Distributed Algorithm for Optimal Power Flow on a Radial Network

Qiuyu Peng and Steven H. Low

Abstract— **The optimal power flow (OPF) problem is fundamental in power system operations and planning. Large-scale renewable penetration calls for real-time feedback control, and hence the need for fast and distributed solutions for OPF. This is difficult because OPF is nonconvex and Kirchhoff's laws are global. In this paper we propose a solution for 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 (SOCP) relaxation. Our distributed algorithm is based on alternating direction method of multiplier (ADMM), but unlike standard ADMM algorithms that often require iteratively solving optimization subproblems in each ADMM iteration, our decomposition allows us to derive closed form solutions for these subproblems, greatly speeding up each ADMM iteration. We present simulations on a real-world 2,065-bus distribution network to illustrate the scalability and optimality of the proposed algorithm.**

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 problems 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 OPF in such an environment has at least two challenges.

First the OPF problem is hard to solve because of its nonconvex feasible set. Recently a new approach through convex relaxation has been developed to address nonconvexity due to the Kirchhoff's laws. Specifically semidefinite program (SDP) relaxation [1] and second order cone program (SOCP) relaxation [2] have bee proposed in the bus injection model, and SOCP relaxation has been proposed in the branch flow model [3], [4]. See the tutorial [5], [6] 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 those 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, e.g., centrally schedule a relatively small number of generators. In future networks that optimize simultaneously the (possibly real-time) 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 by proposing a distributed algorithm for solving the SOCP relaxation of OPF for radial networks.

Various distributed algorithms have been developed to solve the OPF problem. In [7], augmented Lagrangian decomposition method is used to solve the multi-area OPF problem where, in each iteration, each agent solves its own subproblem and communicates its result with neighbors. These early distributed algorithms do not deal with the nonconvexity issue of OPF. In contrast, dual decompositions are applied to solve the SDP relaxation of OPF in [8]. To improve the convergence rate, alternating direction method of multiplier (ADMM) [9]–[11] have been applied to develop distributed algorithms for (possibly the convexified) OPF problems.

Each iteration of an ADMM algorithm needs to solve multiple subproblems to update the primal variables [12]. The total computation time is determined by the number of iterations and the computation time to solve the subproblems in each iteration. To improve the total computation time, we can reduce the computation time for each subproblem. To the best of our knowledge, all ADMM algorithms for OPF in the literature solve these subproblems iteratively. In this paper we minimize the computation time of each iteration by deriving a closed form solution for the subproblems, significantly reducing the overall computation time.

Specifically we decompose the OPF problem into smaller subproblems based on ADMM. The proposed algorithm has two advantages: 1) There is a closed form solution for each subproblem, thus eliminating the need for an iterative procedure for each ADMM iteration. 2) Communication is only required between adjacent buses.

We demonstrate the scalability of the proposed algorithms using a real-life network. In particular, we show that the algorithm converges within 0.6s for a 2,065-bus system. As expected, solving each subproblem in a closed form can be orders of magnitude faster than solving it iteratively using an off-the-shelf optimization solver CVX [13]: our solver

This work was supported by ARPA-E grant DE-AR0000226, Southern California Edison, National Science Council of Taiwan, R.O.C. grant NSC 103-3113-P-008-001, Los Alamos National Lab through an DoE grant and Resnick Institute.

Qiuyu Peng is with Department of EE and Steven H. Low is with Department of CMS and EE, California Institute of Technology, Pasadena, CA 91125, USA. {qpeng, slow}@caltech.edu

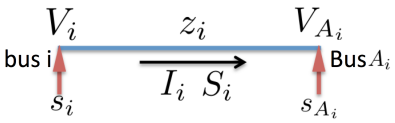


Fig. 1: Notations.

requires on average 6.8×10^{-4}s while CVX requires on average 0.5s.

The rest of the paper is structured as follows. The OPF problem is defined in section II. In section III, we develop our distributed algorithm. In section IV, we test its scalability using data from a real-world distribution network. We conclude in section V.

II. Problem formulation

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

A. Branch flow model

We model a distribution network by a *directed* tree graph $\mathcal{T} := (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} := \{0, \ldots, n\}$ represent the set of buses and $\mathcal{E}$ represent the set of distribution lines connecting the buses in $\mathcal{N}$. Index the root of the tree by 0 and let $\mathcal{N}_+ := \mathcal{N} \setminus \{0\}$ denote the other buses. For each node i, it has a unique ancestor A_i and a set of children nodes, denoted by C_i. We adopt the graph orientation where every line points towards the root. Each directed line connects a node i and its unique ancestor A_i. We hence label the lines by $\mathcal{E} := \{1, \ldots, n\}$ where each $i \in \mathcal{E}$ denotes a line from i to A_i.

The root of the tree $\mathcal{T}$ is a substation bus that is connected to the transmission network. It has a fixed voltage and redistributes the bulk power it receives from the transmission network to other buses. 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 defined as 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.

$v := (v_i, i \in \mathcal{N})$	$p := (p_i, i \in \mathcal{N})$	$q := (q_i, i \in \mathcal{N})$
$\ell := (\ell_i, i \in \mathcal{E})$	$P := (P_i, i \in \mathcal{E})$	$Q := (Q_i, i \in \mathcal{E})$

We adopt a branch flow model first proposed in [14], [15] for radial networks. It ignores the phase angles of voltages and currents and uses only the variables $x := (v, \ell, P, Q, p, q)$. Compared with bus injection model, branch flow model is more numerical stable and has broad application in distribution network, [14]–[16]. Given the network $\mathcal{T}$, the branch flow model is defined by:

$$v_{A_i} = v_i - 2(r_i P_i + x_i Q_i) + \ell_i (r_i^2 + x_i^2) \quad i \in \mathcal{E} \quad \text{(1a)}$$

$$\sum_{j \in C_i} (P_j - \ell_j r_j) + p_i = P_i \quad i \in \mathcal{N} \quad \text{(1b)}$$

$$\sum_{j \in C_i} (Q_j - \ell_j x_j) + q_i = Q_i \quad i \in \mathcal{N} \quad \text{(1c)}$$

$$P_i^2 + Q_i^2 = v_i \ell_i \quad i \in \mathcal{E} \quad \text{(1d)}$$

where $P_0, Q_0 = 0$ for ease of presentation. Given a vector x that satisfies (1), the phase angles of the voltages and currents can be uniquely determined if the network is a tree. Hence this (relaxed) model (1) is equivalent to a full AC power flow model. See [4, Section III-A] for details.

B. OPF and SOCP relaxation

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

$$F(x) = \sum_{i \in \mathcal{N}} f_i(p_i, \ell_i) \quad \text{(2)}$$

For instance, to minimize total power loss, we can set $f_0(p_0) = 0$ and $f_i(p_i, \ell_i) = \ell_i r_i$ for each $i \in \mathcal{N}_+$.

We consider two operational constraints. First, the power injection at each bus $i \in \mathcal{N}_+$ is constrained to be in a region $\mathcal{S}_i$, i.e.

$$(p_i, q_i) \in \mathcal{S}_i \quad \text{(3a)}$$

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 $\mathcal{S}_i = \{(p, q) \in \mathbb{R}^2 \mid p \in [\underline{p}_i, \overline{p}_i], q \in [\underline{q}_i, \overline{q}_i]$.

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

$$\underline{v}_i \le v_i \le \overline{v}_i \quad \text{(3b)}$$

Typically the voltage magnitude is allowed to deviate by 5% from its nominal value, i.e. $\underline{v}_i = 0.95^2$ and $\overline{v}_i = 1.05^2$.

The OPF problem is:

$$\text{OPF:} \quad \min_x \quad \sum_{i \in \mathcal{N}} f_i(p_i, \ell_i) \quad \text{(4)}$$
$$\text{s.t.} \quad \text{(1) and (3)}$$

The OPF problem (4) is nonconvex due to the equality (1d). This is relaxed to inequality in [3], [4]

$$P_i^2 + Q_i^2 \le v_i \ell_i \quad i \in \mathcal{N}_+ \quad \text{(5)}$$

resulting in a (convex) second-order cone program (SOCP):

$$\text{ROPF:} \quad \min_x \quad \sum_{i \in \mathcal{N}} f_i(p_i, \ell_i) \quad \text{(6)}$$
$$\text{s.t.} \quad \text{(1a)} - \text{(1c)}, \text{(5) and (3)}$$

Clearly the relaxation ROPF (6) provides a lower bound for the original OPF problem (4) since the original feasible set is enlarged. The relaxation is called *exact* if every optimal solution of ROPF attains equality in (5) and hence is also

optimal for the original OPF. For network with tree topology, SOCP relaxation is exact under some mild conditions [4], [17].

III. Distributed Algorithm for OPF

We assume SOCP relaxation is exact and develop in this section a distributed algorithm that solves ROPF. We first review a standard alternating direction method of multiplier (ADMM). We then make use of the structure of ROPF to speed up the standard ADMM algorithm by deriving close form expressions for the optimization subproblems in each ADMM iteration.

A. Preliminary: ADMM

ADMM blends the decomposability of dual decomposition with the superior convergence properties of the method of multipliers [12]. It solves optimization problem of the form:

$$\begin{array}{ll} \min\limits_{x,z} & f(x)+g(z) \\ \text{s.t.} & x\in\mathcal{K}_x,\quad z\in\mathcal{K}_z \\ & Ax+Bz=c \end{array} \tag{7}$$

where $\mathcal{K}_x, \mathcal{K}_z$ are convex sets. Let λ denote the Lagrange multiplier for the constraint $Ax+Bz=c$. Then the augmented Lagrangian is defined as

$$\begin{aligned} L_\rho(x,z,\lambda) \;:=\; & f(x)+g(z)+\lambda^T(Ax+Bz-c) \\ & +\frac{\rho}{2}\|Ax+Bz-c\|^2, \end{aligned}$$

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

$$x^{k+1} \;\in\; \arg\min_{x\in\mathcal{K}_x} L_\rho(x,z^k,\lambda^k) \tag{8a}$$

$$z^{k+1} \;\in\; \arg\min_{z\in\mathcal{K}_z} L_\rho(x^{k+1},z,\lambda^k) \tag{8b}$$

$$\lambda^{k+1} \;=\; \lambda^k+\rho(Ax^{k+1}+Bz^{k+1}-c) \tag{8c}$$

Compared to dual decomposition, ADMM is guaranteed to converge to an optimal solution under less restrictive conditions. Let

$$r^k \;:=\; \|Ax^k+Bz^k-c\| \tag{9a}$$

$$s^k \;:=\; \rho\|A^TB(z^k-z^{k-1})\| \tag{9b}$$

They can be viewed as the residuals for primal and dual feasibility. Under mild conditions, it can be shown that

$$\lim_{k\to\infty} r^k=0,\quad \lim_{k\to\infty} s^k=0$$

implying

$$\lim_{k\to\infty} f(x^k)+g(z^k)=p^*$$

One can refer to [12] for details.

B. Decoupling in ADMM

In applying ADMM to ROPF, we first exploit the structure of ROPF to derive subproblems that are decoupled and can be solved concurrently. We then derive closed form solutions to these subproblems. In this subsection we explain the standard idea of decoupling through local variables, which we will use in the next subsection.

Consider the problem:

$$\min_{x} \quad f(x) \tag{10a}$$

$$\text{s.t.} \quad a_i^Tx=b_i,\quad i\in\mathcal{I} \tag{10b}$$

$$x\in\mathcal{K}_z \tag{10c}$$

where $f(x)$ is a convex function and $\mathcal{K}_z$ is a convex set. The variable x must satisfies the linear constraints (10b) for all $i\in\mathcal{I}$ as well as be in $\mathcal{K}_z$. As we will see below, for speedup, we wish to make the constraints (10b) local so that the update (8) can be decomposed into several small optimization subproblems that can be solved simultaneously. To this end we create local copies of x and compute them in parallel. Each copy satisfies a different subset of the constraints before the algorithm converges. At optimality, all the local copies are required to be equal and hence satisfy all the constraints.

Formally, let $\{\mathcal{I}_l, 1\leq l\leq m\}$ be a partition of $\mathcal{I}$, i.e. $\mathcal{I}_l$ are disjoint and $\cup_{1\leq l\leq m}\mathcal{I}_l=\mathcal{I}$. There are $m+1$ constraints defined by the sets $\mathcal{K}_z$ and

$$\mathcal{K}_x^l \;:=\; \{x\in\mathbb{R}^n \mid a_i^Tx=b_i, i\in\mathcal{I}_l\},\quad 1\leq l\leq m$$

Consider the $m+1$ copies $(z,(x^{(l)},1\leq l\leq m))$ of the original variable. The decoupled version of (10) is:

$$\begin{array}{ll} \min\limits_{x^{(l)},z} & f(z) \\ \text{s.t.} & z\in\mathcal{K}_z \\ & x^{(l)}\in\mathcal{K}_x^l,\quad 1\leq l\leq m \\ & x^{(l)}-z=0,\quad 1\leq l\leq m \end{array} \tag{11}$$

Let $x:=(x^{(l)},1\leq l\leq m)$ denote the variable obtained by stacking all vectors $x^{(l)}, 1\leq l\leq m$. Relax the last equality for each l. Let $\lambda^{(l)}$ denote the corresponding Lagrange multipliers and $\lambda:=(\lambda^{(l)},1\leq l\leq m)$. Then the augmented Lagrangian is

$$L_\rho(x,z,\lambda):=f(z)+\sum_{l=1}^m\left((\lambda^{(l)})^T(x^{(l)}-z)+\frac{\rho}{2}\|x^{(l)}-z\|^2\right)$$

We can update the primal variables (x,z) and the multipliers λ according to (8).

Next, we show that two partitions $(m=2)$ are sufficient for designing distributed OPF algorithm using this approach.

C. Distributed OPF algorithm

We now derive a distributed algorithm for solving ROPF (6) that has the following advantages:

- Each bus only needs to solves a local subproblem in each iteration of (8). Moreover there is a closed form solution for each subproblem, in contrast to most

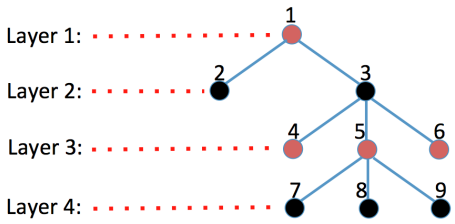


Fig. 2: Example of a graph with tree topology.

algorithms that employ iterative procedure to solve each subproblem [7]–[9], [11], [18]–[21].

- Communication is only required between adjacent buses.

We can write the ROPF problem (6) in the form of (10) as:

$$\begin{aligned} \min_{x} \quad & \sum_{i\in\mathcal{N}} f_i(p_i,\ell_i) \\ \text{s.t.} \quad & x \text{ satisfies (1a)} - \text{(1c)} \\ & x \in \mathcal{K}_z \end{aligned}$$

where $\mathcal{K}_z := \{z \mid z \text{ satisfies (3) and (5)}\}$. Next, we partition the constraints in (1a)-(1c) to write it equivalently in the form of (11) such that the update in (8) can be done simultaneously by each bus.

We assume each bus $i \in \mathcal{N}$ is an agent with its local variables $x_i := (v_i, \ell_i, P_i, Q_i, p_i, q_i)$. Then the constraints in $\mathcal{K}_z$ are local, i.e. they are separable for each agent i. Note that the network $\mathcal{T}$ is a tree, which is a bipartite graph and can be partitioned into two groups $\mathcal{I}_1 := \{i \in \mathcal{N} \mid i \text{ is in the odd layer}\}$ and $\mathcal{I}_2 := \{i \in \mathcal{N} \mid i \text{ is in the even layer}\}$. For instance, there are four layers in Fig. 2 and $\mathcal{I}_1 = \{1,4,5,6\}$ and $\mathcal{I}_2 = \{2,3,7,8,9\}$. Let

$$\mathcal{K}_x^l = \{x_i^{(l)} \text{ satisfies (1a)-(1c) for } i \in \mathcal{I}_l\}, \quad l = 1,2$$

Under such partition, bus $i \in \mathcal{I}_1$ ($\mathcal{I}_2$) is only coupled with buses $k \in \mathcal{I}_2$ ($\mathcal{I}_1$). In particular, by (1a), each bus i needs the voltage v_{A_i} from its ancestor A_i. Thus we create a copy $v_{A_i,i}$, representing the replication of v_{A_i} at bus i. On the other hand, in (1b) and (1c), each bus i needs ℓ_j, P_j, Q_j from all of its children $j \in C_i$ and we create a copy $\ell_{j,i}, P_{j,i}, Q_{j,i}$ of each $j \in C_i$ at bus i. Then the ROPF problem (6) can be written in the form of (11).

E-ROPF:

$$\begin{aligned} \min_{x,z} \quad & \sum_{i\in\mathcal{N}} f_i(p_i^{(z)},\ell_i^{(z)}) \\ \text{s.t.} \quad & v_{A_i,i}^{(l)} = v_i^{(l)} - 2(r_iP_i^{(l)} + x_iQ_i^{(l)}) + \ell_i^{(l)}(r_i^2+x_i^2) \quad \text{(12a)} \\ & \qquad i \in \mathcal{E} \quad l \in \{1,2\} \\ & \sum_{j\in C_i}(P_{j,i}^{(l)} - \ell_{j,i}^{(l)} r_j) + p_i^{(l)} = P_i^{(l)} \quad i \in \mathcal{N} \; l \in \{1,2\} \quad \text{(12b)} \\ & \sum_{j\in C_i}(Q_{j,i}^{(l)} - \ell_{j,i}^{(l)} x_j) + q_i^{(l)} = Q_i^{(l)} \quad i \in \mathcal{N} \; l \in \{1,2\} \quad \text{(12c)} \\ & (P_i^{(z)})^2 + (Q_i^{(z)})^2 \le v_i^{(z)}\ell_i^{(z)} \quad i \in \mathcal{E} \quad \text{(12d)} \\ & (p_i^{(z)}, q_i^{(z)}) \in \mathcal{S}_i \quad i \in \mathcal{N} \quad \text{(12e)} \\ & \underline{v}_i \le v_i^{(z)} \le \overline{v}_i \quad i \in \mathcal{N} \quad \text{(12f)} \\ & x^{(l)} - z = 0 \quad l \in \{1,2\} \quad \text{(12g)} \end{aligned}$$

where (12a)-(12c) form $(\mathcal{K}_x^{(l)}, l = 1,2)$ and (12d) – (12f) form $\mathcal{K}_z$. The value of the superscript l depends on the partition that i belongs to, i.e. $l = 1$ (2) if $i \in \mathcal{I}_1$ ($\mathcal{I}_2$). Let λ, γ and μ be Lagrangian multipliers associated with $x^{(l)} - z = 0$, specifically

$\lambda_{1,i}$:	$v_i^{(l)} - v_i^{(z)} = 0$	$\lambda_{2,i}$:	$\ell_i^{(l)} - \ell_i^{(z)} = 0$
$\lambda_{3,i}$:	$P_i^{(l)} - P_i^{(z)} = 0$	$\lambda_{4,i}$:	$Q_i^{(l)} - Q_i^{(z)} = 0$
$\lambda_{5,i}$:	$p_i^{(l)} - p_i^{(z)} = 0$	$\lambda_{6,i}$:	$q_i^{(l)} - q_i^{(z)} = 0$
$\mu_{1,i}$:	$\ell_{i,A_i}^{(l)} - \ell_i^{(z)} = 0$	$\mu_{2,i}$:	$P_{i,A_i}^{(l)} - P_i^{(z)} = 0$
$\mu_{3,i}$:	$Q_{i,A_i}^{(l)} - Q_i^{(z)} = 0$		
γ_j:	$v_{i,j}^{(l)} - v_i^{(z)} = 0$ for $j \in C_i$		

Denote

$$\begin{aligned} x_i^{(l)} &:= (v_i^{(l)}, \ell_i^{(l)}, P_i^{(l)}, Q_i^{(l)}, p_i^{(l)}, q_i^{(l)}) \\ z_i &:= (v_i^{(z)}, \ell_i^{(z)}, P_i^{(z)}, Q_i^{(z)}, p_i^{(z)}, q_i^{(z)}) \\ x_{i,A_i}^{(l)} &:= (\ell_{i,A_i}^{(l)}, P_{i,A_i}^{(l)}, Q_{i,A_i}^{(l)}) \\ \lambda_i &:= (\lambda_{k,i} \mid k = 1,2,3,4,5,6) \\ \mu_i &:= (\mu_{k,i} \mid k = 1,2,3) \end{aligned}$$

Then the variables maintained by each bus (agent) i are:

$$\mathcal{A}_i := \{x_i^{(l)}, x_{i,A_i}^{(l)}, \{v_{i,j}^{(l)}, \gamma_j \mid j \in C_i\}, z_i, \lambda_i, \mu_i, \}.$$

Next, we demonstrate that the E-ROPF problem (12) can be solved in a distributed manner, i.e. both the x-update (8a) and z-update (8b) can be decomposed into small subproblems that can be solved simultaneously. For ease of presentation, we remove the iteration number k in (8) for all the variables, which will be updated accordingly after each subproblem is solved. The augmented Lagrangian for modified ROPF problem is given in (13). We abuse notations in (13) and denote $x_{i,A_i}^{(l)} - z_i := (\ell_{i,A_i}^{(l)} - \ell_i^{(z)}, P_{i,A_i}^{(l)} - P_i^{(z)}, Q_{i,A_i}^{(l)} - Q_i^{(z)})$ although there are 3 entries in $x_{i,A_i}^{(l)}$ but 6 entries in z_i. By (13c), in the x-update step (8a), we solve

$$\arg\min_{x\in\mathcal{K}_x} L_\rho(x,z,\lambda,\gamma,\mu) = \arg\min_{x\in\mathcal{K}_x} \sum_{i\in\mathcal{N}} G_i(x^{(l)}),$$

where

$$G_i(x^{(l)}) := \lambda_i^T x_i^{(l)} + \sum_{j\in C_i} \mu_j^T x_{j,i}^{(l)} + \gamma_i v_{A_i,i}^{(l)} +$$

$$\frac{\rho}{2}\left(\left\|x_i^{(l)} - z_i\right\|^2 + \sum_{j\in C_i}\left\|x_{j,i}^{(l)} - z_j\right\|^2 + \left(v_{A_i,i}^{(l)} - v_{A_i}^{(z)}\right)^2\right)$$

For each agent i, the corresponding subproblem is

$$\begin{aligned} \min \; & G_i(x^{(l)}) \quad (14) \\ \text{s.t.} \; & v_{A_i,i}^{(l)} = v_i^{(l)} - 2\left(r_iP_i^{(l)} + x_iQ_i^{(l)}\right) + \ell_i^{(l)}\left(r_i^2+x_i^2\right) \\ & \sum_{j\in C_i}\left(P_{j,i}^{(l)} - \ell_{j,i}^{(l)} r_j\right) + p_i^{(l)} = P_i^{(l)} \\ & \sum_{j\in C_i}\left(Q_{j,i}^{(l)} - \ell_{j,i}^{(l)} x_j\right) + q_i^{(l)} = Q_i^{(l)} \end{aligned}$$

which takes the following form

$$\min_x \quad \frac{\rho}{2}\|x\|_2^2 + c^T x \quad \text{s.t. } Bx = 0 \quad (15)$$

$$L_\rho(x,z,\lambda,\gamma,\mu) \tag{13a}$$

$$= \sum_{i\in\mathcal{N}} \left(f_i\left(p_i^{(z)}\right) + \lambda_i^T\left(x_i^{(l)} - z_i\right) + \mu_i^T\left(x_{i,A_i}^{(l)} - z_i\right) + \sum_{j\in C_i}\gamma_j\left(v_{i,j}^{(l)} - v_i^{(z)}\right) + \frac{\rho}{2}\left(\left\|x_i^{(l)} - z_i\right\|^2 + \left\|x_{i,A_i}^{(l)} - z_i\right\|^2 + \sum_{j\in C_i}\left(v_{i,j}^{(l)} - v_i^{(z)}\right)^2\right)\right) \tag{13b}$$

$$= \sum_{i\in\mathcal{N}} \left(f_i\left(p_i^{(z)}\right) + \lambda_i^T\left(x_i^{(l)} - z_i\right) + \sum_{j\in C_i}\mu_j^T\left(x_{j,i}^{(l)} - z_j\right) + \gamma_i\left(v_{A_i,i}^{(l)} - v_{A_i}^{(z)}\right) + \frac{\rho}{2}\left(\left\|x_i^{(l)} - z_i\right\|^2 + \sum_{j\in C_i}\left\|x_{j,i}^{(l)} - z_j\right\|^2 + \left(v_{A_i,i}^{(l)} - v_{A_i}^{(z)}\right)^2\right)\right) \tag{13c}$$

whose close form solution is given as

$$x = \frac{1}{\rho}(B^T(BB^T)^{-1}Bc - \frac{1}{\rho}c$$

Prior to performing the update, each agent i needs to request variables from its parent A_i and children $j \in C_i$. In particular, it needs to get $v_{A_i,i}^{(l)}$, $v_{A_i}^{(z)}$ and γ_i from its ancestor A_i and $x_{j,i}^{(l)}, z_j, \mu_j$ from all of its children $j \in C_i$. After the update, it needs to send the updated variables back to its ancestor A_i and its children $j \in C_i$. As we will see, only the x update requires communication with neighbors.

Based on (13b), in the z-update step, we solve

$$\arg\min_{z\in\mathcal{K}_z} L_\rho(x,z,\lambda,\gamma,\mu) = \arg\min_{z\in\mathcal{K}_z}\sum_{i\in\mathcal{N}} H_i(z),$$

where

$$H_i(z) := f\left(p_i^{(z)}, \ell_i^{(z)}\right) - \lambda_i^T z_i - \mu_i^T z_i - \sum_{j\in C_i}\gamma_j v_i^{(z)} +$$

$$\frac{\rho}{2}\left(\left\|x_i^{(l)} - z_i\right\|^2 + \left\|x_{i,A_i}^{(l)} - z_i\right\|^2 + \sum_{j\in C_i}\left(v_{i,j}^{(l)} - v_i^{(z)}\right)^2\right)$$

The subproblem solved by each agent i is

$$\begin{aligned} \min \quad & H_i(z) \\ \text{s.t.} \quad & (P_i^{(z)})^2 + (Q_i^{(z)})^2 \le v_i^{(z)}\ell_i^{(z)} \\ & \underline{v}_i \le v_i^{(z)} \le \overline{v}_i \\ & \left(p_i^{(z)}, q_i^{(z)}\right) \in \mathcal{S}_i \end{aligned}$$

Suppose $f_i(p_i^{(z)}, \ell_i^{(z)})$ is linear or quadratic in its argument. Let $\kappa = (|C_i|+1)^{-\frac{1}{2}}$ and scale $v_i^{(z)}$ down by κ in the above problem. Then it takes the following form:

$$\begin{aligned} \min_{y} \quad & \sum_{i=1}^{6}(y_i^2 + c_i y_i) \\ \text{s.t.} \quad & y_1^2 + y_2^2 \le \kappa^2 y_3 y_4 \\ & \underline{y}_3 \le y_3 \le \overline{y}_3 \\ & (y_5, y_6) \in \mathcal{S}_i \end{aligned} \tag{16}$$

Note that $(y_1,\ldots,y_4)$ and (y_5,y_6) are independent in the optimization problem (16). Thus we have two independent subproblems. The first subproblem solves

$$\min_{y_5,y_6} \sum_{i=5}^{6}(y_i^2 + c_i y_i) \text{ s.t. } (y_5,y_6) \in \mathcal{S}_i$$

which determines the update of (y_5, y_6). The solution is

$$(y_5, y_6) = \left(\left[\frac{-c_5}{2}\right]_{\underline{p}_i}^{\overline{p}_i}, \left[\frac{-c_6}{2}\right]_{\underline{q}_i}^{\overline{q}_i}\right),$$

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

$$\begin{aligned} \min_{y_1,y_2,y_3,y_4} \quad & \sum_{i=1}^{4}(y_i^2 + c_i y_i) \\ \text{s.t.} \quad & y_1^2 + y_2^2 \le k^2 y_3 y_4 \\ & \underline{y}_3 \le y_3 \le \overline{y}_3 \end{aligned}$$

which determines the update of $(y_1,\ldots,y_4)$. Due to space limitation, we will not derive the close form expression here and interested readers may refer to [22, Appendix I]. After the z-update, we update the Lagrange multipliers for the relaxed constraints as (8c). Both the z-update and multiplier update steps only involve local variables of an agent and no communication is required.

Finally, we specify the stopping criteria for the algorithm. Empirical results show that the the solution is accurate enough when both the primal residual r^k defined in (9a) and the dual residual s^k defined in (9b) are below $10^{-4}\sqrt{N}$, where N is the number of buses. The pseudo code for the algorithm is summarized in Table I.

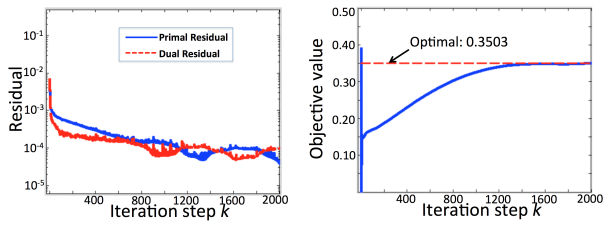
TABLE I: Distributed algorithm of OPF

Distributed Algorithm of OPF
Input: network $\mathcal{T}$, power injection region $\mathcal{S}_i$, voltage region $(\underline{v}_i, \overline{v}_i)$, line impedance z_i **Output**: voltage v, power injection s
1. Initialize the variables with any number. 2. Iterate the following step until both the primal residual s^k (9a) and the dual residual r^k (9b) are below $10^{-4}\sqrt{N}$. a. x-update: each agent i solves (14) to update x. b. z-update: each agent i solves (16) to update z. c. multiplier update: update λ, μ, γ by (8c).

IV. Case Study

To demonstrate the scalability of the distributed algorithm proposed in section III, we test it on the model of a 2,065-bus distribution circuit in the service territory of the Southern California Edison. 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 balanced and consider a single phase network. The voltage magnitude at each load bus is allowed within $[0.95, 1.05]$ per unit (pu), i.e. $\overline{v}_i = 1.05^2$ and $\underline{v}_i = 0.95^2$ for $i \in \mathcal{N}_+$. The control devices are the rooftop PV panels whose reactive power injections are controlled. The objective is to minimize power loss across the network,



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

Fig. 3: Simulation results for 2065 bus Rossi circuit.

namely $f_i(p_i, \ell_i) = \ell_i r_i$ for $i \in \mathcal{N}_+$ in (2). Each bus is an agent and there are 2,065 agents in the network that solve the OPF problem in a distributed manner.

The algorithm is implemented in Matlab 2013a and run on Macbook pro 2013 with i5 dual core processor. We mainly focus on the following aspects:

- Solution feasibility: the primal residual r^k defined in (9a) measures the feasibility of the solution for ADMM [12]. In our algorithm, (12g) are relaxed and $r^k = \sqrt{\|(x^{(1)})^k - z^k\|^2 + \|(x^{(2)})^k - z^k\|^2}$ with respect to the iterations k.
- Optimality: the dual feasibility error s^k defined in (9b) measures the optimality of the solution for ADMM [12]. In our algorithm, the dual residual $s^k = \sqrt{2}\rho\|z^k - z^{k-1}\|$ with respect to the iterations k.
- Computation time: the proposed distributed algorithm is run on a single machine. We can divide the total time by the number of agents to roughly estimate the time required for each agent if the algorithm is run on distributed severs (excluding communication overhead).

The stopping criteria is that both the primal and dual residual are below $10^{-4}\sqrt{N}$ and Figure 3a illustrates the evolution of $r^k/\sqrt{N}$ and $s^k/\sqrt{N}$ over 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 average time spent by each agent is roughly 0.56s (excluding communication overhead) if we implement the algorithm in a distributed manner.

Moreover, we show the advantage of deriving close form expression by comparing the computation time of solving the subproblems between off-the-shelf solver (CVX) and our algorithm. In particular, we compare the average computation time of solving the subproblem in both the x and z update. In the x update, the average time required to solve the subproblem is 1.7×10^{-4}s for our algorithm but 0.2s for CVX. In the z update, the average time required to solve the subproblem is 5.1×10^{-4}s for our algorithm but 0.3s for CVX. Thus, each ADMM iteration takes about 6.8×10^{-4}s for our algorithm but 0.5s for using iterative algorithm.

V. Conclusion

In this paper, we have developed a distributed algorithm for optimal power flow problem based on alternating direction method of multiplier. We have derived a close form solution for the subproblems solved by each agent thus significantly reducing the computation time. Preliminary simulation shows that the algorithm is scalable to a 2,065-bus system.

References

[1] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, "Semidefinite programming for optimal power flow problems," *Int'l J. of Electrical Power & Energy Systems*, vol. 30, no. 6-7, pp. 383–392, 2008.

[2] R. Jabr, "Radial Distribution Load Flow Using Conic Programming," *IEEE Trans. on Power Systems*, vol. 21, no. 3, pp. 1458–1459, Aug 2006.

[3] M. Farivar, C. R. Clarke, S. H. Low, and K. M. Chandy, "Inverter var control for distribution systems with renewables," in *Proceedings of IEEE SmartGridComm Conference*, October 2011.

[4] M. Farivar and S. H. Low, "Branch flow model: relaxations and convexification (parts I, II)," *IEEE Trans. on Power Systems*, vol. 28, no. 3, pp. 2554–2572, August 2013.

[5] S. H. Low, "Convex relaxation of optimal power flow, I: formulations and relaxations," *IEEE Trans. on Control of Network Systems*, vol. 1, no. 1, pp. 15–27, March 2014.

[6] ——, "Convex relaxation of optimal power flow, II: exactness," *IEEE Trans. on Control of Network Systems*, 2014, to appear.

[7] B. H. Kim and R. Baldick, "Coarse-grained distributed optimal power flow," *Power Systems, IEEE Transactions on*, vol. 12, no. 2, pp. 932–939, 1997.

[8] A. Lam, B. Zhang, and D. N. Tse, "Distributed algorithms for optimal power flow problem," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 430–437.

[9] E. Dall'Anese, H. Zhu, and G. B. Giannakis, "Distributed optimal power flow for smart microgrids," *arXiv preprint arXiv:1211.5856*, 2012.

[10] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, "Dynamic network energy management via proximal message passing," *Optimization*, vol. 1, no. 2, pp. 1–54, 2013.

[11] A. X. Sun, D. T. Phan, and S. Ghosh, "Fully decentralized ac optimal power flow algorithms," in *Power and Energy Society General Meeting (PES), 2013 IEEE*. IEEE, 2013, pp. 1–5.

[12] 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, pp. 1–122, 2011.

[13] M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

[14] M. E. Baran and F. F. Wu, "Optimal Capacitor Placement on radial distribution systems," *IEEE Trans. Power Delivery*, vol. 4, no. 1, pp. 725–734, 1989.

[15] ——, "Optimal Sizing of Capacitors Placed on A Radial Distribution System," *IEEE Trans. Power Delivery*, vol. 4, no. 1, pp. 735–743, 1989.

[16] Q. Peng, Y. Tang, and S. H. Low, "Feeder reconfiguration in distribution networks based on convex relaxation of opf," *IEEE Transaction on Power System*, 2014.

[17] L. Gan, N. Li, U. Topcu, and S. Low, "Exact convex relaxation for optimal power flow in radial networks," *IEEE Transaction on Automatic Control*, 2014.

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

[19] E. Devane and I. Lestas, "Stability and convergence of distributed algorithms for the opf problem," in *52nd IEEE Conference on Decision and Control*, 2013.

[20] N. Li, L. Chen, and S. H. Low, "Demand response in radial distribution networks: Distributed algorithm," in *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*. IEEE, 2012, pp. 1549–1553.

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

[22] Q. Peng and S. H. Low, "Distributed algorithm for optimal power flow on a radial network," *arXiv preprint arXiv:1404.0700*, 2014.