Distributed Optimal Power Flow Algorithm for Radial Networks, I: Balanced Single Phase Case

Qiuyu Peng and Steven H. Low, *Fellow, IEEE*

***Abstract*—The optimal power flow (OPF) problem determines a network operating point that minimizes a certain objective such as generation cost or power loss. Traditionally, OPF is solved in a centralized manner. With increasing penetration of renewable energy in distribution system, we need faster and distributed solutions for real-time feedback control. This is difficult due to the nonlinearity of the power flow equations. In this paper, we propose a solution for balanced radial networks. It exploits recent results that suggest solving for a globally optimal solution of OPF over a radial network through the second-order cone program relaxation. Our distributed algorithm is based on alternating direction method of multiplier (ADMM), but unlike standard ADMM-based distributed OPF algorithms that require solving optimization subproblems using iterative method, our decomposition allows us to derive closed form solutions for these subproblems, greatly speeding up each ADMM iteration. We illustrate the scalability of the proposed algorithm by simulating it on a real-world 2065-bus distribution network.**

***Index Terms*—Power distribution, distributed algorithms, nonlinear systems, power system control.**

I. Introduction

THE OPTIMAL power flow (OPF) problem seeks to optimize certain objective such as power loss and generation cost subject to power flow equations and operational constraints. It is a fundamental problem because it underlies many power system operations and planning applications such as economic dispatch, unit commitment, state estimation, stability and reliability assessment, volt/var control, demand response, etc. The continued growth of highly volatile renewable sources on distribution systems calls for real-time feedback control. Solving the OPF problems in such an environment has at least two challenges.

Manuscript received September 28, 2015; revised January 12, 2016; accepted March 14, 2016. Date of publication March 24, 2016; date of current version December 21, 2017. This work was supported in part by the Advanced Research Projects Agency-Energy under Grant DE-AR0000226, in part by the Los Alamos National Laboratory under Department of Energy Grant DE-AC52-06NA25396, in part by the Defense Threat Reduction Agency under Grant HDTRA 1-15-1-0003, and in part by the Skotech under Grant 1075-MRA. A preliminary version of this paper has appeared in [1]. Paper no. TSG-01233-2015.

Q. Peng is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: qpeng@caltech.edu).

S. H. Low is with the Department of Computing and Mathematical Sciences, and the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: slow@caltech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2016.2546305

First the OPF problem is hard to solve because of its nonconvex feasible set. Recently a new approach through convex relaxation has been developed. Specifically semidefinite program (SDP) relaxation [2] and second order cone program (SOCP) relaxation [3] have been proposed in the bus injection model, and SOCP relaxation has been proposed in the branch flow model [4], [5]. See the tutorial [6], [7] for further pointers to the literature. When an optimal solution of the original OPF problem can be recovered from any optimal solution of a convex relaxation, we say the relaxation is *exact*. For radial networks (whose graphs are trees), several sufficient conditions have been proved that guarantee SOCP and SDP relaxations are exact. This is important because almost all distribution systems are radial. Moreover some of these conditions have been shown to hold for many practical networks. In these cases we can rely on off-the-shelf convex optimization solvers to obtain a globally optimal solution for the nonconvex OPF problem.

Second, most algorithms proposed in the literature are centralized and meant for applications in today's energy management systems that centrally schedule a relatively small number of generators. In future networks that optimize the operation of a large number of intelligent endpoints, a centralized approach will not scale because of its computation and communication overhead. In this paper we address this challenge. Specifically we propose a distributed algorithm for solving the SOCP relaxation of OPF for balanced radial networks.

Various distributed algorithms have been developed to solve the OPF problem. Typically, the original OPF problem is decomposed into several local subproblems that can be solved simultaneously. Some distributed algorithms do not deal with the nonconvexity of the OPF problem, including [8], [9], which leverage method of multipliers, and [10], which is based on alternating direction method of multiplier (ADMM). However, the convergence of these algorithms is not guaranteed due to non-convexity of the problem. In contrast, algorithms for the convexified OPF problem (through either convex relaxation or linearization) are proposed to guarantee convergence, e.g., dual decomposition method [11], [12], auxiliary variable method [13], [14] and ADMM [15]–[17].

One of the key performance metrics of a distributed algorithm is the time of convergence (ToC), which depends on the number of iterations and the computation time to solve the subproblems in each iteration. To the best of our knowledge, all the distributed OPF algorithms in the literature rely on generic iterative optimization solvers, which are

computationally intensive, to solve the optimization subproblems. In this paper, we derive closed form solutions for all the optimization subproblems, which significantly reduce the computation time and thus improve ToC.

Specifically we develop a scalable distributed algorithm through decomposing the relaxed OPF problem into smaller subproblems based on ADMM. ADMM blends the decomposability of dual decomposition and superior convergence properties of the method of multipliers [18]. It has broad applications in different areas and particularly useful when the subproblems can be solved efficiently [19], for example when they admit closed form expressions, e.g., matrix factorization [20], image recovery [21].

The proposed algorithm has two advantages: 1) We provide a sufficient condition for the existence of closed form solutions to the optimization subproblems, thus eliminating the need for an iterative procedure to solve a SDP/SOCP problem for each ADMM iteration. 2) Communication is only required between adjacent buses.

We demonstrate the scalability of the proposed algorithms using a real-world network. In particular, we show that the algorithm converges within 0.6s for a 2,065-bus system on a MacBook computer. To show the superiority of closed form expression of each subproblems, we compare the computation time for solving a subproblem by our algorithm and an off-the-shelf optimization solver (SDPT3 [22]). Our solver requires on average 6.8×10^{-4}s for each ADMM iteration while SDPT3 requires on average 0.5s. Finally, we investigate the impact of network topology on the convergence rate and find that the convergence rate is mainly determined by the diameter[1] of the network.

The rest of the paper is structured as follows. The OPF problem and its second order cone program (SOCP) relaxation is defined in Section II. In Section III, we develop a distributed algorithm for a broad class of optimization problem and show the existence of closed form solution by applying it to the OPF problem. In Section IV, we demonstrate the scalability of the proposed algorithm by simulating it on large scale real-world distribution networks and comparing the computation time with a generic iterative solver (SDPT3). We conclude this paper in Section V.

II. Problem Formulation

In this section, we define the optimal power flow (OPF) problem on a balanced radial network and review how to solve it through SOCP relaxation.

We denote the set of complex numbers by $\mathbb{C}$, the set of n-dimensional complex numbers by $\mathbb{C}^n$. The hermitian transpose of a vector is denoted by $()^H$. To differentiate vector and scaler operations, the conjugate of a complex scaler is denoted by $()^*$. The inner product of two vectors $x, y \in \mathbb{C}^n$ is denoted by $\langle x, y\rangle := \mathbf{Re}(x^H y)$. The Euclidean norm of a vector $x \in \mathbb{C}^n$ is defined as $\|x\|_2 := \sqrt{\langle x, x\rangle}$.

[1]The diameter of a graph is defined as the maximum number of hops between any two nodes.

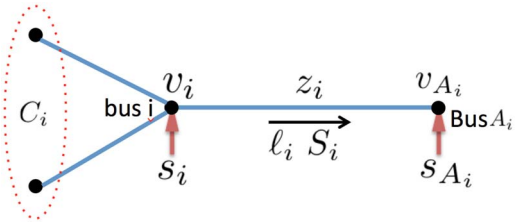


Fig. 1. Notations.

A. Branch Flow Model

We model a distribution network by a *directed* tree graph $\mathcal{T} := (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} := \{0, \ldots, n\}$ represents the set of buses and $\mathcal{E}$ represents the set of distribution lines connecting the buses in $\mathcal{N}$. Without loss of generality, index the root of the tree by 0. Each node $i \in \mathcal{N}\backslash\{0\}$ has a unique ancestor A_i and a set of children nodes, denoted by C_i. Denote the set of neighbors of node i by N_i, which includes its ancestor A_i, its set of children C_i and itself, i.e., $N_i := \{A_i\} \cup \{i\} \cup C_i$. We assume each directed line points towards the root, i.e., each line points from a node i to its unique ancestor A_i. We hence use a single index i to label the line from i to A_i and $\mathcal{E} := \{1, \ldots, n\}$.

For each bus $i \in \mathcal{N}$, let $V_i = |V_i|e^{\mathbf{i}\theta_i}$ be its complex voltage and $v_i := |V_i|^2$ be its magnitude squared. Let $s_i := p_i + \mathbf{i}q_i$ be its net complex power injection which is generation minus load. For each line $i \in \mathcal{E}$, let $z_i = r_i + \mathbf{i}x_i$ be its complex impedance. Let I_i be the complex branch current from bus i to A_i and $\ell_i := |I_i|^2$ be its magnitude squared. Let $S_i := P_i + \mathbf{i}Q_i$ be the branch power flow from bus i to A_i. The notations are illustrated in Fig. 1. A variable without a subscript denotes a column vector with appropriate components, as summarized below.

$v := (v_i, i \in \mathcal{N})$	$s := (s_i, i \in \mathcal{N})$
$\ell := (\ell_i, i \in \mathcal{E})$	$S := (S_i, i \in \mathcal{E})$

Branch flow model is first proposed in [23] and [24] for radial networks. It has better numerical stability than bus injection model and has been advocated for the design and operation for radial distribution network, [5], [14], [25], [26]. The branch flow model eliminates the phase angles of voltages and currents, and uses only the set of variables (v, s, ℓ, S). Given a radial network $\mathcal{T}$, the branch flow model is defined by:

$$v_{A_i} - v_i + \left(z_i S_i^* + S_i z_i^*\right) - \ell_i |z_i|^2 = 0 \quad i \in \mathcal{E} \tag{1a}$$

$$\sum_{j \in C_i} \left(S_j - \ell_j z_j\right) + s_i - S_i = 0 \quad i \in \mathcal{N} \tag{1b}$$

$$|S_i|^2 = v_i \ell_i \quad i \in \mathcal{N} \tag{1c}$$

where $S_0 = 0, \ell_0 = 0$ (the root of the tree does not have parent) for ease of presentation. Given a vector (v, s, ℓ, S) that satisfies (1), the phase angles of the voltages and currents can be uniquely determined if the network is a tree. Hence the branch flow model (1) is equivalent to a full AC power flow model. See [5, Sec. III-A] for details.

B. OPF and SOCP Relaxation

The OPF problem seeks to optimize certain objective, e.g., total line loss or total generation cost, subject to power flow equations (1) and various operational constraints. We consider an objective function of the following form:

$$F(s) = \sum_{i\in\mathcal{N}} f_i(s_i), \tag{2}$$

For instance,

- to minimize total line loss, we can set $f_i(s_i) = p_i$ for each bus $i \in \mathcal{N}$.
- to minimize generation cost, we can set $f_i(s_i) = \frac{\alpha_i}{2}p_i^2 + \beta_i p_i$, where $\alpha_i, \beta_i \geq 0$. The value of α_i, β_i are determined by the characteristic of the generators. (For load buses i that do not have generators, the associated $\alpha_i, \beta_i = 0$.)

We consider two operational constraints. First, the power injection s_i at each bus i is in a region $\mathcal{I}_i$, i.e.,

$$s_i \in \mathcal{I}_i \quad \text{for } i \in \mathcal{N} \tag{3}$$

The feasible power injection region $\mathcal{I}_i$ is determined by the controllable devices attached to bus i. Some common controllable loads are:

- For controllable load, whose real power can vary within $[\underline{p}_i, \overline{p}_i]$ and reactive power can vary within $[\underline{q}_i, \overline{q}_i]$, the injection region $\mathcal{I}_i$ is
$$\mathcal{I}_i = \left\{p + \mathbf{i}q \mid p \in \left[\underline{p}_i, \overline{p}_i\right], q \in \left[\underline{q}_i, \overline{q}_i\right]\right\} \subseteq \mathbb{C} \tag{4a}$$
- For solar panel connecting the grid through a inverter with nameplate $\overline{s}_i$, the injection region $\mathcal{I}_i$ is
$$\mathcal{I}_i = \left\{p + \mathbf{i}q \mid p \geq 0, p^2 + q^2 \leq \overline{s}_i^2\right\} \subseteq \mathbb{C} \tag{4b}$$

Second, the voltage magnitude at each bus $i \in \mathcal{N}$ needs to be within a prescribed region, i.e.,

$$\underline{v}_i \leq v_i \leq \overline{v}_i \quad \text{for } i \in \mathcal{N} \tag{5}$$

Typically the voltage magnitude at the substation bus 0 is assumed to be fixed at some prescribed value, i.e., $\underline{v}_0 = \overline{v}_0$. At other bus $i \in \mathcal{N}_+$, the voltage magnitude is typically allowed to deviate by 5% from its nominal value 1, i.e., $\underline{v}_i = 0.95^2$ and $\overline{v}_i = 1.05^2$.

To summarize, the OPF problem for radial network is

$$\begin{aligned} \textbf{OPF:}\ \min \quad & \sum_{i\in\mathcal{N}} f_i(s_i) \\ \text{over} \quad & v, s, S, \ell \\ \text{s.t.} \quad & (1),\ (3) \text{ and } (5) \end{aligned} \tag{6}$$

The OPF problem (6) is nonconvex due to the quadratic equality constraint (1c). In [4] and [5], (1c) is relaxed to a second order cone constraint:

$$|S_i|^2 \leq v_i \ell_i \quad \text{for} \quad i \in \mathcal{E}, \tag{7}$$

resulting in a second-order cone program (SOCP) relaxation of (6)

$$\begin{aligned} \textbf{ROPF:}\ \min \quad & \sum_{i\in\mathcal{N}} f_i(s_i) \\ \text{over} \quad & v, s, S, \ell \\ \text{s.t.} \quad & (1a),\ (1b),\ (7) \text{ and } (3),\ (5) \end{aligned} \tag{8}$$

Clearly the relaxation ROPF (8) provides a lower bound for the original OPF problem (6) since the original feasible set is enlarged. The relaxation is called *exact* if every optimal solution of ROPF attains equality in (1c) and hence is also optimal for the original OPF. For network with tree topology, SOCP relaxation is exact under some mild conditions [5], [25].

III. Distributed Algorithm for OPF

We assume SOCP relaxation is exact and develop in this section a distributed algorithm that solves the ROPF problem (8). We first develop an ADMM based distributed algorithm for a broad class of optimization problem. Then we show ROPF, which is a special case, can be solved in a distributed manner using the above method. In particular, a sufficient condition is derived for the existence of closed form solutions to the optimization subproblems of ROPF. The sufficient condition holds for most practical applications, and hence computation time can be greatly reduced by the proposed algorithm in practice.

A. ADMM Based Distributed Algorithm

For our application, consider the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{N}} f_i(x_i) && \text{(9a)} \\ \text{over} \quad & \{x_i \mid i \in \mathcal{N}\} \\ \text{s.t.} \quad & \sum_{j\in N_i} A_{ij}x_j = 0 \quad \text{for} \quad i \in \mathcal{N} && \text{(9b)} \\ & x_i \in \mathcal{K}_i \quad \text{for} \quad i \in \mathcal{N} && \text{(9c)} \end{aligned}$$

where for each $i \in \mathcal{N}$, x_i is a complex vector, $f_i(x_i)$ is a convex function, $\mathcal{K}_i$ is a convex set, and A_{ij} $(j \in N_i)$ are matrices with appropriate dimensions. A broad class of graph optimization problem (including ROPF) can be formulated as (9). Each node $i \in \mathcal{N}$ is associated with some local variables stacked as x_i, which belongs to a local feasible set $\mathcal{K}_i$ and has an objective function $f_i(x_i)$. Variables in node i is coupled with variables from its neighboring nodes in N_i through linear constraints (9b).

The goal is to develop a distributed algorithm that solves (9) such that each node i solves its own subproblems and only exchanges information with its neighboring nodes N_i. ADMM blends the decomposability of dual decomposition with the superior convergence properties of the method of multipliers [18]. To solve (9) using ADMM, we introduce a set of slack variables y_{ji}, which represents the variable from node j observed at node i, for $j \in N_i$. Then (9) can be reformulated as

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{N}} f_i(x_i) && \text{(10a)} \\ \text{over} \quad & x = \{x_i \mid i \in \mathcal{N}\} \\ & y = \left\{y_{ji} \mid j \in N_i, i \in \mathcal{N}\right\} \\ \text{s.t.} \quad & \sum_{j\in N_i} A_{ij}y_{ji} = 0 \quad i \in \mathcal{N} && \text{(10b)} \\ & x_i \in \mathcal{K}_i \quad i \in \mathcal{N} && \text{(10c)} \\ & x_i = y_{ij} \quad j \in N_i \quad i \in \mathcal{N} && \text{(10d)} \end{aligned}$$

where x and y represent the two groups of variables in standard ADMM defined in [18, Eq. (3.1)].[2] All the "observations" y_{ij} are forced to equal to the original variable x_i, thus the solution x^* to (10) is also optimal to the original problem (9).

Note that in (10), (10b) and (10c) involves either x or y and the consensus constraint (10d) involves both x and y. Hence, we will relax the constraint (10d) according to ADMM. Let μ_{ij} denote the Lagrangian multiplier for (10d) and the augmented Lagrangian is defined as

$$L_\rho(x, y, \mu) \coloneqq \sum_{i\in\mathcal{N}}\left(f_i(x_i) + \sum_{j\in N_i}\left(\langle \mu_{ij}, x_i - y_{ij}\rangle + \frac{\rho}{2}\|x_i - y_{ij}\|_2^2\right)\right) \tag{11}$$

where $\rho \geq 0$ is a constant. When $\rho = 0$, the augmented Lagrangian reduces to the standard Lagrangian. At each iteration k, ADMM consists of the iteration:

$$x^{k+1} \in \arg\min_{x\in\mathcal{K}_x} L_\rho\left(x, y^k, \mu^k\right) \tag{12a}$$

$$y^{k+1} \in \arg\min_{y\in\mathcal{K}_y} L_\rho\left(x^{k+1}, y, \mu^k\right) \tag{12b}$$

$$\mu^{k+1} = \mu^k + \rho\left(x^{k+1} - y^{k+1}\right) \tag{12c}$$

where $\mathcal{K}_x = \{x \mid x_i \in \mathcal{K}_i, i \in \mathcal{N}\}$ and $\mathcal{K}_y = \{y \mid \sum_{j\in N_i} A_{ij}y_{ji} = 0, i \in \mathcal{N}\}$. Specifically, at each iteration, ADMM first updates x based on (12a), then updates y based on (12b), and finally updates the multipliers (12c). Compared to dual decomposition, ADMM is guaranteed to converge to an optimal solution under less restrictive conditions. Interested readers may refer to [18, Sec. 3.2] for more details. Denote

$$r^k \coloneqq \left\|x^k - y^k\right\|_2 \tag{13a}$$

$$s^k \coloneqq \rho\left\|y^k - y^{k-1}\right\|_2 \tag{13b}$$

which can be viewed as the residuals for primal and dual feasibility and converge to 0 at optimality. They are used as convergence metrics in our experiment.

Next, we show that both the x-update (12a) and y-update (12b) can be decomposed into small subproblems that can be solved in parallel by each node i with only neighborhood communications, i.e., the problem (10) can be solved in a distributed manner using ADMM. For ease of presentation, we remove the iteration number k in (12) for all the variables, which will be updated accordingly after each subproblem is solved.

For each node $i \in \mathcal{N}$, it updates not only the original variable x_i, but also the "observation" of variables y_{ji} from its neighbor N_i and the associated multiplier μ_{ji}. Let $\mathcal{A}_i$ denote the set of local variables and

$$\mathcal{A}_i \coloneqq \{x_i\} \cup \left\{y_{ji} \mid j \in N_i\right\} \cup \left\{\mu_{ji} \mid j \in N_i\right\}$$

[2]In standard ADMM, the y variable is labeled as z. We use y in this paper since z represents impedance in power system.

In the x-update (12a), we solve the following problem to update $x_i \in \mathcal{A}_i$:

$$\min_{x\in\mathcal{K}_x} L_\rho(x, y, \mu)$$

For the objective, it can be written as a sum of local objective as shown below.

$$\begin{aligned} &L_\rho(x, y, \mu) \\ &\quad= \sum_{i\in\mathcal{N}}\left(f_i(x_i) + \sum_{j\in N_i}\left(\langle \mu_{ij}, x_i - y_{ij}\rangle + \frac{\rho}{2}\|x_i - y_{ij}\|_2^2\right)\right) \\ &\quad= \sum_{i\in\mathcal{N}} H_i(x_i) - \sum_{i\in\mathcal{N}}\sum_{j\in N_i}\langle \mu_{ij}, y_{ij}\rangle \end{aligned}$$

where the last term is independent of the decision variable x and

$$H_i(x_i) \coloneqq f_i(x_i) + \sum_{j\in N_i}\left(\langle \mu_{ij}, x_i\rangle + \frac{\rho}{2}\|x_i - y_{ij}\|_2^2\right) \tag{14}$$

For the constraint set $\mathcal{K}_x$, it can be represented as a Cartesian product of $|\mathcal{N}|$ disjoint set, i.e.,

$$\mathcal{K}_x = \otimes_{i\in\mathcal{N}}\mathcal{K}_i$$

Hence, the x-update (12a) can be decomposed into $|\mathcal{N}|$ subproblems and the subproblem for node i is:

$$\min_{x_i\in\mathcal{K}_i} H_i(x_i) \tag{15}$$

whose solution is the new update of $x_i \in \mathcal{A}_i$.

In (14), the variables $y_{ij}, \mu_{ij} \in \mathcal{A}_j$ are stored in i's neighbor N_i. Hence, each agent i needs to collect (y_{ij}, μ_{ij}) from all of its neighbor prior to solving (15). The message exchanges in the x-update is illustrated in Fig. 2a.

In the y-update (12b), we solve the following problem to update $\{y_{ji} \mid j \in N_i\} \in \mathcal{A}_i$:

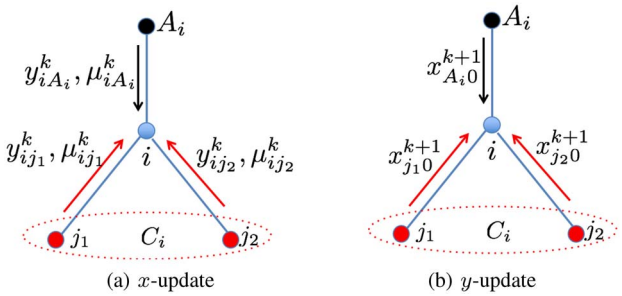
$$\min_{y\in\mathcal{K}_y} L_\rho(x, y, \mu)$$

The objective can also be written as a sum of local objective as shown below.

$$\begin{aligned} &L_\rho(x, y, \mu) \\ &\quad= \sum_{i\in\mathcal{N}}\left(f_i(x_i) + \sum_{j\in N_i}\left(\langle \mu_{ij}, x_i - y_{ij}\rangle + \frac{\rho}{2}\|x_i - y_{ij}\|_2^2\right)\right) \\ &\quad= \sum_{i\in\mathcal{N}}\left(f_i(x_i) + \sum_{j\in N_i}\left(\langle \mu_{ji}, x_j - y_{ji}\rangle + \frac{\rho}{2}\|x_j - y_{ji}\|_2^2\right)\right) \\ &\quad= \sum_{i\in\mathcal{N}} G_i\left(\left\{y_{ji} \mid j \in N_i\right\}\right) + \sum_{i\in\mathcal{N}}\left(f_i(x_i) + \sum_{j\in N_i}\langle \mu_{ji}, x_j\rangle\right) \end{aligned}$$

where

$$G_i\left(\left\{y_{ji} \mid j \in N_i\right\}\right) = \sum_{j\in N_i}\left(-\langle \mu_{ji}, y_{ji}\rangle + \frac{\rho}{2}\|x_j - y_{ji}\|_2^2\right) \tag{16}$$



(a) x-update (b) y-update

Fig. 2. Message exchanges in the x and y-update step for node i.

For the constraint set $\mathcal{K}_y$, it can also be represented as a Cartesian product of $|\mathcal{N}|$ disjoint set, i.e.,

$$\mathcal{K}_y = \otimes_{i\in\mathcal{N}_i}\left\{ y_{ji}, j\in N_i \middle| \sum_{j\in N_i} A_{ij}y_{ji} = 0 \right\}$$

Hence, the y-update (12b) can be decomposed into $|\mathcal{N}|$ subproblems and the subproblem for node i is

$$\begin{aligned} \min \quad & G_i\big(\{y_{ji} \mid j\in N_i\}\big) \\ \text{over} \quad & \{y_{ji} \mid j\in N_i\} \\ \text{s.t.} \quad & \sum_{j\in N_i} A_{ij}y_{ji} = 0 \end{aligned} \tag{17}$$

whose solution is the new update of $\{y_{ji} \mid j\in N_i\} \in \mathcal{A}_i$. In (16), the variables $x_j \in \mathcal{A}_j$ are stored in i's neighbor N_i. Hence, each agent i needs to collect x_j from all of its neighbor prior to solving (17). The message exchanges in the y-update is illustrated in Fig. 2b.

Next, we show how to solve (17) in closed form. For each i, we can stack the real and imaginary part of the variables $(y_{ji} \mid j\in N_i)$ in a vector with appropriate dimensions and denote it as $\tilde{y}$. Then the subproblem solved by agent i in the y-update (17) takes the following form:

$$\begin{aligned} \min_{\tilde{y}} \quad & \frac{1}{2}\tilde{y}^H M\tilde{y} + c^H\tilde{y} \\ \text{s.t.} \quad & B\tilde{y} = 0 \end{aligned} \tag{18}$$

where M is a positive diagonal matrix, B is a full row rank real matrix and c is a real vector. M, c, B are derived from (17). There exists a closed form expression for (18) given by

$$\tilde{y} = \left(M^{-1}B^H\left(BM^{-1}B^H\right)^{-1}BM^{-1} - M^{-1}\right)c \tag{19}$$

In summary, the original problem (9) is decomposed into local subproblems that can be solved in a distributed manner using ADMM. At each iteration, each node i solves (15) in the x-update and (17) in the y-update. There exists a closed form solution to the subproblem (17) in the y-update as shown in (19), hence whether (9) can be solved efficiently in a distributed manner depends on the existence of a closed form solution to (15), which depends on $f_i(x_i)$ and the constraint set $\mathcal{K}_i$.

Next, we show the ROPF problem (8) is a special case of (9), hence can be solved in a distributed manner using the above method.

B. Apply ADMM to OPF Problem

We assume the SOCP relaxation is exact and now derive a distributed algorithm for solving ROPF (8). Specifically, using the ADMM based algorithm developed in Section III-A, the global ROPF problem is decomposed into local subproblems that can be solved in a distributed manner with only neighborhood communication. In addition, we provide a sufficient condition, which holds in practice, for the existence of closed form solutions to all the optimization subproblems. Compared with existing methods, e.g., [8]–[15], that use generic iterative optimization solver to solve each subproblem, the computation time is improved by more than 1000 times.

We first describe the sufficient condition for the existence of a closed form solution below.

Theorem 1: Suppose there exists a closed form solution to the following optimization problem for all $i\in\mathcal{N}$

$$\begin{aligned} \min \quad & f_i(s) + \frac{\rho}{2}\|s-\hat{s}\|_2^2 \\ \text{over} \quad & s\in\mathcal{I}_i \end{aligned} \tag{20}$$

given any constant $\hat{s}$ and ρ, then there exists a closed form solution to the subproblem in the x-update (15) for ROPF.

Proof: We will prove Theorem 1 through elaborating the procedure to solve (15). ■

Recall that there is always a closed form solution to the optimization subproblem (17) in the y-update. If the objective function $f_i(s)$ and injection region $\mathcal{I}_i$ satisfy the sufficient condition in Theorem 1, all the subproblems can be solved with closed form solutions using the ADMM based algorithm proposed in Section III-A.

Remark 1: In practice, the objective function $f_i(s)$, usually takes the form of $f_i(s) := \frac{\alpha_i}{2}p^2 + \beta_i p$, which models both line loss and generation cost minimization as discussed in Section II-B. For the injection region $\mathcal{I}_i$, it usually takes either (4a) or (4b). It is shown in Appendix B that there exists a closed form solution for all of those cases. Thus, (15) can be solved with closed form for practical applications.

The ROPF problem defined in (8) can be written explicitly as:

$$\min \quad \sum_{i\in\mathcal{N}} f_i(s_i) \tag{21a}$$

$$\text{over} \quad v, s, S, \ell$$

$$\text{s.t.} \quad v_{A_i} - v_i + z_i S_i^* + S_i z_i^* - \ell_i |z_i|^2 = 0 \quad i\in\mathcal{E} \tag{21b}$$

$$\sum_{j\in C_i}(S_j - z_j\ell_j) - S_i + s_i = 0 \quad i\in\mathcal{N} \tag{21c}$$

$$|S_i|^2 \le v_i\ell_i \quad i\in\mathcal{N} \tag{21d}$$

$$s_i \in \mathcal{I}_i \quad i\in\mathcal{N} \tag{21e}$$

$$\underline{v}_i \le v_i \le \bar{v}_i \quad i\in\mathcal{N} \tag{21f}$$

Denote

$$x_i := \{v_i, s_i, S_i, \ell_i\} \tag{22}$$

$$\mathcal{K}_i := \left\{ x_i \mid |S_i|^2 \le v_i\ell_i,\ s_i\in\mathcal{I}_i,\ \underline{v}_i \le v_i \le \bar{v}_i \right\} \tag{23}$$

TABLE I
MULTIPLIERS ASSOCIATED WITH CONSTRAINTS (24b)

$\mu_{ii}^{(1)}$:	$S_i^{(x)} = S_{ii}^{(y)}$	$\mu_{ii}^{(2)}$:	$\ell_i^{(x)} = \ell_{ii}^{(y)}$
$\mu_{ii}^{(3)}$:	$v_i^{(x)} = v_{ii}^{(y)}$	$\mu_{ii}^{(4)}$:	$s_i^{(x)} = s_{ii}^{(y)}$
$\mu_{iA_i}^{(1)}$:	$S_i^{(x)} = S_{iA_i}^{(y)}$	$\mu_{iA_i}^{(2)}$:	$\ell_i^{(x)} = \ell_{iA_i}^{(y)}$
$\mu_{ij} (j \in C_i)$:	$v_i^{(x)} = v_{ij}^{(y)}$		

Then (1) can be written in the form of (9) where (21b)–(21c) correspond to (9b) and (21d)–(21f) correspond to (9c).

Following the procedure in Section III-A, we introduce an observation y_{ji}, which represents the variable of node j observed by node i. Then the counterpart of (10) is

$$\begin{aligned}
\min \quad & \sum_{i\in\mathcal{N}} f_i\left(s_i^{(x)}\right) && \text{(24a)}\\
\text{over} \quad & x \coloneqq \{x_i, i \in \mathcal{N}\} \\
& y \coloneqq \left\{y_{ji},\ j \in N_i, \quad i \in \mathcal{N}\right\} \\
\text{s.t.} \quad & v_{A_i i}^{(y)} - v_{ii}^{(y)} + z_i\left(S_{ii}^{(y)}\right)^* + S_{ii}^{(y)} z_i^* - \ell_{ii}^{(y)}|z_i|^2 = 0 \quad i \in \mathcal{E} \\
& \sum_{i\in C_i}\left(S_{ji}^{(y)} - z_j \ell_{ji}^{(y)}\right) - S_{ii}^{(y)} + s_{ii}^{(y)} = 0 \quad i \in \mathcal{N} \\
& \left|S_i^{(x)}\right|^2 \le v_i^{(x)} \ell_i^{(x)} \quad i \in \mathcal{E} \\
& s_i^{(x)} \in \mathcal{I}_i \quad i \in \mathcal{N} \\
& \underline{v}_i \le v_i^{(x)} \le \bar{v}_i \quad i \in \mathcal{N} \\
& x_i - y_{ij} = 0 \quad j \in N_i \quad i \in \mathcal{N} && \text{(24b)}
\end{aligned}$$

where we put superscripts $(\cdot)^{(x)}$ and $(\cdot)^{(y)}$ on each variable to denote whether the variable is updated in the x-update or y-update step.

For our application, each bus i does not need full information of its neighbor. Only voltage information $v_{A_i i}^{(y)}$ is needed from its parent A_i, and branch power $S_{ji}^{(y)}$ and current $\ell_{ji}^{(y)}$ information from its children $j \in C_i$ based on (24). Thus, y_{ij} contains only partial information about x_i, i.e.,

$$y_{ij} \coloneqq \begin{cases} \left(S_{ii}^{(y)}, \ell_{ii}^{(y)}, v_{ii}^{(y)}, s_{ii}^{(y)}\right) & j = i \\ \left(S_{iA_i}^{(y)}, \ell_{iA_i}^{(y)}\right) & j = A_i \\ \left(v_{ij}^{(y)}\right) & j \in C_i \end{cases}$$

As a result, x_i and y_{ij} do not consist of the same component for $j \neq i$ in the consensus constraint (24b). Here, we abuse notation and $x_i - y_{ij}$ is composed of the components that appear in both x_i and y_{ij}, i.e.,

$$x_i - y_{ij} \coloneqq \begin{cases} \left(S_i^{(x)} - S_{ii}^{(y)}, \ell_i^{(x)} - \ell_{ii}^{(y)}, v_i^{(x)} - v_{ii}^{(y)}, s_i^{(x)} - s_{ii}^{(y)}\right) & j = i \\ \left(S_i^{(x)} - S_{iA_i}^{(y)}, \ell_i^{(x)} - \ell_{iA_i}^{(y)}\right) & j = A_i \\ \left(v_i^{(x)} - v_{ij}^{(y)}\right) & j \in C_i \end{cases}$$

and the associated Lagrangian multiplier for each component is defined in Table I.

In the x-update, the subproblem (15) solved by each bus i can be written explicitly as

$$\begin{aligned}
\min \quad & H_i(x_i) && \text{(25a)}\\
\text{over} \quad & x_i \\
\text{s.t.} \quad & \left|S_i^{(x)}\right|^2 \le v_i^{(x)} \ell_i^{(x)} && \text{(25b)}\\
& s_i^{(x)} \in \mathcal{I}_i && \text{(25c)}\\
& \underline{v}_i \le v_i^{(x)} \le \bar{v}_i && \text{(25d)}
\end{aligned}$$

where $H_i(x_i)$ is defined in (14) and can be rewritten as

$$\begin{aligned}
H_i(x_i) = \ & f_i\left(s_i^{(x)}\right) - \sum_{j\in N_i}\langle \mu_{ij}, x_i\rangle + \frac{\rho}{2}\sum_{j\in N_i}\left\|x_i - y_{ij}\right\|_2^2 \\
= \ & \rho H_i^{(1)}\left(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)}\right) + H_i^{(2)}\left(s_i^{(x)}\right) + \text{constant}
\end{aligned} \tag{26}$$

where

$$\begin{aligned}
H_i^{(1)}\left(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)}\right) &= \left|S_i^{(x)} - \hat{S}_i\right|^2 + \left|\ell_i^{(x)} - \hat{\ell}_i\right|^2 \\
&\quad + \frac{|C_i|+1}{2}\left|v_i^{(x)} - \hat{v}_i\right|^2 \\
H_i^{(2)}\left(s_i^{(x)}\right) &= f_i\left(s_i^{(x)}\right) + \frac{\rho}{2}\left\|s_i^{(x)} - \hat{s}_i\right\|_2^2
\end{aligned}$$

The last step in (26) is obtained using completing the square and the variables labeled with hat are some constants, where

$$\begin{aligned}
\hat{S}_i &= \frac{S_{ii}^{(y)} + S_{iA_i}^{(y)}}{2} + \frac{\mu_{ii}^{(1)} + \mu_{iA_i}^{(1)}}{2\rho} \\
\hat{\ell}_i &= \frac{\ell_{ii}^{(y)} + \ell_{iA_i}^{(y)}}{2} + \frac{\mu_{ii}^{(2)} + \mu_{iA_i}^{(2)}}{2\rho} \\
\hat{v}_i &= \frac{v_{ii}^{(y)} + \sum_{j\in C_i} v_{ij}^{(y)}}{|C_i|+1} + \frac{\mu_{ii}^{(3)} + \sum_{j\in C_i}\mu_{ij}}{\rho(|C_i|+1)} \\
\hat{s}_i &= s_{ii}^{(y)} + \frac{\mu_{ii}^{(4)}}{\rho}
\end{aligned}$$

Thus, the objective (25a) in (25) can be decomposed into two parts, where the first part $H^{(1)}(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)})$ involves variables $(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)})$ and the second part $H^{(2)}(s_i^{(x)})$ involves $s_i^{(x)}$. Note that the constraint (25b)–(25d) can also be separated into two independent constraints. Variables $(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)})$ only depend on (25b) and (25d) and $s_i^{(x)}$ depends on (25c). Hence, (25) can be decomposed into two subproblems, where the first one solves the optimal $(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)})$ and the second one solves the optimal $s_i^{(x)}$. The first subproblem can be written explicitly as

$$\begin{aligned}
\min \quad & \left|S_i^{(x)} - \hat{S}_i\right|^2 + \left|\ell_i^{(x)} - \hat{\ell}_i\right|^2 + \frac{|C_i|+1}{2}\left|v_i^{(x)} - \hat{v}_i\right|^2 \\
\text{over} \quad & v_i^{(x)}, \ell_i^{(x)}, S_i^{(x)} \\
\text{s.t.} \quad & \left|S_i^{(x)}\right|^2 \le v_i^{(x)} \ell_i^{(x)} \\
& \underline{v}_i \le v_i^{(x)} \le \bar{v}_i
\end{aligned} \tag{27}$$

which has a quadratic objective, a second order cone constraint and a box constraint. Equation (27) has a closed form solution and the procedure is illustrated in Appendix A. Compared with

Algorithm 1 Initialization of the Algorithm

$v_i^{(x)} = 1$ for $i \in \mathcal{N}$
$s_i^{(x)}$ using any point in the injection region $\mathcal{I}_i$ for $i \in \mathcal{N}$
Initialize $S^{(x)}$ by calling DFS(0)
$\ell_i^{(x)} = \frac{|S_i^{(x)}|^2}{v_i^{(x)}}$ for $i \in \mathcal{N}$
$y_{ij} = x_i$ for $j \in N_i$ and $i \in \mathcal{N}$

function DFS(i)
  $S_i^{(x)} = s_i^{(x)}$
  for $j \in C_i$ **do**
    $S_i^{(x)} += \text{DFS}(j)$
  end for
  return $S_i^{(x)}$
end function

using generic iterative solver, the procedure is computationally efficient since it only requires solving for the roots of three polynomials with degrees less than or equal to 4, which have closed form expressions.

The second subproblem is

$$\begin{aligned} \min \quad & f_i\left(s_i^{(x)}\right) + \frac{\rho}{2}\left\|s_i^{(x)} - \hat{s}_i\right\|_2^2 \\ \text{over} \quad & s_i^{(x)} \in \mathcal{I}_i \end{aligned} \tag{28}$$

which takes the same form as of (20) in **Theorem 1** and thus can be solved with closed form. Recall that (27) can be solved with closed form, we have proved **Theorem 1**.

In the y-update, the subproblem solved by each agent i takes the form of (17) and has a closed form solution given in (19) and we do not reiterate here.

Finally, we specify the initialization and stopping criteria for the algorithm. A good initialization usually reduce the number of iterations for convergence. We use the following initialization suggested by our empirical results. We first initialize the x variable. The voltage magnitude square $v_i^{(x)} = 1$. The power injection $s_i^{(x)}$ is picked up from a feasible point in the feasible region $\mathcal{I}_i$. The branch power $S_i^{(x)}$ is the aggregate power injection $s_i^{(x)}$ from the nodes connected by line i (Note that the network has a tree topology.). The branch current $\ell_i^{(x)} = \frac{|S_i^{(x)}|^2}{v_i^{(x)}}$ according to (1c). The y variables are initialized using the corresponding x variable according to (24b). Intuitively, the above initialization procedure can be interpreted as a solution to the branch flow equation (1) assuming zero impedance on all the lines. The procedure is formally stated in Algorithm 1.

For the stopping criteria, it is suggested in [18] that a reasonable stopping criteria is that both the primal residual r^k defined in (13a) and the dual residual s^k defined in (13b) are within $10^{-3}\sqrt{|\mathcal{N}|}$ or $10^{-4}\sqrt{|\mathcal{N}|}$. The stopping criteria we adopt is that both r^k and s^k are below $10^{-4}\sqrt{|\mathcal{N}|}$ and empirical results show that the solution is accurate enough. The pseudo code for the algorithm is summarized in Algorithm 2.

Algorithm 2 Distributed OPF Algorithm for Balanced Radial Networks

Input:network $\mathcal{G}(\mathcal{N}, \mathcal{E})$, power injection region $\mathcal{I}_i$, voltage region $(\underline{v}_i, \bar{v}_i)$, line impedance z_i for $i \in \mathcal{N}$.
Output:voltage v, power injection s.

Initialize the x and y variables using Algorithm 1.
while $r^k > 10^{-4}\sqrt{|\mathcal{N}|}$ **and** $s^k > 10^{-4}\sqrt{|\mathcal{N}|}$ **do**
  In the x-update, each agent i solves (15) to update x.
  In the y-update, each agent i solves (17) to update y.
  In the multiplier update, update μ by (12c).
end while

IV. Case Study

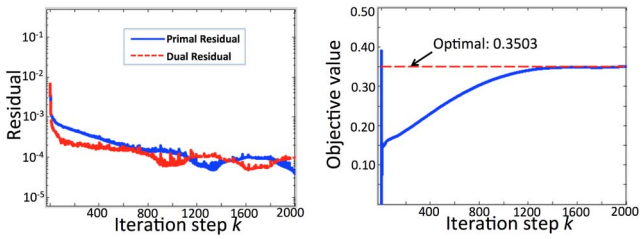
In this section, we first demonstrate the scalability of the distributed algorithm proposed in Section III by testing it on the model of a 2,065-bus distribution circuit in the service territory of the Southern California Edison (SCE). In particular, we also show the advantage of deriving closed form expression by comparing the computation time of solving the subproblems using off-the-shelf solver (SDPT3 [22] through CVX [27]) and our algorithm. Second, we simulate the proposed algorithm on networks of different sizes to understand the impact of network topology on convergence rate. The algorithm is implemented in Matlab 2014 and run on a Macbook pro 2014 with i5 dual core processor.

A. Simulation on a 2,065 Bus Circuit

In the 2,065 bus distribution network, there are 1,409 household loads whose power consumptions are within 0.07kw–7.6kw and 142 commercial loads, whose power consumptions are within 5kw–36.5kw. There are 135 rooftop PV panels, whose nameplates are within 0.7–4.5kw, distributed across the 1,409 houses.

The network is unbalanced three phase. We assume that the three phases are decoupled such that the network becomes identical single phase network. The voltage magnitude at each load bus is allowed to lie within [0.95, 1.05] per unit (pu), i.e., $\bar{v}_i = 1.05^2$ and $\underline{v}_i = 0.95^2$ for $i \in \mathcal{N}_+$. The control devices are the rooftop PV panels whose real and reactive power injections are controlled. The objective is to minimize power loss across the network, namely $f_i(s_i) = p_i$ for $i \in \mathcal{N}$ in (2). Each bus is an agent and there are 2,065 nodes in the network that solve the OPF problem in a distributed manner.

We mainly focus on the time of convergence (ToC) for the proposed distributed algorithm. The algorithm is run on a single machine. To roughly estimate the ToC (excluding communication overhead) if the algorithm is run on distributed machines, we divide the total time by the number of agents. Recall that the stopping criteria is that both the primal and dual residual are below $10^{-4}\sqrt{|\mathcal{N}|}$ and Figure 3a illustrates the evolution of $r^k/\sqrt{|\mathcal{N}|}$ and $s^k/\sqrt{|\mathcal{N}|}$ versus iterations k. The stopping criteria are satisfied after $1,114$ iterations. The evolution of the objective value is illustrated in Figure 3b. It takes 1,153s to run 1,114 iterations on a single computer. Then the ToC is roughly 0.56s if we implement the algorithm in a distributed manner not counting communication overhead.



(a) Primal and dual residual (b) Objective value

Fig. 3. Simulation results for 2065 bus distribution network.

TABLE II
STATISTICS OF DIFFERENT NETWORKS

Network	Diameter	Iteration	Total Time(s)	Avg time(s)
2065Bus	64	1114	1153	0.56
1313Bus	53	671	471	0.36
792Bus	45	524	226	0.29
363Bus	36	289	112	0.24
108Bus	16	267	16	0.14

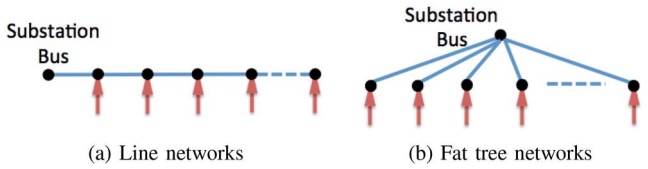
To further show the advantage of closed form solution, we also compare the computation time of solving the subproblems by an off-the-shelf solver (SDPT3) and by our algorithm. Specifically, we compare the average computation time of solving the subproblem in both the x-update and the y-update step. In the x-update step, the average time required to solve the subproblem is 5.1×10^{-4}s for the proposed algorithm but 0.3s for SDPT3. In the y-update step, the average time required to solve the subproblem is 1.7×10^{-4}s for the proposed algorithm but 0.2s for SDPT3. Thus, each ADMM iteration only takes about 6.8×10^{-4}s for the proposed algorithm but 0.5s for using iterative algorithm, which is a 1,000x speedup.

B. Rate of Convergence

In Section IV-A, we demonstrate that the proposed distributed algorithm can dramatically reduce the computation time within each iteration and therefore is scalable to a large practical 2,065 bus distribution network. Besides the computation time within each iteration, the time of convergence ToC also depends on the number of iterations, namely rate of convergence. In this subsection, we investigate the impact of the network topology on the number of iterations.

To the best of our knowledge, most of the works on convergence rate for ADMM based algorithms study how the primal/dual residual changes as the number of iterations increases. Specifically, it is proved in [28] and [29] that the general ADMM based algorithms converge linearly under certain assumptions. Here, we consider the rate of convergence from another two factors, network size N and diameter D, i.e., given the termination criteria in Algorithm 2, the impact from network size and diameter on the number of iterations. The impacts from other factors, e.g., form of objective function and constraints, etc. are beyond the scope of this paper.

First, we simulate the algorithm on different networks (that are subnetworks of the 2,065-bus system) and some statistics are given in Table II. For simplicity, we assume the number of iterations T to converge takes the linear form $T = aN + bD$. Using the data in Table II, the parameters $a = 0.34$, $b = 5.53$ give the least square error. It means that the network diameter has a stronger impact than the network size on the rate of convergence.



(a) Line networks (b) Fat tree networks

Fig. 4. Topology for tree and fat tree networks.

TABLE III
STATISTICS OF LINE AND FAT TREE NETWORKS

Size	# of iterations (Line)	# of iterations (Fat tree)
5	43	31
10	123	51
15	198	148
20	286	87
25	408	173
30	838	119
35	1471	187
40	2201	109
45	2586	182
50	3070	234

To further illustrate the phenomenon, we simulate the algorithm on two extreme cases: 1) Line network in Fig. 4a, whose diameter is the largest given the network size, 2) Fat tree network in Fig. 4b, whose diameter is the smallest (2) given the network size. In Table III, we record the number of iterations for both line and fat tree network of different sizes. For line network, the number of iterations increases notably as the size increases. For fat tree network, the trend is less obvious compared to line network.

V. CONCLUSION

In this paper, we developed an ADMM based distributed algorithm that solves the optimal power flow problem for balanced radial networks. In contrast to standard ADMM based distributed OPF algorithms that require solving optimization subproblems using iterative methods, our decomposition allows us to derive closed form solutions for these subproblems, greatly speeding up each ADMM iteration. Our simulation shows that the proposed algorithm is scalable to a 2,065-bus distribution system and the optimization subproblems in each ADMM iteration are solved 1,000x faster than using iterative methods.

APPENDIX A

Denote $x_1 := \mathbf{Re}(S_i^{(x)})$, $x_2 := \mathbf{Im}(S_i^{(x)})$, $x_3 := \sqrt{\frac{|C_i|+1}{2}} v_i^{(x)}$ and $x_4 := \ell_i^{(x)}$. Then the optimization problem (27) can be written equivalently as

$$\begin{aligned} \min \quad & \sum_{i=1}^{4} \left(x_i^2 + c_i x_i \right) && \text{(29a)} \\ \text{over} \quad & x_1,\ x_2,\ x_3,\ x_4 \\ \text{s.t.} \quad & \frac{x_1^2 + x_2^2}{x_3} \le k^2 x_4 && \text{(29b)} \\ & \underline{x}_3 \le x_3 \le \bar{x}_3 && \text{(29c)} \end{aligned}$$

where $\bar{x}_3 > \underline{x}_3 > 0$ and c_i, k are constants that hinges on the constants in (27).

Below we will derive a procedure that solves (29). Let $\gamma \geq 0$ denote the Lagrangian multiplier for constraint (29b) and $\underline{\lambda}, \overline{\lambda} \geq 0$ denote the Lagrangian multipliers for constraint (29c), then the Lagrangian of P1 is

$$L(x,\gamma,\lambda) = \sum_{i=1}^{4}\left(x_i^2 + c_i x_i\right) + \gamma\left(\frac{x_1^2+x_2^2}{x_3} - k^2 x_4\right) + \overline{\lambda}(x_3 - \bar{x}_3) - \underline{\lambda}\left(x_3 - \underline{x}_3\right)$$

The KKT optimality conditions imply that the optimal solution x^* together with the multipliers $\gamma^*, \underline{\lambda}^*, \overline{\lambda}^*$ satisfy the following equations. For ease of notations, we sometimes skip the superscript $\star$ of the variables in the following analysis.

$$2x_1 + c_1 + 2\gamma\frac{x_1}{x_3} = 0 \tag{30a}$$

$$2x_2 + c_2 + 2\gamma\frac{x_2}{x_3} = 0 \tag{30b}$$

$$2x_3 + c_3 - \gamma\frac{x_1^2+x_2^2}{x_3^2} + \overline{\lambda} - \underline{\lambda} = 0 \tag{30c}$$

$$2x_4 + c_4 - k^2\gamma = 0 \tag{30d}$$

$$\overline{\lambda}(x_3 - \bar{x}_3) = 0 \quad \overline{\lambda} \geq 0 \quad x_3 \leq \bar{x}_3 \tag{30e}$$

$$\underline{\lambda}\left(\underline{x}_3 - x_3\right) = 0 \quad \underline{\lambda} \geq 0 \quad x_3 \geq \underline{x}_3 \tag{30f}$$

$$\gamma\left(\frac{x_1^2+x_2^2}{x_3} - k^2 x_4\right) = 0 \quad \gamma \geq 0 \quad \frac{x_1^2+x_2^2}{x_3} \leq k^2 x_4 \tag{30g}$$

Lemma 1: There exists a unique solution $(x^*, \gamma^*, \underline{\lambda}^*, \overline{\lambda}^*)$ to (30) if $\bar{x}_3 > \underline{x}_3 \geq 0$.

Proof: P1 is feasible since $z = (0, 0, \underline{x}_3, 1)$ satisfies (29b)-(29c). In addition, P1 is a strictly convex optimization problem since the objective (29a) is a strictly convex function of z and the constraints (29b) and (29c) are also convex. Hence, there exists a unique solution z^* to P1, which indicates there exists a unique solution $(x^*, \gamma^*, \underline{\lambda}^*, \overline{\lambda}^*)$ to the KKT optimality conditions (30). ■

Lemma 1 says that there exists a unique solution to (30), which is also the optima to P1. In the following, we will solve (30) through enumerating value of the multipliers $\gamma, \overline{\lambda}, \underline{\lambda}$. Specifically, we first assume $\gamma^* = 0$ (Case 1 below), which is equivalent to assume constraint (29b) is inactive. If there is a feasible solution to (30), it is the unique solution to (30). Otherwise, we assume $\gamma^* = 0$ (Case 2 below), which is equivalent to assume that the equality is obtained at optimality in (29b).

Case 1: If $\gamma = 0$, (30) becomes

$$2x_1 + c_1 = 0 \tag{31a}$$

$$2x_2 + c_2 = 0 \tag{31b}$$

$$2x_3 + c_3 + \overline{\lambda} - \underline{\lambda} = 0 \tag{31c}$$

$$2x_4 + c_4 = 0 \tag{31d}$$

$$\overline{\lambda}(x_3 - \bar{x}_3) = 0 \quad \overline{\lambda} \geq 0 \quad x_3 \leq \bar{x}_3 \tag{31e}$$

$$\underline{\lambda}\left(x_3 - \underline{x}_3\right) = 0 \quad \underline{\lambda} \geq 0 \quad x_3 \geq \underline{x}_3 \tag{31f}$$

$$\frac{x_1^2+x_2^2}{x_3} \leq k^2 x_4 \tag{31g}$$

The solution to (31a)–(31f) ignoring (31g) is

$$x_1 = -\frac{c_1}{2},\ x_2 = -\frac{c_2}{2},\ x_3 = \left[-\frac{c_3}{2}\right]_{\underline{x}_3}^{\bar{x}_3},\ x_4 = -\frac{c_4}{2}$$
$$\overline{\lambda} = -(2x_3 + c_3)1_{\{x_3 = \bar{x}_3\}},\ \underline{\lambda} = -(2x_3 + c_3)1_{\{x_3 = \underline{x}_3\}}$$

and if the solution satisfies (31g), it is the solution to (30). Otherwise, we go to Case 2.

Case 2: If $\gamma > 0$, (30) becomes

$$2x_1 + c_1 + 2\gamma\frac{x_1}{x_3} = 0 \tag{32a}$$

$$2x_2 + c_2 + 2\gamma\frac{x_2}{x_3} = 0 \tag{32b}$$

$$2x_3 + c_3 - \gamma\frac{x_1^2+x_2^2}{x_3^2} + \overline{\lambda} - \underline{\lambda} = 0 \tag{32c}$$

$$\overline{\lambda}(x_3 - \bar{x}_3) = 0 \quad \overline{\lambda} \geq 0 \quad x_3 \leq \bar{x}_3 \tag{32d}$$

$$\underline{\lambda}\left(x_3 - \underline{x}_3\right) = 0 \quad \underline{\lambda} \geq 0 \quad x_3 \geq \underline{x}_3 \tag{32e}$$

$$\gamma = \frac{1}{k^2}(2x_4 + c_4) \tag{32f}$$

$$x_4 = \frac{x_1^2+x_2^2}{k^2 x_3} \tag{32g}$$

Substitute (32g) into (32f), we obtain

$$\gamma = \frac{1}{k^2}(2x_4 + c_4) = \frac{2\left(x_1^2+x_2^2\right)}{k^4 x_3} + \frac{c_4}{k^2} \tag{33}$$

Then substituting (32f) into (32a)-(32e), we can write (32) equivalently as

$$2 + \frac{c_1}{x_1} + 4\frac{\left(x_1^2+x_2^2\right)}{k^4 x_3^2} + \frac{2c_4}{k^2 x_3} = 0 \tag{34a}$$

$$2 + \frac{c_2}{x_2} + 4\frac{\left(x_1^2+x_2^2\right)}{k^4 x_3^2} + \frac{2c_4}{k^2 x_3} = 0 \tag{34b}$$

$$2 + \frac{c_3}{x_3} - 2\frac{\left(x_1^2+x_2^2\right)^2}{k^4 x_3^4} - c_4\frac{x_1^2+x_2^2}{x_3^3} + \frac{\overline{\lambda} - \underline{\lambda}}{x_3} = 0 \tag{34c}$$

$$\overline{\lambda}(x_3 - \bar{x}_3) = 0 \quad \overline{\lambda} \geq 0 \quad x_3 \leq \bar{x}_3 \tag{34d}$$

$$\underline{\lambda}\left(x_3 - \underline{x}_3\right) = 0 \quad \underline{\lambda} \geq 0 \quad x_3 \geq \underline{x}_3 \tag{34e}$$

where (34a)–(34c) are obtained through dividing both sides of (32a)–(32c) by x_1, x_2 and x_3, respectively. The variables γ, x_4 can be recovered via (32f) and (32g) after we solve (34).

By (34a) and (34b),

$$\frac{c_1}{x_1} = \frac{c_2}{x_2}$$

Denote $p := \frac{x_1}{c_1 x_3} = \frac{x_2}{c_2 x_3}$. Then (34) is equivalent to the following equations.

$$p = \frac{x_1}{c_1 x_3} = \frac{x_2}{c_2 x_3} \tag{35a}$$

$$2 + \frac{1}{p x_3} = -\left(\frac{4\left(c_1^2+c_2^2\right)}{k^4}p^2 + 2\frac{c_4}{k^2 x_3}\right) \tag{35b}$$

$$2+\frac{c_3}{x_3}=\frac{2\left(c_1^2+c_2^2\right)^2}{k^4}p^4+\frac{c_4\left(c_1^2+c_2^2\right)}{k^2}\frac{p^2}{x_3}+\frac{\underline{\lambda}-\overline{\lambda}}{x_3} \tag{35c}$$

$$\overline{\lambda}(x_3-\bar{x}_3)=0 \quad \overline{\lambda}\geq 0 \quad x_3\leq \bar{x}_3 \tag{35d}$$

$$\underline{\lambda}\left(x_3-\underline{x}_3\right)=0 \quad \underline{\lambda}\geq 0 \quad x_3\geq \underline{x}_3 \tag{35e}$$

where (35b) is obtained by substitute $x_2 = c_2 p x_3$ into (34b), (35c) is obtained by substitute $x_1 = c_1 p x_3$ and $x_2 = c_2 p x_3$ into (34c). To solve (35), we further divide our analysis into two sub-cases depending on whether x_3^* hits the lower or upper bound.

- **Case 2.1**: $x_3^* = \bar{x}_3$. ($\underline{\lambda} = 0, \overline{\lambda} > 0$) (The case of $x_3^* = \underline{x}_3$ can be solved using similar procedure.) We first substitute $x_3 = \bar{x}_3$ into (35b) and have

$$\frac{4\left(c_1^2+c_2^2\right)}{k^4}p^3+\left(2\frac{c_4}{k^2\bar{x}_3}+2\right)p+\frac{1}{\bar{x}_3}=0, \tag{36}$$

  whose solution[3] is denoted by p^*. Then substitute p^* and $\bar{x}_3$ into (35a), we can recover x_1^* and x_2^*. Then we can obtain $\gamma^*, \overline{\lambda}^*$ using (33) and (34c) by substituting $x_1^*, \ldots, x_4^*$. If $\gamma^*, \overline{\lambda}^* \geq 0$, they collectively solve (30). Otherwise, we go to Case 2.2.
- **Case 2.2**: $\underline{x}_3 < x_3^* < \bar{x}_3$ ($\underline{\lambda}, \overline{\lambda} = 0$). Since $\overline{\lambda}$ and $\underline{\lambda} = 0$, (35) reduces to

$$p=\frac{x_1}{c_1x_3}=\frac{x_2}{c_2x_3} \tag{37a}$$

$$2+\frac{1}{px_3}=-\left(\frac{4\left(c_1^2+c_2^2\right)}{k^4}p^2+2\frac{c_4}{k^2x_3}\right) \tag{37b}$$

$$2+\frac{c_3}{x_3}=\frac{2\left(c_1^2+c_2^2\right)^2}{k^4}p^4+\frac{c_4\left(c_1^2+c_2^2\right)}{k^2}\frac{p^2}{x_3} \tag{37c}$$

Dividing each side of (37b) by (37c) gives

$$\frac{2x_3+\frac{1}{p}}{2x_3+c_3}=-\frac{2}{\left(c_1^2+c_2^2\right)p^2},$$

which implies

$$x_3=-\frac{\left(c_1^2+c_2^2\right)p+2c_3}{2\left(\left(c_1^2+c_2^2\right)p^2+2\right)} \tag{38}$$

Then substitute (38) into (37b), we have

$$\frac{\left(c_1^2+c_2^2\right)p^2+2}{\left(c_1^2+c_2^2\right)p^2+2c_3p}-\frac{2\left(c_1^2+c_2^2\right)}{k^4}p^2+\frac{2c_4\left(\left(c_1^2+c_2^2\right)p^2+2\right)}{k^2\left(\left(c_1^2+c_2^2\right)p+2c_3\right)}=1$$

which is equivalent to

$$\frac{\left(c_1^2+c_2^2\right)^2}{k^4}p^4+\frac{c_1^2+c_2^2}{k^2}\left(\frac{2c_3}{k^2}-c_4\right)p^3+\left(c_3-\frac{2c_4}{k^2}\right)p-1=0$$

whose solution[3] is denoted by p^*. Substitute p^* into (38), we can recover x_3^*, then x_1^*, x_2^* can be recovered via (37a). γ^* is recovered using (33). If $\gamma^* \geq 0$, the corresponding solution solves (30).

[3]There are potentially multiple solutions and we need to check all the real solution p^* using the following procedure.

APPENDIX B

We assume $f_i(s) := \frac{\alpha_i}{2}p^2 + \beta_i p$ and derive a closed form solution to (20).

A. $\mathcal{I}_i$ Takes the Form of (4a)

In this case, (20) takes the following form:

$$\begin{aligned}\min_{p,q}\quad & \frac{a_1}{2}p^2+b_1p+\frac{a_2}{2}q^2+b_2q\\ \text{s.t.}\quad & \underline{p}_i\leq p\leq \overline{p}_i\\ & \underline{q}_i\leq q\leq \overline{q}_i\end{aligned}$$

where $a_1, a_2 > 0$ and b_1, b_2 are constants. Then the closed form solution is

$$p=\left[-\frac{b_1}{a_1}\right]_{\underline{p}_i}^{\overline{p}_i} \quad q=\left[\hat{-}\frac{2_1}{a_2}\right]_{\underline{q}_i}^{\overline{q}_i}$$

where $[x]_a^b := \min\{a, \max\{x, b\}\}$.

B. $\mathcal{I}_i$ Takes the Form of (4b)

The optimization problem (20) takes the following form

$$\min_{p,q}\quad \frac{a_1}{2}p^2+b_1p+\frac{a_2}{2}q^2+b_2q \tag{39a}$$

$$\text{s.t.}\quad p^2+q^2\leq c^2 \tag{39b}$$

$$p\geq 0 \tag{39c}$$

where $a_1, a_2, c > 0$, b_1, b_2 are constants. The solutions to (39) are given as below.

Case 1: $b_1 \geq 0$.

$$p^*=0 \qquad q^*=\left[-\frac{b_2}{a_2}\right]_{-c}^{c}$$

Case 2: $b_1 < 0$ and $\frac{b_1^2}{a_1^2}+\frac{b_2^2}{a_2^2}\leq c^2$.

$$p^*=-\frac{b_1}{a_1} \qquad q^*=-\frac{b_2}{a_2}$$

Case 3: $b_1 < 0$ and $\frac{b_1^2}{a_1^2}+\frac{b_2^2}{a_2^2} > c^2$.

First solve the following equation in terms of variable λ:

$$b_1^2(a_2+2\lambda)^2+b_2^2(a_1+2\lambda)^2=(a_1+2\lambda)^2(a_2+2\lambda)^2 \tag{40}$$

which is a polynomial with degree of 4 and has closed form expression. There are four solutions to (40), but there is only one strictly positive λ^*, which can be proved via the KKT conditions of (39). Then we can recover p^*, q^* from λ^* using the following equations.

$$p^*=-\frac{b_1}{a_1+2\lambda^*} \quad \text{and} \quad q^*=-\frac{b_2}{a_2+2\lambda^*}$$

The above procedure to solve (39) is derived from standard applications of the KKT conditions of (39). For brevity, we skip the proof here.

REFERENCES

[1] Q. Peng and S. H. Low, "Distributed algorithm for optimal power flow on a radial network," in *Proc. IEEE 53rd Annu. Conf. Decis. Control (CDC)*, Los Angeles, CA, USA, 2014, pp. 167–172.

[2] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, "Semidefinite programming for optimal power flow problems," *Int. J. Elect. Power Energy Syst.*, vol. 30, nos. 6–7, pp. 383–392, 2008.

[3] R. Jabr, "Radial distribution load flow using conic programming," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1458–1459, Aug. 2006.

[4] M. Farivar, C. R. Clarke, S. H. Low, and K. M. Chandy, "Inverter VAR control for distribution systems with renewables," in *Proc. IEEE SmartGridComm Conf.*, Brussels, Belgium, Oct. 2011, pp. 457–462.

[5] M. Farivar and S. H. Low, "Branch flow model: Relaxations and convexification (parts I, II)," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2554–2572, Aug. 2013.

[6] S. H. Low, "Convex relaxation of optimal power flow—Part I: Formulations and relaxations," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 15–27, Mar. 2014.

[7] S. H. Low, "Convex relaxation of optimal power flow—Part II: Exactness," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 2, pp. 177–189, Jun. 2014.

[8] B. H. Kim and R. Baldick, "Coarse-grained distributed optimal power flow," *IEEE Trans. Power Syst.*, vol. 12, no. 2, pp. 932–939, May 1997.

[9] R. Baldick, B. H. Kim, C. Chase, and Y. Luo, "A fast distributed implementation of optimal power flow," *IEEE Trans. Power Syst.*, vol. 14, no. 3, pp. 858–864, Aug. 1999.

[10] A. X. Sun, D. T. Phan, and S. Ghosh, "Fully decentralized AC optimal power flow algorithms," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PES)*, Vancouver, BC, Canada, 2013, pp. 1–5.

[11] A. Y. S. Lam, B. Zhang, and D. N. Tse, "Distributed algorithms for optimal power flow problem," in *Proc. IEEE 51st Annu. Conf. Decis. Control (CDC)*, 2012, pp. 430–437.

[12] A. Lam, B. Zhang, A. Dominguez-Garcia, and D. Tse, "Optimal distributed voltage regulation in power distribution networks," *arXiv preprint arXiv:1204.5226*, 2012.

[13] E. Devane and I. Lestas, "Stability and convergence of distributed algorithms for the OPF problem," in *Proc. IEEE 52nd Annu. Conf. Decis. Control (CDC)*, Florence, Italy, 2013, pp. 2933–2938.

[14] N. Li, L. Chen, and S. H. Low, "Demand response in radial distribution networks: Distributed algorithm," in *Proc. 46th Asilomar Conf. Rec. Signals Syst. Comput. (ASILOMAR)*, Pacific Grove, CA, USA, 2012, pp. 1549–1553.

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

[16] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, "Dynamic network energy management via proximal message passing," *Optimization*, vol. 1, no. 2, pp. 70–122, 2013.

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

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[19] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 644–658, Mar. 2013.

[20] D. L. Sun and C. Fevotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 6201–6205.

[21] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

[22] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3—A MATLAB software package for semidefinite programming, version 1.3," *Optim. Methods Softw.*, vol. 11, nos. 1–4, pp. 545–581, 1999.

[23] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Trans. Power Del.*, vol. 4, no. 1, pp. 725–734, Jan. 1989.

[24] M. E. Baran and F. F. Wu, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Trans. Power Del.*, vol. 4, no. 1, pp. 735–743, Jan. 1989.

[25] L. Gan, N. Li, U. Topcu, and S. H. Low, "Exact convex relaxation of optimal power flow in radial networks," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 72–87, Jan. 2015.

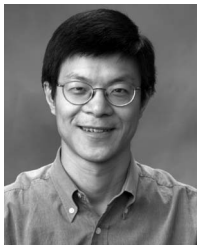
[26] Q. Peng, Y. Tang, and S. H. Low, "Feeder reconfiguration in distribution networks based on convex relaxation of OPF," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 1793–1804, Jul. 2015.

[27] M. Grant, S. Boyd, and Y. Ye. (2015). *CVX: MATLAB Software for Disciplined Convex Programming. 2009*. [Online]. Available: http://stanford.edu/∼boyd/cvx

[28] E. Wei and A. Ozdaglar, "On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers," *arXiv preprint arXiv:1307.8254*, 2013.

[29] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *arXiv preprint arXiv:1208.3922*, 2012.

Qiuyu Peng received the B.S. degree in electrical engineering from Shanghai Jiaotong University, China, in 2011. He is currently pursuing the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena, USA. His research interests are in control and optimization of communication and power systems.



Steven H. Low (F'08) received the B.S. degree from Cornell University, and the Ph.D. degree from the University of California, Berkeley, both in electrical engineering. He was at AT&T Bell Laboratories, Murray Hill, NJ, and the University of Melbourne, Australia. He is a Professor with the Department of Computing and Mathematical Sciences, and the Department of Electrical Engineering, California Institute of Technology. His research interests are in cyber and cyber-physical networks such as power systems. He is a Senior Editor of the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS, and the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. He is on the Editorial Board of NOW Foundations and Trends in Electric Energy Systems, and Foundations and Trends in Networking, as well as the *Journal on Sustainable Energy, Grids, and Networks*.