[\mathcal{N}\setminus\mathbb{K}_0]$ may be composed of several disconnected subgraphs in general. Every subgraph of $\hat{\mathcal{G}}[\mathcal{N}\setminus\mathbb{K}_0]$ has at least one sink node because $\hat{\mathcal{G}}[\mathcal{N}\setminus\mathbb{K}_0]$ remains acyclic. Let $\mathbb{K}_1$ be the set of sink nodes of $\hat{\mathcal{G}}[\mathcal{N}\setminus\mathbb{K}_0]$. The optimization on the primal variables of $i\in\mathbb{K}_1$ can be written as

$$\begin{aligned}W_i^{n+1}=&\operatorname{argmin}_{W_i\in\mathcal{W}_i} f_i(W_i) \\ &+\sum_{\{i,k\}\in\hat{\mathcal{E}},k\in\mathbb{K}_0}\frac{\rho_{ik}}{2}\|G_{ik}(W_i,W_k^n)+\frac{p_{ik}^n}{\rho_{ik}}\|^2 \\ &+\sum_{\{k,i\}\in\hat{\mathcal{E}}}\frac{\rho_{ik}}{2}\|G_{ki}(W_k^{n+1},W_i)+\frac{p_{ik}^n}{\rho_{ik}}\|^2.\end{aligned} \tag{23}$$

We can show that the quadratic-cost functional terms associated with $\{k,i\}\in\hat{\mathcal{E}}$ do not change for $n\geq 1$ because of Eq. (21), which holds for any $\{k,i\}\in\hat{\mathcal{E}}$. The quadratic-cost functional terms associated with links $\{i,k\}\in\hat{\mathcal{E}}$, for $k\in\mathbb{K}_0$, do not change either for $n\geq 1$ because Eq. (22) holds for every $k\in\mathbb{K}_0$ for $n\geq 1$. Therefore, we have Eq. (22) satisfied for all $i\in\mathbb{K}_1$ for every $n\geq 2$. With similar arguments for the subgraph $\hat{\mathcal{G}}[\mathcal{N}\setminus(\mathbb{K}_0\cup\mathbb{K}_1)]$, we can conclude that Eq. (22) holds for nodes in $\mathbb{K}_2$ within finite iterations, where $\mathbb{K}_2$ is the set of sink nodes of $\hat{\mathcal{G}}[\mathcal{N}\setminus(\mathbb{K}_0\cup\mathbb{K}_1)]$. Repeating the induction, it follows that after a finite number of iterations, Eq. (22) holds for all $i\in\mathcal{N}$. The number of iterations required for Eq. (22) to hold is the diameter of the directed graph $\hat{\mathcal{G}}(\mathcal{N},\hat{\mathcal{E}})$.

In the reasoning above, we have shown that there exists $n_0<\infty$ such that

$$\|r_{ik}^{n_0+1}\|=\|D_{ki}(W_k^{n_0+1}-W_k^{n_0})\|^2=0, \tag{24}$$

holds for all $\{i,k\}\in\hat{\mathcal{E}}$. Notice that $r^{n_0+1}=0$ indicates the solution of (A.39) for iteration n_0+1 is $W_{\mathcal{N}}^\star$. Thus, $W_{\mathcal{N}}^{n_0+1}=W_{\mathcal{N}}^\star$, and $p^{n_0+1}=p^{n_0}=p^\star$ because of strong duality. Therefore, we have $V^{n_0+1}=V(W_{\mathcal{N}}^{n_0+1},p^{n_0+1})=V(W_{\mathcal{N}}^\star,p^\star)=0$. By definition of $\mathcal{I}_0$, V remains constant throughout the trajectory starting from $(\mathcal{W}_{\mathcal{N}}^0,p^0)$. Therefore, we can conclude that the largest invariant set is

$$\{(W_{\mathcal{N}},p)\mid V(W_{\mathcal{N}},p)=0\}. \tag{25}$$

By LaSalle theorem, we next conclude that as $n\to\infty$

$$D_{ki}(W_k^n)\to D_{ki}(W_k^\star),\quad p_{ik}^n\to p_{ik}^\star, \tag{26}$$

for all $\{i,k\}\in\hat{\mathcal{E}}$. Combining (26) and the optimization steps in (12), we also have that for all $\{i,k\}\in\hat{\mathcal{E}}$

$$D_{ik}(W_i^n)\to D_{ik}(W_i^\star), \tag{27}$$

as $n\to\infty$. The equalities (26) and (27) hold if and only if $(W_{\mathcal{N}}^n,p^n)\to(W_{\mathcal{N}}^\star,p^\star)$, completing the proof. □

Remark IV.4. ***(Convergence rate of Algorithm 1).*** Characterizing the convergence rate of the scheduled-asynchronous algorithm is challenging. The main reason for this is the fact that, given the particular characteristics of the design of Algorithm 1, the proof of Theorem IV.3 identifies a LaSalle function, and not a Lyapunov one. This is in contrast with the convergence proof of ADMM, cf., [16], where the availability of a Lyapunov function makes the convergence analysis and the rate characterization easier. If the network topology is bipartite, Algorithm 1 reduces to ADMM and hence shares the $O(1/n)$ convergence rate established in [16]. Based on this and our simulation results, we conjecture that Algorithm 1 also has $O(1/n)$ convergence rate with a smaller constant. □

Remark IV.5. ***(Heterogeneous selection of parameter ρ).*** The convergence rate of Algorithm 1 depends on the parameter ρ which weighs how the errors in the constraint satisfaction affect the evolution of the dual variables. Our simulation studies show that selecting ρ_{ik} heterogeneously can improve the convergence rate dramatically. In particular, we observe that choosing ρ_{ik} proportionally to the norm of the complex line admittance of edge $\{i,k\}$, $\|y_{ik}\|$ improves the convergence rate compared to selecting a uniform ρ, i.e., $\rho_{ik}=\rho_0,\forall\{i,k\}\in\mathcal{E}$. The intuition behind this fact is that the network is more sensitive to the perturbation on the edges with larger admittances. To justify this point, consider an edge where G_{ik} in (9) is not zero,

$$G_{ik}(W_i^t,W_k^t)=\delta_{ik}^t,\ \forall\{i,k\}\in\hat{\mathcal{E}}, \tag{28}$$

with $0\neq\delta_{ik}^t\in\mathbb{R}^4$. Due to (28), $W_{\mathcal{N}}^t$ is not a feasible solution of **(P2)**. Consider then the virtual network in Fig. 1, where nodes i and k are not physically connected and only the communication between them remains. Node i is connected to a virtual node k_v with admittance y_{ik}, and the same applies to k and i_v. The communication between i and k have W_i and W_k satisfy $G_{ik}(W_i^t,W_k^t)=\delta_{ik}^t$. Let $V_{i\hat{i}}$ and $V_{i\hat{k}}$, respectively, denote the copy of the voltages of bus i and k contained in W_i. Assign to nodes k_v and i_v the voltages $V_{i\hat{k}}$ and $V_{k\hat{i}}$,

[2]Recall that p_{ik}^n and ρ_{ik} are non-directional, i.e., $p_{ik}^n=p_{ki}^n$ and $\rho_{ik}^n=\rho_{ki}^n$. However, $r_{ik}^n=G_{ik}(W_i^n,W_k^n)=D_{ik}(W_i^n)+D_{ki}(W_k^n)$ are directional.

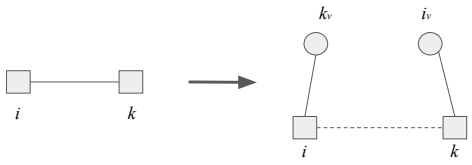


Fig. 1. Illustration of the virtual subnetwork with perturbation on the constraint $G_{ik}(W_i, W_k) = 0$. The solid lines represent the physically connected edge with communication. The dash line corresponds to a communication link.

respectively. In this way, the virtual network has the power flow from k_v to i given by

$$V_{i\hat{k}}\Big(y_{ik}(V_{i\hat{k}} - V_{i\hat{i}})\Big)^*. \tag{29}$$

On the other hand, the power flow from k to i_v is

$$V_{k\hat{k}}\Big(y_{ik}(V_{k\hat{k}} - V_{k\hat{i}})\Big)^*. \tag{30}$$

The copies of voltages of nodes i and k are related by

$$V^t_{i\hat{k}} = V^t_{k\hat{k}} + \hat{\delta}^t_{ik,k}, \quad V^t_{i\hat{i}} = V^t_{k\hat{i}} + \hat{\delta}^t_{ik,i}, \tag{31}$$

where $\hat{\delta}^t_{ik,k} = \sqrt{|\delta^t_{ik}(1)|}\exp(j\angle(\delta^t_{ik}(3) + j\delta^t_{ik}(4)))$ and $\hat{\delta}^t_{ik,i} = \sqrt{|\delta^t_{ik}(2)|}$ (assume $\angle V_i = 0$ without loss of generality). If $\delta^t_{ik} = 0$, then the values of (29) and (30) are the same because $V_{i\hat{k}} = V_{k\hat{k}}$ and $V_{i\hat{i}} = V_{k\hat{i}}$. Therefore, the virtual network is equivalent to the original one if $\delta^t_{ik} = 0$. On the other hand, if the first two elements of δ^t_{ik} are non-zero, then $V^t_{i\hat{k}} \neq V^t_{k\hat{k}}$ and $V^t_{i\hat{k}} \neq V^t_{k\hat{i}}$ according to (31). In such case, the virtual network is not equivalent to the original one. Furthermore, we observe that for a given non-zero δ^t_{ik}, the discrepancy between Eq. (29) and (30) is proportional to y_{ik}. This discrepancy has the interpretation of a perturbation on the apparent power flow for edge $\{i, k\}$. □

V. Directed Graph Design

Here, we first describe how the average time between two consecutive iterations under the scheduled-asynchronous algorithm for a node depends on the diameter of the orientation of the network graph. Motivated by this observation, we set out to find an acyclic orientation with minimal diameter in a distributed fashion. In general, finding such an orientation is equivalent to finding the chromatic number, which is NP-hard. However, exploiting the planar property of many power networks, we develop a distributed algorithm to determine an orientation with small diameter.

A. Relationship Between Graph Coloring and Diameter

The updating sequence of the scheduled-asynchronous algorithm depends on the orientation $\hat{\mathcal{G}} = (\mathcal{N}, \hat{\mathcal{E}})$. To see this, consider a path in $\hat{\mathcal{G}}$, with i being the initial node and k the end node. The update of k requires that all other nodes in the path finish at least an update before it, which are processed in sequence starting from node i. The "waiting time" of node k associated to this path corresponds therefore to its length. Since the diameter of the orientation bounds the length of all the paths, this justifies finding an acyclic orientation which the smallest diameter. We formalize this problem next. Let $\Omega(\mathcal{G})$ be the collection of all acyclic orientations of $\mathcal{G}$. We denote by $\mathcal{G}_\omega$ the directed graph corresponding to $\omega \in \Omega(\mathcal{G})$. The problem we aim to solve is

$$\omega^\star = \arg\min_{\omega\in\Omega(\mathcal{G})} \big(\max_{h\in\mathcal{P}_\omega} |h|\big), \tag{32}$$

where $\mathcal{P}_\omega$ is the set of paths in $\mathcal{G}_\omega$, and $|h|$ is the length of the path h. The optimization (32) is directly related to the classical problem of finding the chromatic number $\mathcal{X}(\mathcal{G})$ of an undirected graph $\mathcal{G}$ [28]. In fact, one has

$$\mathcal{X}(\mathcal{G}) = 1 + \min_{\omega\in\Omega(\mathcal{G})} \big(\max_{h\in\mathcal{P}_\omega} |h|\big). \tag{33}$$

In general, computing $\mathcal{X}(\mathcal{G})$ is NP-hard, see e.g. [29]. There are only approximate algorithms to find a solution, see for example [30]. However, for planar graphs, the chromatic number is upper bounded by four [31] and, furthermore, there exists a quadratic-time algorithm to find a graph-coloring with four colors, cf. [32]. Fortunately, the following assumption holds for most electrical networks.

Assumption 1. ***(Planar network topology [33]).*** *Electrical networks have simple planar network topology.*

If a centralized entity has information on the network topology, then the algorithm in [32] can be run to assign a number (or color) to every agent. This procedure might be problematic in large-scale scenarios, specially in the presence of plug-and-play devices that easily change the network topology. This motivates our design of a distributed algorithm.

B. Distributed Orientation Computation with Small Diameter

The following distributed algorithm from [34] finds an orientation of an arbitrary planar graph with diameter bounded by five (or equivalently $\mathcal{X}(\mathcal{G}) \leq 6$):

1) find an acyclic orientation in a distributed way such that every node has out-degree at most five.
2) every node chooses a color that is different from all the out-neighboring nodes defined in step 1).

Note that any ordering of the colors induces an orientation of the graph, which is furthermore acyclic, cf. Remark IV.1. The algorithm above is simple to implement but may be conservative for electrical networks because, in general, the degree of most nodes is far less than six. In fact, empirical studies [35], [36] have shown that electrical networks have a degree distribution that follows the exponential distribution. Since the degree takes integer values, it is more appropriate to characterize the degree distribution with the geometric distribution – the discrete analogy of the exponential distribution.

Assumption 2. ***(Geometric degree distribution).*** *In electrical networks, the number of nodes with degree at least $d_0 \in \mathbb{N}$ satisfies*

$$\mathit{Prob}(d = d_0) = \lambda(1-\lambda)^{d_0}, \tag{34}$$

with parameter $0 \leq \lambda \leq 1$.

Most nodes have small degree due to (34). According to [35], [37], the average degree of empirical electrical networks is between 2 and 3. Many electrical networks, as a result, have the chromatic number far less than six. For example, IEEE 14, 30 and 57 bus test cases have $\mathcal{X}(\mathcal{G}) = 3$. Therefore, it is conceivable that one can find an orientation with diameter less than five. Motivated by this observation, we modify the algorithm in [34] to exploit the geometric degree distribution property.

We propose Algorithm 2 to find a preliminary orientation. The idea is to first try to find an orientation with the out-degree of all nodes no bigger than 2. If this is not possible, then the strategy searches instead for an orientation with one more out-degree. If necessary, this process is repeated until Algorithm 2 eventually finds an orientation where the out-degree of all nodes is at most five.

Algorithm 2

Initialize:
  For all $i \in \mathcal{N}$, set η_i s.t. $\eta_i \neq \eta_k \ \forall \{i,k\} \in \mathcal{E}$
  $m_i \leftarrow 0, \quad \overline{h}_i \leftarrow \overline{h}^0$
For every $i \in \mathcal{N}$**:**
  Update $\mathcal{N}_{\eta_i} := \{k \in \mathcal{N}_i \,|\eta_k > \eta_i\}$
  If $|\mathcal{N}_{\eta_i}| \geq \overline{h}_i$
    If $\overline{h}_i = 6$ **or** $m_i \leq \overline{m}$
      $\eta_i = \max_{k \in \mathcal{N}_{\eta_i}} \eta_k + 1$
      $m_i \leftarrow m_i + 1$
      Send η_i to neighbors $\mathcal{N}_i$
    else if $m_i > \overline{m}$ **and** $\overline{h}_i < 6$
      $m_i \leftarrow 0, \quad \overline{h}_i \leftarrow \overline{h}_i + 1$
    end
  end

We next explain the pseudocode of Algorithm 2. Every node i starts with an initial number η_i such that $\eta_i \neq \eta_k$, for all $\{i,k\} \in \mathcal{E}$. These numbers induce an acyclic orientation by declaring that node i is a tail of $\{i,k\}$ if $\eta_i < \eta_k$, cf. Remark IV.1. Under the algorithm, every node i recursively updates η_i if its current out-degree is bigger than or equal to a number $\overline{h}_i$, initially set to $\overline{h}^0 = 2$, for all i. The update of η_i follows a simple rule to choose a number bigger than η_k, for all $k \in \mathcal{N}_i$. In this way, node i becomes a sink node with out-degree zero in the new induced orientation. Node i then sends η_i to all $k \in \mathcal{N}_i$ for them to recompute their out-degree. If all nodes do not require any further update in their η, Algorithm 2 converges to an orientation in which all nodes have an out-degree of at most two. If, instead, some nodes require an update for more than $\overline{m} > 1$ times, then the strategy has these nodes increase its $\overline{h}_i$ by one (since an orientation with out-degree less than two might not exist). Any node i that again updates its variable $\overline{m}$ times has $\overline{h}_i$ increased in a similar way. The procedure repeats until every node stops updating.

The following result establishes the convergence properties of Algorithm 2.

Proposition V.1. ***(Convergence of Algorithm 2).*** *Algorithm 2 converges with* $\overline{h}_i \leq 6$ *for all* $i \in \mathcal{N}$ *in a finite number of iterations.*

Proof. Because $\overline{m} < \infty$, it is sufficient to show that every node stops updating η_i in a finite number of steps if $\overline{h}_i = 6$, $\forall i \in \mathcal{N}$. Since every simple planar graph has at least one node with degree strictly less than six, this node stops updating η_i as $|\mathcal{N}_{\eta_i}| \leq 5 < \overline{h}_i$. We then consider the subgraph of $\mathcal{G}$ induced from $\mathcal{N} \setminus \{i\}$. This subgraph is also planar so we can find another node that will stop updating η after a finite number of steps. The result follows by repeating this argument. □

With the orientation induced by the η_i variables resulting from the algorithm, one can find a coloring of the graph with $\overline{h} := \max_{i \in \mathcal{N}} \overline{h}_i$ colors. Given the set of numbers $C := \{1, 2, \ldots, \overline{h}\}$, Algorithm 3, from [34], assigns a number in C to each $i \in \mathcal{N}$. The resulting acyclic orientation has diameter at most $\overline{h} - 1$ due to (33).

Algorithm 3

Initialize:
  For all $k \in \mathcal{N}$, set $\zeta_k \in \{1, 2, \ldots, \overline{h}_k\}$
For every $k \in \mathcal{N}$**,**
  Update $\mathcal{N}_{\zeta_k} := \{i \in \mathcal{N}_k \,|\zeta_i = \zeta_k \text{ and } \eta_i > \eta_k\}$
    If $\mathcal{N}_{\zeta_k} \neq \emptyset$
      Choose $\zeta_k \in \{1, 2, \ldots, \overline{h}_k\} \setminus \{\zeta_i | i \in \mathcal{N}_{\zeta_k}\}$
    end
  Send ζ_k to $i \in \mathcal{N}_k$

In Algorithm 2, setting $\overline{m}$ too small can be conservative because the strategy gives up too early in finding orientations with small diameter by rapidly increasing $\overline{h}_k$ toward six. On the other hand, setting $\overline{m}$ too large slows down the convergence. The challenge lies then in characterizing the value $\overline{m}$ that strikes a balance between maximizing the convergence rate and minimizing the average path length of the resulting orientation. The optimal choice of $\overline{m}$ depends on the size of the network and the constant λ in the geometric degree distribution. Theorem V.2 illustrates how $\overline{m}$ is related to these factors.

Theorem V.2. ***(On the degree of subgraphs and convergence of Algorithm 2).*** *If* $\overline{h}_i^0 = c_0 < 6$, *for all* $i \in \mathcal{N}$ *and* $\overline{m} = \infty$, *then Algorithm 2 converges if and only if every vertex-induced subgraph of* $\mathcal{G}$ *has at least one node with degree less than* c_0.

Proof. We first show the implication from right to left. We term every node k with degree less than c_0 "stable", because it will not update its η_k regardless of the change of η of any other node. Let $\mathcal{S}_1$ be the set of stable nodes of $\mathcal{G}$. We then consider the induced subgraph $\mathcal{G}[\mathcal{S}_2]$, where $\mathcal{S}_2 = \mathcal{N} \setminus \mathcal{S}_1$. The graph $\mathcal{G}[\mathcal{S}_2]$ also has at least one node with degree less than c_0. Again, nodes in $\mathcal{G}[\mathcal{S}_2]$ with degree less than c_0 are called "stable" because they will not change their η in response to the change of η of any other node in $\mathcal{S}_2$. Every stable node i in $\mathcal{G}[\mathcal{S}_2]$ updates η_i at most once. The reason of the single update is the following. First, a stable node i in $\mathcal{S}_2$ is

connected to at least one node in $\mathcal{S}_1$. Otherwise, node i is in $\mathcal{S}_1$. Hence, once a stable node $i \in \mathcal{S}_2$ updates η_i, its out-degree, $|\mathcal{N}_{\eta_i}|$, is at most $\overline{h}_i - 1$ and the node will not update η_i again. We can reason with $\mathcal{S}_3, \mathcal{S}_4, \cdots, \mathcal{S}_s$ in a similar way until $\cup_{\alpha=1,2,\cdots,s} \mathcal{S}_\alpha = \mathcal{N}$ (note that $s < \infty$ because $N < \infty$) and conclude that Algorithm 2 converges in finite time.

Next, we show the implication from left to right. If there exists a vertex-induced subgraph with all nodes having degree at least c_0, then we can show that at least one node in the subgraph updates η infinitely often. Recall that every acyclic orientation has a source node. The source of the subgraph updates its η_i because $|\mathcal{N}_{\eta_i}| \geq c_0$. After this update, the orientation of the subgraph remains acyclic because it is induced by the values of the variables η_i. As a consequence, at least one node of the subgraph becomes a source and makes an update. The sequence repeats for infinite times because there always exists one node in the subgraph with $|\mathcal{N}_{\eta_i}| \geq c_0$. Since the number of nodes is finite, there exists at least one node that updates η infinitely often. As a consequence, Algorithm 2 does not converge. □

Theorem V.2 provides insight into the selection of $\overline{m}$ in Algorithm 2. With the notation of the proof, if we set $\overline{m} = \infty$ and Algorithm 2 converges with $\overline{h}_i^0 = c_0 < 6$, for all $i \in \mathcal{N}$, the nodes in $\mathcal{S}_\alpha$ update their variables at least the number of times that nodes in $\mathcal{S}_\beta$ do, for $\beta < \alpha$, because $\mathcal{S}_\alpha$ becomes "stable" after $\mathcal{S}_\beta$ does. If, instead, we choose a finite $\overline{m}$, then only a subset of nodes of $\mathcal{N}$ keep their variable $\overline{h}_i$ non-increasing, while the remaining nodes are forced to increase it, resulting in a more conservative upper bound of their out-degrees. The number of updates required for the last node being stable highly depends on the network topology. If λ is large, then most nodes have degree one or two, and $\mathcal{S}_1$ contains most nodes in $\mathcal{N}$. In this case, we can expect $\cup_{\alpha=1,\cdots,s} \mathcal{S}_\alpha = \mathcal{N}$ with a small s, so selecting a small $\overline{m}$ still provides a small $\overline{h}$ with fast convergence time. Our experience shows that selecting $\overline{m}$ around 10 is sufficient to yield an optimal coloring for electrical networks. The number may increase for some large networks that involve thousands of nodes.

VI. Simulations

Here, we validate the performance of the scheduled-asynchronous algorithm over the six bus test cases in [38], IEEE 14, 30, and 57 bus test cases. We first use Algorithms 2-3 to find an acyclic orientation of each test case, cf. Table I. This results in orientations with diameter two for all test cases except the six bus test cases, with diameter three.

TABLE I
Simulation parameters and results of Algorithm 2.

	$\overline{m}$	$\overline{h}^0$	Final $\overline{h}$	Diam. of acyc. ori.
6 bus	10	2	4	3
14 bus	10	2	3	2
30 bus	10	2	3	2
57 bus	10	2	3	2

Figure 2 shows the convergence of Algorithm 1 for the various test cases. The stopping criteria is $\gamma_i < 10^{-4}$ for all $i \in \mathcal{N}$. The horizontal axis in the plots is the global iteration number, not the iteration per bus (the number of iterations per bus is roughly the global iteration number divided by the diameter of the acyclic orientation). Table II and Figure 2 show that the weighted selection of ρ, discussed in Remark IV.5, leads to a much faster convergence than the uniform $\rho_{ik} = \rho_0$, for all $\{i, k\} \in \mathcal{E}$. We have also simulated the case with unreliable communication, where every link has a 10% probability of packet drop if the previous communication was successful. Table II shows that the Algorithm 1 still converges, albeit requiring more iterations than the case with no packet drops. Figure 3 illustrates the transient behavior for the packet drop case.

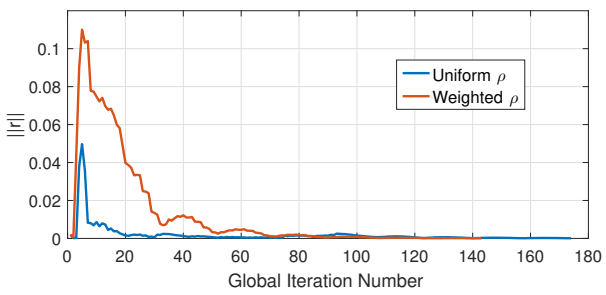
TABLE II
Number of iterations needed for $\gamma_i < 10^{-4}, \forall i \in \mathcal{N}$.

	Iter./bus	Iter./bus pack. drop.	Iter./bus weighted ρ	ρ_0
6 bus	62	65	50	700
14 bus	110	127	57	700
30 bus	140	260	82	700
57 bus	1520	1810	660	1000

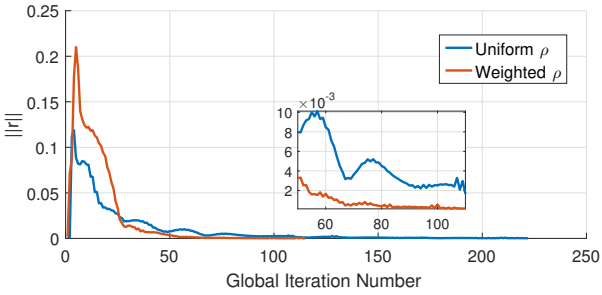
Finally, we evaluate the impact of the acyclic orientation in the algorithm convergence in Figure 4. We simulate how Algorithm 1 converges for the IEEE 14 and 30 bus test cases with an arbitrarily chosen acyclic orientation instead of the one obtained from the execution of Algorithms 2-3. Compared with Figure 2(b)-(c), one can observe that the algorithm requires many more iterations to converge and that $\|r\|$ increases dramatically at several iterations. These abrupt changes can also be explained as the result of several long paths in the digraph. In fact, recall that every node in a path makes one iteration before transferring its update to the following node. Starting with one of the terminal nodes in a long path, the update propagates through many nodes before reaching the other terminal node for its next update, at which point it may introduce a big change on its decision variables resulting in a dramatic change on $\|r\|$. Therefore, the selection of an acyclic orientation with small diameter has the added benefit, beyond reducing the average waiting time per iteration, of resulting in less abrupt changes in the algorithm execution.

VII. Conclusions

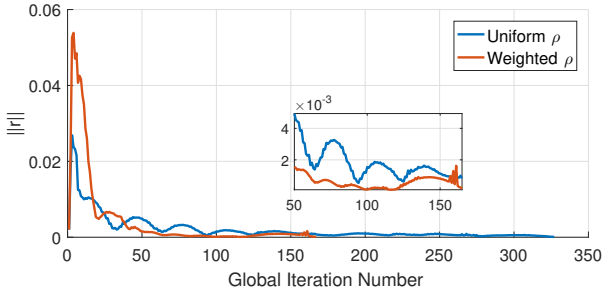
We have designed the scheduled-asynchronous algorithm to solve SDP convexified OPF problems in a distributed way. Under the proposed strategy, every pair of nodes connected in the electrical network update their local variables in an alternating fashion and the ordering of node updates is encoded by an orientation of the network. We have established the algorithm convergence to the optimizer when the orientation is acyclic and shown how, when the network is bipartite, the strategy corresponds to the ADMM scheme and has therefore $O(1/n)$ convergence rate. The convergence result does not require strong convexity of the cost function, which makes



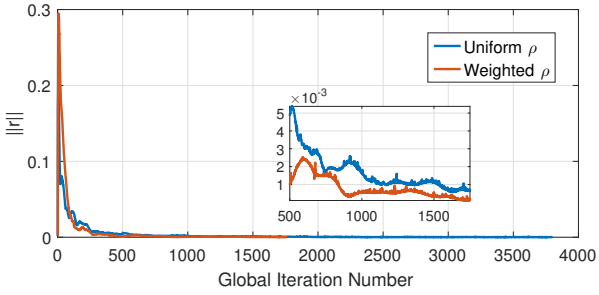
(a) Six bus test case



(b) IEEE 14 bus test case

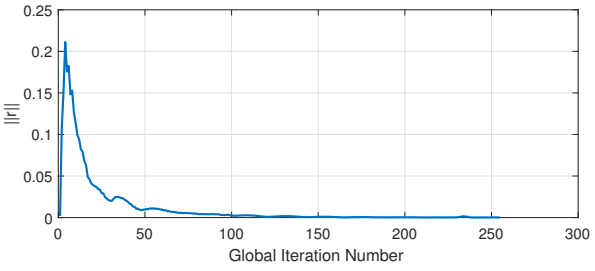


(c) IEEE 30 bus test case

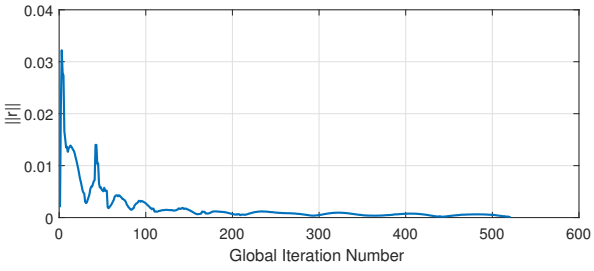


(d) IEEE 57 bus test case

Fig. 2. Convergence of the scheduled-asynchronous algorithm for various test cases: (a) six bus test case in [38], (b) IEEE 14, (c) IEEE 30, and (d) IEEE 57.

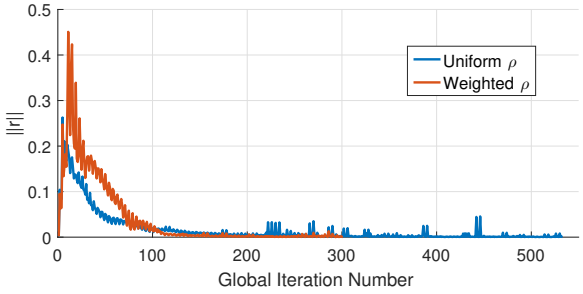


(a) IEEE 14 bus test case

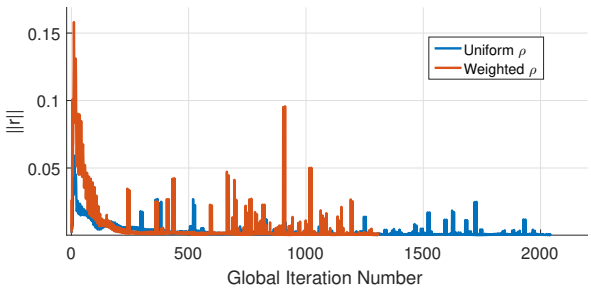


(b) IEEE 30 bus test case

Fig. 3. Convergence of the scheduled-asynchronous algorithm under packet drops.



(a) IEEE 14 bus test case



(b) IEEE 30 bus test case

Fig. 4. Convergence of the scheduled-asynchronous algorithm with arbitrary acyclic orientations of the network graph.

it especially suitable in OPF applications. To improve the algorithm convergence rate, we have introduced a distributed graph coloring algorithm that finds an acyclic orientation with small diameter for networks with geometric degree distribution. Future work will explore the characterization of the convergence rate for general network topologies, the optimal selection of the algorithm parameters, and the formal analysis of the algorithm robustness properties observed in simulation.

ACKNOWLEDGMENTS

This research was supported by the ARPA-e Network Optimized Distributed Energy Systems (NODES) program, Cooperative Agreement DE-AR0000695.

REFERENCES

[1] C.-Y. Chang, J. Cortés, and S. Martínez, "A scheduled-asynchronous distributed optimization algorithm for the optimal power flow problem," Seattle, WA, May 2017, pp. 3968–3973.
[2] 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," vol. 14, no. 1, pp. 96–104, 1999.
[3] V. Ajjarapu and C. Christy, "The continuation power flow: a tool for steady state voltage stability analysis," vol. 7, no. 1, pp. 416–423, 1992.
[4] M. Abido, "Optimal power flow using particle swarm optimization," *International Journal of Electrical Power & Energy Systems*, vol. 24, no. 7, pp. 563–571, 2002.
[5] A. J. Conejo and J. A. Aguado, "Multi-area coordinated decentralized DC optimal power flow," vol. 13, no. 4, pp. 1272–1278, 1998.
[6] P. N. Biskas, A. G. Bakirtzis, N. I. Macheras, and N. K. Pasialis, "A decentralized implementation of DC optimal power flow on a network of computers," vol. 20, no. 1, pp. 25–33, 2005.
[7] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," vol. 27, no. 1, pp. 92–107, 2012.
[8] B. Zhang, "Control and optimization of power systems with renewables: Voltage regulation and generator dispatch," Ph.D. dissertation, University of California, Berkeley, 2013.
[9] D. K. Molzahn, "Application of semidefinite optimization techniques to problems in electric power systems," Ph.D. dissertation, University of Wisconsin-Madison, 2013.
[10] D. K. Molzahn, C. Josz, I. A. Hiskens, and P. Panciatici, "A Laplacian-based approach for finding near globally optimal solutions to OPF problems," vol. 32, no. 1, pp. 305–315, 2017.
[11] R. Madani, M. Ashraphijuo, and J. Lavaei, "Promises of conic relaxation for contingency-constrained optimal power flow problem," 2014, pp. 1064–1071.
[12] P. Biskas, A. Bakirtzis, N. Macheras, and N. Pasialis, "A decentralized implementation of DC optimal power flow on a network of computers," vol. 20, no. 1, pp. 25–33, 2005.
[13] A. Lam, B. Zhang, and N. T. David, "Distributed algorithms for optimal power flow problem," 2012, pp. 430–437.
[14] E. Dall'Anese, H. Zhu, and G. B. Giannakis, "Distributed optimal power flow for smart microgrids," vol. 4, no. 3, pp. 1464–1475, 2013.
[15] D. Jakovetić, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," vol. 59, no. 5, pp. 1131–1146, 2014.
[16] 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.
[17] S. Lee and A. Nedić, "Asynchronous gossip-based random projection algorithms over networks," vol. 61, no. 4, pp. 953–968, 2016.
[18] L. Mazzarella, A. Sarlette, and F. Ticozzi, "A new perspective on gossip iterations: from symmetrization to quantum consensus," Florence, Italy, 2013.
[19] D. Shah, "Gossip algorithms," *Foundations and Trends in Networking*, vol. 3, no. 1, pp. 1–125, 2009.
[20] 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.
[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.
[22] S. Bose, D. F. Gayme, S. H. Low, and K. M. Chandy, "Optimal power flow over tree networks," 2011, pp. 1342–1348.
[23] C.-Y. Chang and W. Zhang, "On near and exact optimal power flow solutions for microgrid applications," Las Vegas, NV, 2016.
[24] S. Bose, S. H. Low, T. Teeraratkul, and B. Hassibi, "Equivalent relaxations of optimal power flow," vol. 60, no. 3, pp. 729–742, 2015.
[25] C.-Y. Chang and W. Zhang, "General opf problems with reactive power costs: A distributed sdp approach," *arXiv preprint arXiv:1612.04508*, 2016.
[26] 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.
[27] 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.
[28] 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.
[29] A. Sánchez-Arroyo, "Determining the total colouring number is NP-hard," *Discrete Mathematics*, vol. 78, no. 3, pp. 315–319, 1989.
[30] D. Corneil and B. Graham, "An algorithm for determining the chromatic number of a graph," vol. 2, no. 4, pp. 311–318, 1973.
[31] 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.
[32] ——, "Efficiently four-coloring planar graphs," in *Proceedings of the twenty-eighth annual ACM symposium on theory of computing*, 1996, pp. 571–575.
[33] K. C. Sou, "A branch-decomposition approach to power network design," 2016, pp. 6483–6488.
[34] S. Ghosh and M. H. Karaata, "A self-stabilizing algorithm for coloring planar graphs," *Distributed Computing*, vol. 7, no. 1, pp. 55–59, 1993.
[35] P. Hines, S. Blumsack, E. C. Sanchez, and C. Barrows, "The topological and electrical structure of power grids," in *43rd Hawaii International Conference on System Sciences (HICSS)*, 2010, pp. 1–10.
[36] R. Albert, I. Albert, and G. L. Nakarado, "Structural vulnerability of the north american power grid," *Physical review E*, vol. 69, no. 2, p. 025103, 2004.
[37] W. Deng, W. Li, X. Cai, and Q. A. Wang, "The exponential degree distribution in complex networks: Non-equilibrium network theory, numerical simulation and empirical data," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 8, pp. 1481–1485, 2011.
[38] A. J. Wood and B. Wollenberg, *Power generation operation and control-2nd edition*. New York, USA: John Wiley and Sons, 1996, vol. 3.

APPENDIX

A. Auxiliary Results for the Proof of Theorem IV.3

We gather here various auxiliary results used in the proof of Theorem IV.3. Our first result shows that two classes of convex optimization problems with separable cost functions have the same optimal solution.

Lemma A.1. *Let $\phi, \varphi : \mathcal{H}^n \to \mathbb{R}$ be convex and differentiable, and let $\mathcal{X}$ be a convex set. Then*

$$X^\star \in \operatorname{argmin}_{X\in\mathcal{X}} \phi(X) + \varphi(X), \tag{A.35a}$$

$$\Leftrightarrow X^\star \in \operatorname{argmin}_{X\in\mathcal{X}} \phi(X) + \langle \bigtriangledown\varphi(X^\star), X\rangle. \tag{A.35b}$$

Proof. The necessary and sufficient condition for $X^\star$ being the optimal solution of optimization (A.35a) is that

$$\left\langle \bigtriangledown \big(\phi(X^\star) + \varphi(X^\star)\big), X - X^\star \right\rangle \geq 0,$$

for all $X \in \mathcal{X}$. We can rewrite the condition above as

$$\left\langle \bigtriangledown \phi(X^\star), X - X^\star \right\rangle + \left\langle \bigtriangledown \varphi(X^\star), X - X^\star \right\rangle \geq 0, \tag{A.36}$$

for all $X \in \mathcal{X}$. Eq. (A.36) is also the optimality condition of optimization Eq. (A.35b), which completes the proof. □

We use Lemma A.1 to establish two inequalities that will be employed to show that the function (15) is non-increasing along the algorithm executions. The notation we employ next is carried over from the proof of Theorem IV.3.

Lemma A.2. *Under the assumptions of Theorem IV.3, the following inequalities hold*

$$\sum_{i=1}^{N}\Big(f_i(W_i^\star)-f_i(W_i^{n+1})\Big)\leq {p^\star}^\top r^{n+1}, \tag{A.37a}$$

$$\sum_{i=1}^{N}\Big(f_i(W_i^{n+1})-f_i(W_i^\star)\Big)\leq -{p^{n+1}}^\top r^{n+1} + \sum_{\{i,k\}\in\hat{\mathcal{E}}}\rho_{ik}D_{ki}^\top(W_k^n-W_k^{n+1})D_{ik}(W_i^\star-W_i^{n+1}). \tag{A.37b}$$

Proof. **Eq. (A.37a).** Since **(P2)** is convex and Slater's condition holds, strong duality follows and the KKT conditions are necessary and sufficient for the optimal solution of (10). Strong duality together with the KKT conditions imply

$$W_{\mathcal{N}}^\star=\operatorname*{argmin}_{W_i\in\mathcal{W}_i}\sum_{i\in\mathcal{N}}f_i(W_i)+\sum_{\{l,k\}\in\hat{\mathcal{E}}}{p_{lk}^\star}^\top G_{lk}(W_l,W_k). \tag{A.38}$$

Since $W_{\mathcal{N}}^\star$ is the optimizer, using this inequality we deduce

$$\sum_{i=1}^{N}f_i(W_i^\star)+\sum_{\{l,k\}\in\hat{\mathcal{E}}}{p_{lk}^\star}^\top r_{lk}^\star \leq \sum_{i=1}^{N}f_i(W_i^{n+1})+\sum_{\{l,k\}\in\hat{\mathcal{E}}}{p_{lk}^\star}^\top r_{lk}^{n+1}.$$

Inequality (A.37a) follows by noting that $r_{lk}^\star=0$, for all $\{l,k\}\in\hat{\mathcal{E}}$.

Eq. (A.37b). We start by rewriting (12) with the number of iterations instead of the time index,

$$\begin{aligned}W_i^{n+1}=&\operatorname{argmin}_{W_i\in\mathcal{W}_i}f_i(W_i)\\&+\sum_{\{i,k\}\in\hat{\mathcal{E}}}\Big({p_{ik}^n}^\top G_{ik}(W_i,W_k^n)+\frac{\rho_{ik}}{2}\|G_{ik}(W_i,W_k^n)\|^2\Big)\\&+\sum_{\{k,i\}\in\hat{\mathcal{E}}}\Big({p_{ik}^n}^\top G_{ki}(W_k^{n+1},W_i)+\frac{\rho_{ik}}{2}\|G_{ki}(W_k^{n+1},W_i)\|^2\Big).\end{aligned} \tag{A.39}$$

The superscript of every variable in (A.39) represents the number of updates of the associated variable. We resort to Lemma A.1 to rewrite (A.39), viewing f_i as ϕ and grouping all the other objective functions as φ. In addition, W_i^{n+1} and W_i play the role of $X^\star$ and X in (A.35), respectively. We then have

$$\begin{aligned}W_i^{n+1}=&\operatorname{argmin}_{W_i\in\mathcal{W}_i}f_i(W_i)\\&+\sum_{\{i,k\}\in\hat{\mathcal{E}}}\Big({p_{ik}^n}^\top+\rho_{ik}G_{ik}^\top(W_i^{n+1},W_k^n)\Big)D_{ik}(W_i)\\&+\sum_{\{k,i\}\in\hat{\mathcal{E}}}\Big({p_{ik}^n}^\top+\rho_{ik}G_{ki}^\top(W_k^{n+1},W_i^{n+1})\Big)D_{ik}(W_i).\end{aligned}$$

With a slight abuse of notation, we denote by $G_{ki}^\top(\cdot,\cdot)\equiv G_{ki}(\cdot,\cdot)^\top$, the transpose of $G_{ki}(\cdot,\cdot)$ (similarly for the variables $D_{ki}^\top$). According to this equation, evaluating the objective function at $W_i^\star$ gives rise to a larger value than at W_i^{n+1}, and therefore,

$$\begin{aligned}\sum_{i=1}^{N}f_i(W_i^{n+1})-f_i(W_i^\star)\leq\sum_{i=1}^{N}\Bigg(&\sum_{\{i,k\}\in\hat{\mathcal{E}}}\Big({p_{ik}^n}^\top+\rho_{ik}G_{ik}^\top(W_i^{n+1},W_k^n)\Big)D_{ik}(W_i^\star-W_i^{n+1})\\&+\sum_{\{k,i\}\in\hat{\mathcal{E}}}{p_{ik}^{n+1}}^\top D_{ik}(W_i^\star-W_i^{n+1})\Bigg).\end{aligned} \tag{A.40}$$

Note that we used (13) to substitute $p_{ik}^n+\rho_{ik}G_{ki}(W_k^{n+1},W_i^{n+1})$ by p_{ik}^{n+1}. The term ${p_{ik}^n}^\top+\rho_{ik}G_{ik}^\top(W_i^{n+1},W_k^n)$ in (A.40) can be written as

$$\begin{aligned}&{p_{ik}^n}^\top+\rho_{ik}G_{ik}^\top(W_i^{n+1},W_k^n)\\=&{p_{ik}^n}^\top+\rho_{ik}\Big(D_{ik}^\top(W_i^{n+1})+D_{ki}^\top(W_k^n)\Big)\\=&{p_{ik}^n}^\top+\rho_{ik}\Big(D_{ik}^\top(W_i^{n+1})+D_{ki}^\top(W_k^{n+1})+D_{ki}^\top(W_k^n-W_k^{n+1})\Big)\\=&{p_{ik}^n}^\top+\rho_{ik}\Big(G_{ik}^\top(W_i^{n+1},W_k^{n+1})+D_{ki}^\top(W_k^n-W_k^{n+1})\Big)\\=&{p_{ik}^{n+1}}^\top+\rho_{ik}D_{ki}^\top(W_k^n-W_k^{n+1}).\end{aligned}$$

Using this equation in (A.40) and the fact that $G_{ik}(W_i^\star,W_k^\star)=0$, we obtain (A.37b). □