Distributed Optimal Power Flow Algorithm for Balanced Radial Distribution Networks

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 in distribution networks 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 *balanced* radial distribution 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.

Index Terms

Power Distribution, Nonlinear systems, Optimal Power Flow, Distributed 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 the OPF problems 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. 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 distribution 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 simultaneously optimize (possibly real-time) 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 distribution networks.

Various distributed algorithms have been developed to solve the OPF problem. Through optimization decomposition, the original OPF problem is decomposed into several local subproblems that can be solved simultaneously. Some distributed algorithms do not deal with the non convexity issue 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 are proposed to guarantee convergence, *e.g.* dual decomposition method [11], [12], auxiliary variable method [13], [14] and ADMM [15], [16].

One of the key performance metrics of a distributed algorithm is the time of convergence (ToC), which is the product of the number of iterations and the computation time to solve the subproblems in each iteration. To 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 will improve ToC by reducing the computation time for each subproblem.

Specifically we develop a scalable distributed algorithm through decomposing the *convexified* OPF problem into smaller subproblems based on alternating direction method of multiplier (ADMM). ADMM blends the decomposability of dual

A preliminary version has appeared in [1].

Qiuyu Peng is with the Electrical Engineering Department and Steven H. Low is with the Computing and Mathematical Sciences and the Electrical Engineering Departments, California Institute of Technology, Pasadena, CA 91125, USA. {qpeng, slow}@caltech.edu

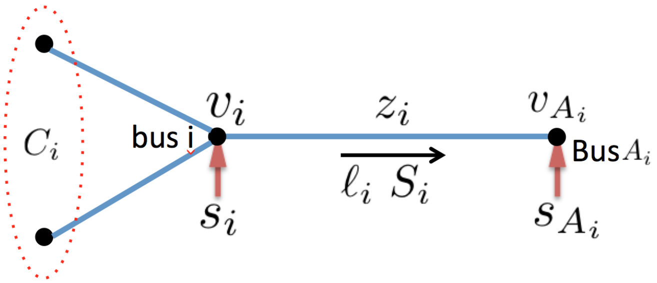


Fig. 1: Notations.

decomposition and superior convergence properties of the method of multipliers [17]. It has broad applications in different areas and particularly useful when the subproblems can be solved efficiently [18], for example when they admit closed form expressions, *e.g.* matrix factorization [19], image recovery [20].

The proposed algorithm has two advantages: 1) There is closed form solution for each optimization subproblem, 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-life network. In particular, we show that the algorithm converges within 0.6s for a 2,065-bus system. To show the superiority of deriving close form expression of each subproblems, finally we compare the computation time for solving a subproblem by our algorithm and an off-the-shelf optimization solver (CVX, [21]). Our solver requires on average 6.8×10^{-4}s while CVX requires on average 0.5s. On the other hand, we also show that the convergence rate is mainly determined by the diameter [1] of the network through simulating the algorithm on different networks.

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 this paper in section V.

II. Problem Formulation

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

We denote the set of complex numbers with $\mathbb{C}$, the set of n-dimensional complex numbers with $\mathbb{C}^n$. 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 $\langle x, y\rangle := \mathbf{Re}(tr(x^H y))$. The Euclidean norm of a vector $x \in \mathbb{C}^n$ is defined as $\|x\|_2 := \sqrt{\langle x, x\rangle}$.

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 $\mathcal{N}$. 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.

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 [22], [23] for radial network. It has better numerical stability than bus injection model and has been advocated for the design and operation for radial distribution network, [5], [14], [24]. It ignores 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

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

defined by:

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

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

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

where $S_0 = 0$ (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, Section 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:

$$F(p) = \sum_{i \in \mathcal{N}} f_i(p_i) := \sum_{i \in \mathcal{N}} \left(\frac{\alpha_i}{2} p_i^2 + \beta_i p_i \right) \tag{2}$$

where $\alpha_i, \beta_i \geq 0$. For instance,

- to minimize total line loss, we can set $\alpha_i = 0$ and $\beta_i = 1$ for each bus $i \in \mathcal{N}$.
- to minimize generation cost, we can set $\alpha_i = 0$ and $\beta_i = 0$ for bus i where there is no generator and for generator bus i, the corresponding α_i, β_i depends on the characteristic of the generator.

We consider two operational constraints. First, the power injection s_i at each bus i is constrained to be 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

$$\mathcal{I}_i = \{p + \mathbf{i}q \mid p \in [\underline{p}_i, \overline{p}_i], q \in [\underline{q}_i, \overline{q}_i]\} \subseteq \mathbb{C} \tag{4a}$$

- For solar panel connecting the grid through a inverter with nameplate $\overline{s}_i$, the injection region $\mathcal{I}_i$ is

$$\mathcal{I}_i = \{p + \mathbf{i}q \mid p \geq 0, p^2 + q^2 \leq \overline{s}_i^2\} \subseteq \mathbb{C} \tag{4b}$$

Second, the voltage magnitude at each bus $i \in \mathcal{N}$ needs to be maintained 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

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

The OPF problem (6) is nonconvex due to the quadratic equality constraint (1c). In [4], [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)

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

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], [24].

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 closed 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 [17]. For our application, we consider optimization problems of the form:

$$\begin{array}{rl} \min & f(x)+g(z) \\ \text{over} & x\in\mathcal{K}_x, \quad z\in\mathcal{K}_z \\ \text{s.t.} & x=z \end{array} \tag{9}$$

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

$$L_\rho(x,z,\lambda) := f(x)+g(z)+\langle \lambda, x-z\rangle + \frac{\rho}{2}\|x-z\|_2^2, \tag{10}$$

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 iterations:

$$x^{k+1} \in \arg\min_{x\in\mathcal{K}_x} L_\rho(x, z^k, \lambda^k) \tag{11a}$$

$$z^{k+1} \in \arg\min_{z\in\mathcal{K}_z} L_\rho(x^{k+1}, z, \lambda^k) \tag{11b}$$

$$\lambda^{k+1} = \lambda^k + \rho(x^{k+1}-z^{k+1}) \tag{11c}$$

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

$$r^k := \|x^k - z^k\|_2 \tag{12a}$$

$$s^k := \rho\|z^k - z^{k-1}\|_2 \tag{12b}$$

which can be viewed as the residuals for primal and dual feasibility. Assume:

- A1: f and g are closed proper and convex.
- A2: The unaugmented Lagrangian L_0 has a saddle point.

The correctness of ADMM is guaranteed by the following result; see [17, Chapter 3].

Proposition 3.1 ([17]): Suppose A1 and A2 hold. Let p^* be the optimal objective value. Then

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

and

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

B. Apply ADMM to OPF problem

We assume the SOCP relaxation is exact and now derive a distributed algorithm for solving ROPF (8) that has the following advantages:

- Each bus only needs to solves a local subproblem in each iteration of (11). Moreover there is a closed form solution for each subproblem, in contrast to most algorithms that employ iterative procedure to solve these subproblems [8]–[15].
- Communication is only required between adjacent buses.

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

$$\min \sum_{i\in\mathcal{N}} f_i(p_i) \tag{13a}$$

$$\text{over} \quad v, s, S, \ell \tag{13b}$$

$$\text{s.t.} \quad v_{A_i} - v_i + z_i S_i^* + S_i z_i^* - \ell_i |z_i|^2 = 0 \qquad i\in\mathcal{E} \tag{13c}$$

$$\sum_{i\in C_i} (S_j - z_j \ell_j) - S_i + s_i = 0 \qquad i\in\mathcal{N} \tag{13d}$$

$$|S_i|^2 \leq v_i \ell_i \qquad i\in\mathcal{E} \tag{13e}$$

$$s_i \in \mathcal{I}_i \qquad i\in\mathcal{N} \tag{13f}$$

$$\underline{v}_i \leq v_i \leq \overline{v}_i \qquad i\in\mathcal{N} \tag{13g}$$

TABLE I: Multipliers associated with constraints (14g)-(14h)

$\lambda_{1,i}$:	$S_i^{(x)} = S_i^{(z)}$	$\lambda_{2,i}$:	$\ell_i^{(x)} = \ell_i^{(z)}$
$\lambda_{3,i}$:	$v_i^{(x)} = v_i^{(z)}$	$\lambda_{4,i}$:	$s_i^{(x)} = s_i^{(z)}$
$\mu_{1,i}$:	$S_{i,A_i}^{(x)} = S_i^{(z)}$	$\mu_{2,i}$:	$\ell_{i,A_i}^{(x)} = \ell_i^{(z)}$
γ_i:	$v_{A_i,i}^{(x)} = v_{A_i}^{(z)}$		

Assume each bus i is an agent that maintains local variables (v_i, s_i, S_i, ℓ_i). Then (13e)–(13g) are local constraints to agent (bus) i. (13c) and (13d) describe the coupling constraints among i and its parent A_i and the set of children in C_i, *i.e.* (13c) models the voltage of its ancestor A_i as a function of the local variables of i, (13d) describes the power flow balance among the set of children C_i and bus i itself. To decouple the constraints (13c)–(13d), for each bus i, its ancestor A_i sends its voltage v_{A_i} to i, denoted by $v_{A_i,i}$ and each child $j \in C_i$ sends the branch power to i, denoted by $S_{j,i}$ and current to i, denoted by $\ell_{j,i}$. Then ROPF can be written equivalently as follows.

$$
\begin{aligned}
\min \quad & \sum_{i\in\mathcal{N}} f_i(p_i^{(z)}) && && \text{(14a)}\\
\text{over} \quad & x := \{v_i^{(x)}, s_i^{(x)}, S_i^{(x)}, \ell_i^{(x)}, v_{A_i,i}^{(x)}, S_{i,A_i}^{(x)}, \ell_{i,A_i}^{(x)}, i \in \mathcal{N}\} && &&\\
& z := \{v_i^{(z)}, s_i^{(z)}, S_i^{(z)}, \ell_i^{(z)}, i \in \mathcal{N}\} && &&\\
\text{s.t.} \quad & v_{A_i,i}^{(x)} - v_i^{(x)} + z_i\left(S_i^{(x)}\right)^* + S_i^{(x)} z_i^* - \ell_i^{(x)}|z_i|^2 = 0 && i \in \mathcal{E} && \text{(14b)}\\
& \sum_{i\in C_i}\left(S_{j,i}^{(x)} - z_j\ell_{j,i}^{(x)}\right) - S_i^{(x)} + s_i^{(x)} = 0 && i \in \mathcal{N} && \text{(14c)}\\
& |S_i^{(z)}|^2 \le v_i^{(z)}\ell_i^{(z)} && i \in \mathcal{E} && \text{(14d)}\\
& s_i^{(z)} \in \mathcal{I}_i && i \in \mathcal{N} && \text{(14e)}\\
& \underline{v}_i \le v_i^{(z)} \le \overline{v}_i && i \in \mathcal{N} && \text{(14f)}\\
& S_{i,A_i}^{(x)} = S_i^{(z)},\ \ell_{i,A_i}^{(x)} = \ell_i^{(z)},\ v_{A_i,i}^{(x)} = v_{A_i}^{(z)} && i \in \mathcal{N} && \text{(14g)}\\
& S_i^{(x)} = S_i^{(z)}, \ell_i^{(x)} = \ell_i^{(z)}, v_i^{(x)} = v_i^{(z)},\ s_i^{(x)} = s_i^{(z)} && i \in \mathcal{N} && \text{(14h)}
\end{aligned}
$$

where (14g) and (14h) are consensus constraints that force all the copies of each variable to be the same. Since ADMM has two separate groups of variables x and z that is updated alternatively, we put superscripts $(\cdot)^{(x)}$ and $(\cdot)^{(z)}$ on each variable to denote whether the variable is updated in the x-update or z-update step.

Next, we apply ADMM to decompose (14) by relaxing the consensus constraints in (14g) and (14h). Let λ, μ, γ be the Lagrangian multipliers associated with (14g) and (14h) as specified in Table I.

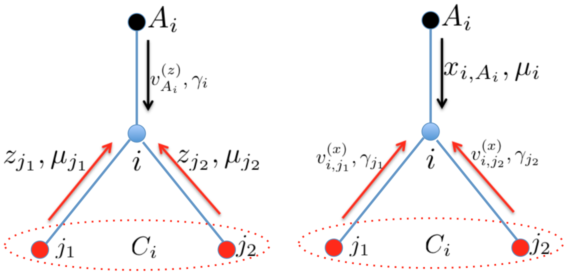
Denote

$$
\begin{aligned}
x_i &:= \left(v_i^{(x)}, \ell_i^{(x)}, S_i^{(x)}, s_i^{(x)}\right)\\
x_{j,i} &:= \left(\ell_{j,i}^{(x)}, S_{j,i}^{(x)}\right)\\
z_i &:= \left(v_i^{(z)}, \ell_i^{(z)}, S_i^{(z)}, s_i^{(z)}\right)\\
\lambda_i &:= (\lambda_{k,i}, k = 1,2,3,4)\\
\mu_i &:= (\mu_{k,i}, k = 1,2)
\end{aligned}
$$

The variables maintained by each agent (bus) i are its local variables for itself: x_i, z_i, the copy of its parent's voltage $v_{A_i,i}^{(x)}$, the copy $x_{j,i}$ from each of its child $j \in C_i$ and the associated Lagrangian multipliers. Let $\mathcal{A}_i$ denote the set of variables, then

$$\mathcal{A}_i := \{x_i, v_{A_i,i}^{(x)}, \{x_{j,i}, \mu_j, j \in C_i\}, z_i, \lambda_i, \gamma_i\}$$

Next, we demonstrate that the problem in (14) can be solved in a distributed manner using ADMM, i.e. both the x-update (11a) and z-update (11b) can be decomposed into small subproblems that can be solved simultaneously by each agent i. For ease of presentation, we remove the iteration number k in (11) for all the variables, which will be updated accordingly after each subproblem is solved. The augmented Lagrangian for modified ROPF problem is given in (15). Note that in (15), x_{i,A_i} and z_i consist of different components and $x_{i,A_i} - z_i$ is composed of the components that appear in both x_{i,A_i} and z_i, *i.e.* $x_{i,A_i} - z_i := \left(S_{i,A_i}^{(x)} - S_i^{(z)}, \ell_{i,A_i}^{(x)} - \ell_i^{(z)}\right)$.



(a) x-update (b) z-update

Fig. 2: Message exchange in the x and z-update step for agent i.

$$
\begin{aligned}
&L_\rho(x, z, \lambda, \gamma, \mu) && (15a)\\
&= \sum_{i\in\mathcal{N}} \left(f_i(p_i^{(z)}) + \langle \lambda_i, x_i - z_i\rangle + \langle \mu_i, x_{i,A_i} - z_i\rangle + \sum_{j\in C_i} \langle \gamma_j, v^{(x)}_{i,j} - v^{(z)}_i\rangle + \frac{\rho}{2}\left(\|x_i - z_i\|_s^2 + \|x_{i,A_i} - z_i\|_n^2 + \sum_{j\in C_i} \|v^{(x)}_{i,j} - v^{(z)}_i\|_2^2 \right)\right) && (15b)\\
&= \sum_{i\in\mathcal{N}} \left(f_i(p_i^{(z)}) + \langle \lambda_i, x_i - z_i\rangle + \sum_{j\in C_i} \langle \mu_j, x_{j,i} - z_j\rangle + \langle \gamma_i, v^{(x)}_{A_i,i} - v^{(z)}_{A_i}\rangle + \frac{\rho}{2}\left(\|x_i - z_i\|_s^2 + \sum_{j\in C_i} \|x_{j,i} - z_j\|_n^2 + \|v^{(x)}_{A_i,i} - v^{(z)}_{A_i}\|_2^2 \right)\right) && (15c)
\end{aligned}
$$

In the x-update, each agent i jointly solves the following x-update (11a).

$$
\arg\min_{x\in\mathcal{K}_x} L_\rho(x, z, \lambda, \gamma, \mu) = \arg\min_{x\in\mathcal{K}_x} \sum_{i\in\mathcal{N}} G_i(x_i, v^{(x)}_{A_i,i}, \{x_{j,i}, j \in C_i\}), \tag{16}
$$

where $G_i(x_i, v^{(x)}_{A_i,i}, \{x_{j,i}, j \in C_i\})$ is obtained from (15b) and

$$
\begin{aligned}
&G_i(x_i, v^{(x)}_{A_i,i}, \{x_{j,i}, j \in C_i\})\\
:=&\langle \lambda_i, x_i\rangle + \sum_{j\in C_i} \langle \mu_j, x_{j,i}\rangle + \langle \gamma_i, v^{(x)}_{A_i,i}\rangle + \frac{\rho}{2}\left(\|x_i - z_i\|_2^2 + \sum_{j\in C_i} \|x_{j,i} - z_j\|_2^2 + \|v^{(x)}_{A_i,i} - v^{(z)}_{A_i}\|_2^2 \right)
\end{aligned}
$$

The corresponding subproblem for each agent i that jointly solves (16) is

$$
\begin{aligned}
&\min\ G_i(x_i, v^{(x)}_{A_i,i}, \{x_{j,i}, j \in C_i\})\\
&\text{over } x_i, v^{(x)}_{A_i,i}, \{x_{j,i}, j \in C_i\}\\
&\text{s.t. } v^{(x)}_{A_i,i} - v^{(x)}_i + z_i \left(S^{(x)}_i\right)^* + S^{(x)}_i z_i^* - \ell^{(x)}_i |z_i|^2 = 0\\
&\qquad \sum_{i\in C_i} \left(S^{(x)}_{j,i} - z_j \ell^{(x)}_{j,i} \right) - S^{(x)}_i + s^{(x)}_i = 0
\end{aligned} \tag{17}
$$

Prior to solving (17), each agent i needs to collect $\left(v^{(z)}_{A_i}, \gamma_i\right)$ from its parent and (z_j, μ_j) from all of its children $j \in C_i$. The message exchanges in the x-update is illustrated in Fig. 2a.

Next, we show how to solve (17) in closed form. For each i, we can stack the real and imaginary part of the variables $(x_i, v^{(x)}_{A_i,i}, \{x_{j,i}, j \in C_i\})$ in a vector with appropriate dimensions and denote it as $\tilde{x}$. Then the subproblem solved by agent i in the x-update (17) takes the following form:

$$
\min_{\tilde{x}} \quad \frac{1}{2}\tilde{x}^H A \tilde{x} + c^H \tilde{x} \quad \text{s.t. } B\tilde{x} = 0 \tag{18}
$$

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

$$
\tilde{x} = \left(A^{-1}B^H (BA^{-1}B^H)^{-1} BA^{-1} - A^{-1} \right) c
$$

In the z-update, each agent i updates z_i by jointly solving the z-update (11b)

$$
\arg\min_{z\in\mathcal{K}_z} L_\rho(x, z, \lambda, \gamma, \mu) = \arg\min_{z\in\mathcal{K}_z} \sum_{i\in\mathcal{N}} H_i(z_i), \tag{19}
$$

where $H_i(z_i)$ is obtained from (15c) and

$$H_i(z_i) := f_i(p_i^{(z)}) - \langle \lambda_i, z_i \rangle - \langle \mu_i, z_i \rangle - \sum_{j \in C_i} \langle \gamma_j, v_i^{(z)} \rangle + \frac{\rho}{2} \left(\|x_i - z_i\|_2^2 + \|x_{i,A_i} - z_i\|_2^2 + \sum_{j \in C_i} \|v_{i,j}^{(x)} - v_i^{(z)}\|_2^2 \right)$$

The corresponding subproblem for each agent i that jointly solves (19) is

$$\begin{aligned} \text{min} \quad & H_i(z_i) \\ \text{over} \quad & z_i \\ \text{s.t.} \quad & |S_i^{(z)}|^2 \le v_i^{(z)} \ell_i^{(z)} \\ & s_i^{(z)} \in \mathcal{I}_i \\ & \underline{v}_i \le v_i^{(z)} \le \overline{v}_i \end{aligned} \tag{20}$$

Prior to solving (20), each agent i needs to collect (x_{i,A_i}, μ_i) from its parent and $(v_{i,j}^{(x)}, \gamma_j)$ from all of its children $j \in C_i$. The message exchanges in the z-update is illustrated in Fig. 2b.

Next, we show how to solve (20) in closed form. Note that

$$\begin{aligned} H_i(z_i) :=& f_i(p_i^{(z)}) - \langle \lambda_i, z_i \rangle - \langle \mu_i, z_i \rangle - \sum_{j \in C_i} \langle \gamma_j, v_i^{(z)} \rangle + \frac{\rho}{2} \left(\|x_i - z_i\|_2^2 + \|x_{i,A_i} - z_i\|_2^2 + \sum_{j \in C_i} \|v_{i,j}^{(x)} - v_i^{(z)}\|_2^2 \right) \\ =& \rho \left(|S_i^{(z)} - \hat{S}_i|^2 + |\ell_i^{(z)} - \hat{\ell}_i|^2 + \frac{|C_i| + 1}{2} |v_i^{(z)} - \hat{v}_i|^2 \right) + f_i(p_i^{(z)}) + \frac{\rho}{2} \|s_i^{(z)} - \hat{s}_i\|_2^2 + \text{constant} \end{aligned} \tag{21}$$

We use square completion to obtain (21) and the variables labeled with hat are some constants. Then (20) can be furthered decomposed into two subproblems as below. The first one is

$$\begin{aligned} \text{min} \quad & \left| S_i^{(z)} - \hat{S}_i \right|^2 + \left| \ell_i^{(z)} - \hat{\ell}_i \right|^2 + \frac{|C_i| + 1}{2} \left| v_i^{(z)} - \hat{v}_i \right|^2 \\ \text{over} \quad & v_i^{(z)}, \ell_i^{(z)}, S_i^{(z)} \\ \text{s.t.} \quad & \left| S_i^{(z)} \right|^2 \le v_i^{(z)} \ell_i^{(z)} \\ & \underline{v}_i \le v_i^{(z)} \le \overline{v}_i \end{aligned} \tag{22}$$

The optimization problem in (22) has a quadratic objective, a second order cone constraint and a box constraint. We illustrate in Appendix A the procedure that solves (22). Compared with using generic iterative solver, the procedure is computationally efficient since it only requires to solve the zero of three polynomials with degree less than or equal to 4, which have closed form expression.

The second problem is

$$\begin{aligned} \text{min} \quad & f_i\left(p_i^{(z)}\right) + \frac{\rho}{2} \left\| s_i^{(z)} - \hat{s}_i \right\|_2^2 \\ \text{over} \quad & s_i^{(z)} \\ \text{s.t.} \quad & s_i^{(z)} \in \mathcal{I}_i \end{aligned} \tag{23}$$

Recall that $f_i\left(p_i^{(z)}\right) := \frac{\alpha_i}{2} p_i^2 + \beta_i p_i$ as in (2). If $\mathcal{I}_i$ takes the form of (4a), the closed form solution to (23) is

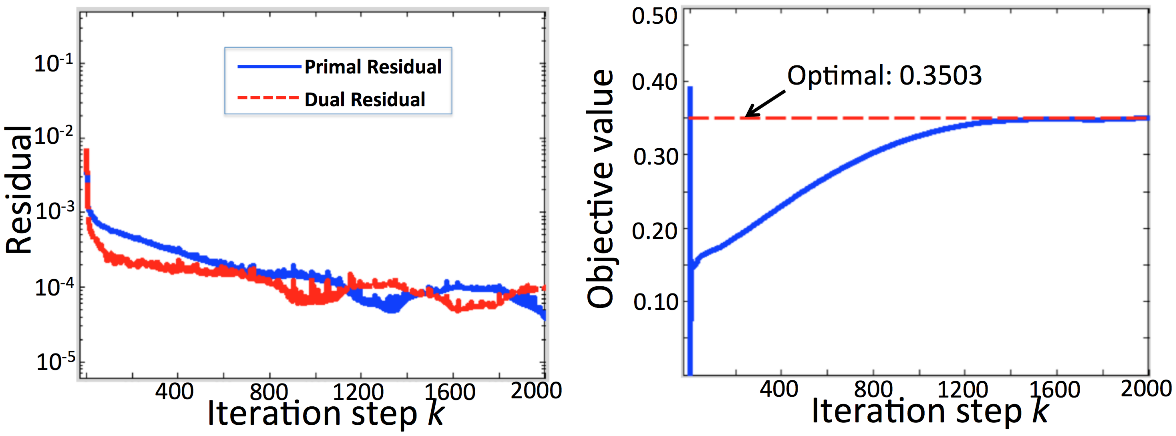
$$p_i^{(z)} = \left[\frac{\rho \hat{p}_i - \beta}{\alpha + \rho} \right]_{\underline{p}_i}^{\overline{p}_i} \qquad q_i^{(z)} = [\hat{q}_i]_{\underline{q}_i}^{\overline{q}_i}$$

where $[x]_a^b := \min\{a, \max\{x, b\}\}$. If $\mathcal{I}_i$ takes the form of (4b), there also exists a closed form expression to (23) and the procedure is relegate to Appendix B.

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 z variable. The voltage magnitude square $v_i^{(z)} = 1$. The power injection $s_i^{(z)}$ is picked up from a feasible point in the feasible region $\mathcal{I}_i$. The branch power $S_i^{(z)}$ is the aggregate power injection $s_i^{(z)}$ from the nodes connected by line i (Note that the network has a tree topology.). The branch current $\ell_i^{(z)} = \frac{|S_i^{(z)}|^2}{v_i^{(z)}}$ according to (1c). The x variables are initialized using the corresponding z variable according to (14g)-(14h). Intuitively, the above initialization procedure can be interpreted as a solution to the branch flow equation (1) assuming zero impedance on all the lines.

TABLE II: Distributed algorithm of OPF

```
Distributed Algorithm of OPF
Input: network $\mathcal{T}$, power injection region $\mathcal{I}_i$, voltage region $(\underline{v}_i, \overline{v}_i)$,
        line impedance $z_i$ for $i \in \mathcal{N}$.
Output: voltage $v$, power injection $s$.
1. Initialize the $x$ and $z$ variables.
2. Iterate the following step until both the primal residual $s^k$ (12a)
and the dual residual $r^k$ (12b) are below $10^{-4}\sqrt{|\mathcal{N}|}$.
    a. In the $x$-update, each agent $i$ solves (17) to update $x$.
    b. In the $z$-update, each agent $i$ solves (20) to update $z$.
    c. In the multiplier update, update $\lambda, \mu, \gamma$ by (11c).
```



(a) Primal and dual residual

(b) Objective value

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

For the stopping criteria, there is no general rule for ADMM based algorithm and it usually hinges on the problem [17]. In [17], it is suggested that a reasonable stopping criteria is that both the primal residual r^k defined in (12a) and the dual residual s^k defined in (12b) 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 Table II.

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 between off-the-shelf solver (CVX [21]) and our algorithm. Second, we simulate the proposed algorithm on networks of different sizes to understand the factors that affect the 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

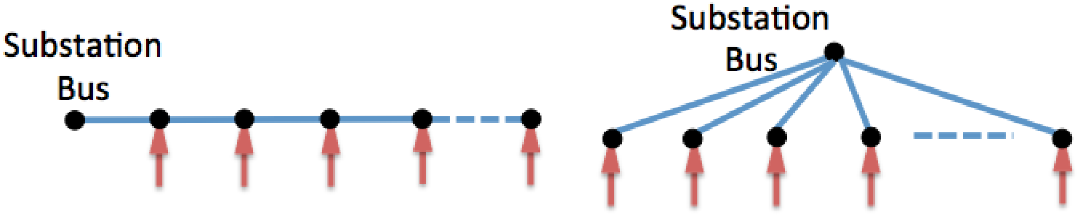
In the 2,065 bus distribution network, there are 1,409 household loads whose power consumptions are within 0.07kw–7.6kw and 142 commercial loads, whose power consumptions are within 5kw–36.5kw. There are 135 rooftop PV panels, whose nameplates are within 0.7–4.5kw, distributed across the 1,409 houses.

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. $\overline{v}_i = 1.05^2$ and $\underline{v}_i = 0.95^2$ for $i \in \mathcal{N}_+$. The control devices are the rooftop PV panels whose real and reactive power injections are controlled. The objective is to minimize power loss across the network, namely $\alpha_i = 0, \beta_i = 1$ for $i \in \mathcal{N}$, where α_i, β_i are coefficients in the objective function and defined 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.

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

TABLE III: 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



(a) Line network

(b) Fat tree network

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

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.

Finally, we show the advantage of closed form solution by comparing the computation time of solving the subproblems by an off-the-shelf solver (CVX) and by our algorithm. In particular, we compare the average computation time of solving the subproblem in both the x-update and the z-update step. In the x-update step, the average time required to solve the subproblem is 1.7×10^{-4}s for the proposed algorithm but 0.2s for CVX. In the z-update step, the average time required to solve the subproblem is 5.1×10^{-4}s for the proposed algorithm but 0.3s for CVX. 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. The time of convergence(ToC) is determined by both the computation time required within each iteration and the number of iterations. In this subsection, we study the number of iterations, namely rate of convergence.

To our best 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 [25], [26] 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 Table II, 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 III. For simplicity, we assume the number of iterations T to converge takes the linear form $T = aN + bD$. Using the data in Table III, 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.

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 IV, 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 have developed a distributed algorithm for optimal power flow problem based on alternating direction method of multiplier for balanced radial distribution network. We have derived a closed 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 and the optimization subproblem in each ADMM iteration is solved 1,000x faster than generic optimization solver.

TABLE IV: 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

References

[1] Q. Peng and S. Low, “Distributed algorithm for optimal power flow on a radial network,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014.

[2] 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.

[3] R. Jabr, “Radial Distribution Load Flow Using Conic Programming,” *IEEE Trans. on Power Systems*, 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 *Proceedings of IEEE SmartGridComm Conference*, October 2011.

[5] 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.

[6] 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.

[7] ——, “Convex relaxation of optimal power flow, II: exactness,” *IEEE Trans. on Control of Network Systems*, vol. 1, no. 2, pp. 177–189, June 2014.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[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 *52nd IEEE Conference on Decision and Control*, 2013.

[14] 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.

[15] E. Dall’Anese, H. Zhu, and G. B. Giannakis, “Distributed optimal power flow for smart microgrids,” *arXiv preprint arXiv:1211.5856*, 2012.

[16] 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.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[18] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, “Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems,” *IEEE Trans. on Automatic Control*, vol. 60, no. 3, pp. 644–658, 2013.

[19] D. L. Sun and C. Fevotte, “Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6201–6205.

[20] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, “Fast image recovery using variable splitting and constrained optimization,” *Image Processing, IEEE Transactions on*, vol. 19, no. 9, pp. 2345–2356, 2010.

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

[22] 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.

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

[24] L. Gan, N. Li, U. Topcu, and S. H. Low, “Exact convex relaxation of optimal power flow in radial networks,” *IEEE Trans. on Automatic Control*, 2014.

[25] 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.

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

Appendix A

Denote $z_1 := \mathbf{Re}(S_i^{(z)})$, $z_2 := \mathbf{Im}(S_i^{(z)})$, $z_3 := \sqrt{\frac{|C_i|+1}{2} v_i^{(z)}}$ and $z_4 := \ell_i^{(z)}$. Then the optimization problem (22) can be written equivalently as

$$\begin{aligned} \min \quad & \sum_{i=1}^{4}(z_i^2 + c_i z_i) && \text{(24a)} \\ \text{over} \quad & z \\ \text{s.t.} \quad & \frac{z_1^2 + z_2^2}{z_3} \le k^2 z_4 && \text{(24b)} \\ & \underline{z}_3 \le z_3 \le \overline{z}_3 && \text{(24c)} \end{aligned}$$

where $\overline{z}_3 > \underline{z}_3 > 0$ and c_i, k are constants that hinges on the constants in (22).

Below we will derive a procedure that solves (24). Let $\mu \geq 0$ denote the Lagrangian multiplier for constraint (24b) and $\underline{\lambda}, \overline{\lambda} \geq 0$ denote the Lagrangian multipliers for constraint (24c), then the Lagrangian of P1 is

$$L(z, \mu, \lambda) = \sum_{i=1}^{4}(z_i^2 + c_i z_i) + \mu\left(\frac{z_1^2 + z_2^2}{z_3} - k^2 z_4\right) + \overline{\lambda}(z_3 - \overline{z}_3) - \underline{\lambda}(z_3 - \underline{z}_3)$$

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

$$2z_1 + c_1 + 2\mu\frac{z_1}{z_3} = 0 \tag{25a}$$

$$2z_2 + c_2 + 2\mu\frac{z_2}{z_3} = 0 \tag{25b}$$

$$2z_3 + c_3 - \mu\frac{z_1^2 + z_2^2}{z_3^2} + \overline{\lambda} - \underline{\lambda} = 0 \tag{25c}$$

$$2z_4 + c_4 - k^2\mu = 0 \tag{25d}$$

$$\overline{\lambda}(z_3 - \overline{z}_3) = 0 \quad \overline{\lambda} \geq 0 \quad z_3 \leq \overline{z}_3 \tag{25e}$$

$$\underline{\lambda}(\underline{z}_3 - z_3) = 0 \quad \underline{\lambda} \geq 0 \quad z_3 \geq \underline{z}_3 \tag{25f}$$

$$\mu\left(\frac{z_1^2 + z_2^2}{z_3} - k^2 z_4\right) = 0 \quad \mu \geq 0 \quad \frac{z_1^2 + z_2^2}{z_3} \leq k^2 z_4 \tag{25g}$$

Lemma A.1: There exists a unique solution $(z^*, \mu^*, \underline{\lambda}^*, \overline{\lambda}^*)$ to (25) if $\overline{z}_3 > \underline{z}_3 \geq 0$.

Proof: P1 is feasible since $z = (0, 0, \underline{z}_3, 1)$ satisfies (24b)-(24c). In addition, P1 is a strictly convex optimization problem since the objective (24a) is a strictly convex function of z and the constraints (24b) and (24c) are also convex. Hence, there exists a unique solution z^* to P1, which indicates there exists a unique solution $(z^*, \mu^*, \underline{\lambda}^*, \overline{\lambda}^*)$ to the KKT optimality conditions (25). ■

Lemma A.1 says that there exists a unique solution to (25), which is also the optima to P1. In the following, we will solve (25) through enumerating value of the multipliers $\mu, \overline{\lambda}, \underline{\lambda}$. Specifically, we first assume $\mu^* = 0$ (Case 1 below), which is equivalent to assume constraint (24b) is inactive. If there is a feasible solution to (25), it is the unique solution to (25). Otherwise, we assume $\mu^* = 0$ (Case 2 below), which is equivalent to assume that the equality is obtained at optimality in (24b).

Case 1: If $\mu = 0$, (25) becomes

$$2z_1 + c_1 = 0 \tag{26a}$$

$$2z_2 + c_2 = 0 \tag{26b}$$

$$2z_3 + c_3 + \overline{\lambda} - \underline{\lambda} = 0 \tag{26c}$$

$$2z_4 + c_4 = 0 \tag{26d}$$

$$\overline{\lambda}(z_3 - \overline{z}_3) = 0 \quad \overline{\lambda} \geq 0 \quad z_3 \leq \overline{z}_3 \tag{26e}$$

$$\underline{\lambda}(z_3 - \underline{z}_3) = 0 \quad \underline{\lambda} \geq 0 \quad z_3 \geq \underline{z}_3 \tag{26f}$$

$$\frac{z_1^2 + z_2^2}{z_3} \leq k^2 z_4 \tag{26g}$$

The solution to (26a)–(26f) ignoring (26g) is

$$z_1 = -\frac{c_1}{2}, \; z_2 = -\frac{c_2}{2}, \; z_3 = \left[-\frac{c_3}{2}\right]_{\underline{z}_3}^{\overline{z}_3}, \; z_4 = -\frac{c_4}{2}$$

$$\overline{\lambda} = -(2z_3 + c_3)1_{\{z_3 = \overline{z}_3\}}, \underline{\lambda} = -(2z_3 + c_3)1_{\{z_3 = \underline{z}_3\}}$$

and if the solution satisfies (26g), it is the solution to (25). Otherwise, we go to Case 2.

Case 2: If $\mu > 0$, (25) becomes

$$2z_1 + c_1 + 2\mu\frac{z_1}{z_3} = 0 \tag{27a}$$

$$2z_2 + c_2 + 2\mu\frac{z_2}{z_3} = 0 \tag{27b}$$

$$2z_3 + c_3 - \mu\frac{z_1^2 + z_2^2}{z_3^2} + \overline{\lambda} - \underline{\lambda} = 0 \tag{27c}$$

$$\overline{\lambda}(z_3 - \overline{z}_3) = 0 \quad \overline{\lambda} \geq 0 \quad z_3 \leq \overline{z}_3 \tag{27d}$$

$$\underline{\lambda}(z_3 - \underline{z}_3) = 0 \quad \underline{\lambda} \geq 0 \quad z_3 \geq \underline{z}_3 \tag{27e}$$

$$\mu = \frac{1}{k^2}(2z_4 + c_4) \tag{27f}$$

$$z_4 = \frac{z_1^2 + z_2^2}{k^2 z_3} \tag{27g}$$

Substitute (27g) into (27f), we obtain

$$\mu = \frac{1}{k^2}(2z_4 + c_4) = \frac{2(z_1^2 + z_2^2)}{k^4 z_3} + \frac{c_4}{k^2} \tag{28}$$

Then substituting (27f) into (27a)-(27e), we can write (27) equivalently as

$$2 + \frac{c_1}{z_1} + 4\frac{(z_1^2 + z_2^2)}{k^4 z_3^2} + \frac{2c_4}{k^2 z_3} = 0 \tag{29a}$$

$$2 + \frac{c_2}{z_2} + 4\frac{(z_1^2 + z_2^2)}{k^4 z_3^2} + \frac{2c_4}{k^2 z_3} = 0 \tag{29b}$$

$$2 + \frac{c_3}{z_3} - 2\frac{(z_1^2 + z_2^2)^2}{k^4 z_3^4} - c_4\frac{z_1^2 + z_2^2}{z_3^3} + \frac{\overline{\lambda} - \underline{\lambda}}{z_3} = 0 \tag{29c}$$

$$\overline{\lambda}(z_3 - \overline{z}_3) = 0 \quad \overline{\lambda} \geq 0 \quad z_3 \leq \overline{z}_3 \tag{29d}$$

$$\underline{\lambda}(z_3 - \underline{z}_3) = 0 \quad \underline{\lambda} \geq 0 \quad z_3 \geq \underline{z}_3 \tag{29e}$$

where (29a)–(29c) are obtained through dividing both sides of (27a)–(27c) by z_1, z_2 and z_3, respectively. The variables μ, z_4 can be recovered via (27f) and (27g) after we solve (29).
By (29a) and (29b),

$$\frac{c_1}{z_1} = \frac{c_2}{z_2}$$

Denote $p := \frac{z_1}{c_1 z_3} = \frac{z_2}{c_2 z_3}$. Then (29) is equivalent to the following equations.

$$p = \frac{z_1}{c_1 z_3} = \frac{z_2}{c_2 z_3} \tag{30a}$$

$$2 + \frac{1}{p z_3} = -\left(\frac{4(c_1^2 + c_2^2)}{k^4}p^2 + 2\frac{c_4}{k^2 z_3}\right) \tag{30b}$$

$$2 + \frac{c_3}{z_3} = \frac{2(c_1^2 + c_2^2)^2}{k^4}p^4 + \frac{c_4(c_1^2 + c_2^2)}{k^2}\frac{p^2}{z_3} + \frac{\underline{\lambda} - \overline{\lambda}}{z_3} \tag{30c}$$

$$\overline{\lambda}(z_3 - \overline{z}_3) = 0 \quad \overline{\lambda} \geq 0 \quad z_3 \leq \overline{z}_3 \tag{30d}$$

$$\underline{\lambda}(z_3 - \underline{z}_3) = 0 \quad \underline{\lambda} \geq 0 \quad z_3 \geq \underline{z}_3 \tag{30e}$$

where (30b) is obtained by substitute $z_2 = c_2 p z_3$ into (29b), (30c) is obtained by substitute $z_1 = c_1 p z_3$ and $z_2 = c_2 p z_3$ into (29c). To solve (30), we further divide our analysis into two sub-cases depending on whether z_3^* hits the lower or upper bound.

- **Case 2.1**: $z_3^* = \overline{z}_3$. $(\underline{\lambda} = 0, \overline{\lambda} > 0)$ (The case of $z_3^* = \underline{z}_3$ can be solved using similar procedure.)
  We first substitute $z_3 = \overline{z}_3$ into (30b) and have

  $$\frac{4(c_1^2 + c_2^2)}{k^4}p^3 + \left(2\frac{c_4}{k^2\overline{z}_3} + 2\right)p + \frac{1}{\overline{z}_3} = 0, \tag{31}$$

  whose solution[2] is denoted by p^*. Then substitute p^* and $\overline{z}_3$ into (30a), we can recover z_1^* and z_2^*. Then we can obtain $\mu^*, \overline{\lambda}^*$ using (28) and (29c) by substituting $z_1^*, \ldots, z_4^*$. If $\mu^*, \overline{\lambda}^* \geq 0$, they collectively solve (25). Otherwise, we go to Case 2.2.

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

- **Case 2.2**: $\underline{z}_3 < z_3^* < \overline{z}_3$ $(\underline{\lambda}, \overline{\lambda} = 0)$.
  Since $\overline{\lambda}$ and $\underline{\lambda} = 0$, (30) reduces to

$$p = \frac{z_1}{c_1 z_3} = \frac{z_2}{c_2 z_3} \tag{32a}$$

$$2 + \frac{1}{p z_3} = -\left(\frac{4(c_1^2 + c_2^2)}{k^4} p^2 + 2\frac{c_4}{k^2 z_3}\right) \tag{32b}$$

$$2 + \frac{c_3}{z_3} = \frac{2(c_1^2 + c_2^2)^2}{k^4} p^4 + \frac{c_4(c_1^2 + c_2^2)}{k^2} \frac{p^2}{z_3} \tag{32c}$$

Dividing each side of (32b) by (32c) gives

$$\frac{2z_3 + \frac{1}{p}}{2z_3 + c_3} = -\frac{2}{(c_1^2 + c_2^2)p^2},$$

which implies

$$z_3 = -\frac{(c_1^2 + c_2^2)p + 2c_3}{2((c_1^2 + c_2^2)p^2 + 2)} \tag{33}$$

Then substitute (33) into (32b), we have

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

which is equivalent to

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

whose solution is denoted by p^*. Substitute p^* into (33), we can recover z_3^*, then z_1^*, z_2^* can be recovered via (32a). μ^* is recovered using (28). If $\mu^* \geq 0$, the corresponding solution solves (25).

Appendix B

If $\mathcal{I}_i$ takes the form of (4b), the optimization problem (23) takes the following form

$$\min_{p,q} \quad \frac{a_1}{2}p^2 + b_1 p + \frac{a_2}{2}q^2 + b_2 q \tag{34a}$$

$$\text{s.t.} \quad p^2 + q^2 \leq c^2 \tag{34b}$$

$$p \geq 0 \tag{34c}$$

where $a_1, a_2, c > 0$, b_1, b_2 are constants. The solutions to (34) 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:

$$b_1^2(a_2 + 2\lambda)^2 + b_2^2(a_1 + 2\lambda)^2 = (a_1 + 2\lambda)^2(a_2 + 2\lambda)^2 \tag{35}$$

which is a polynomial with degree of 4 and has closed form expression. There are four solutions to (35), but there is only one strictly positive λ^*, which can be proved via the KKT conditions of (34). 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 (34) is derived from standard applications of the KKT conditions of (34). For brevity, we skip the proof here.