# A Distributed Approach for the Optimal Power-Flow Problem Based on ADMM and Sequential Convex Approximations

Sindri Magnússon, Pradeep Chathuranga Weeraddana, *Member, IEEE*, and Carlo Fischione, *Member, IEEE*

***Abstract*—The optimal power-flow (OPF) problem, which plays a central role in operating electrical networks, is considered. The problem is nonconvex and is, in fact, NP hard. Therefore, designing efficient algorithms of practical relevance is crucial, though their global optimality is not guaranteed. Existing semidefinite programming relaxation-based approaches are restricted to OPF problems where zero duality holds. In this paper, an efficient novel method to address the *general nonconvex* OPF problem is investigated. The proposed method is based on an alternating direction method of multipliers combined with sequential convex approximations. The global OPF problem is decomposed into smaller problems associated with each bus of the network, the solutions of which are coordinated via a light communication protocol. Therefore, the proposed method is highly scalable. The convergence properties of the proposed algorithm are mathematically substantiated. Finally, the proposed algorithm is evaluated on a number of test examples, where the convergence properties of the proposed algorithm are numerically substantiated, and the performance is compared with a global optimal method.**

***Index Terms*—Distributed optimization, optimal power flow (OPF), smart grid.**

## I. Introduction

THE optimal power-flow (OPF) problem in electrical networks optimally determines the amount of power to be generated at each generator. Moreover, it decides how to dispatch the power such that a global network-wide objective criterion is optimized, while ensuring that the power demand of each consumer is met and that the related laws of physics are held. Traditionally, the OPF problem has only been solved in transmission networks. However, the extensive information gathering of individual power consumption in the smart grid has made the problem relevant, not only in transmission networks, but also in distribution networks which deliver electricity to end users.

Manuscript received January 18, 2014; revised August 14, 2014; accepted December 19, 2014. Date of publication April 20, 2015; date of current version September 14, 2015. This work was supported in part by the Wireless@KTH project DevelopLTE4Grids and in part by EIT project LTE4SE. Recommended by Associate Editor M. Chertkov.

The authors are with the Electrical Engineering School, Access Linnaeus Center, KTH Royal Institute of Technology, Stockholm 10 100 44, Sweden (e-mail: sindrim@kth.se; chatw@kth.se; carlofi@kth.se).

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/TCNS.2015.2399192

### A. Previous Work

The problem was originally presented by Carpentier [1], and has been extensively studied since then and become of great importance in efficient operation of power systems [2]. The OPF problem is nonconvex due to a quadratic relationship between the powers and the voltages and because of a lower bound on the voltage magnitudes. In fact, the problem is NP-hard, see [3]. Therefore, practical and general-purpose algorithms must rely on some approximations or heuristics. We refer the reader to [2] and [4] for a contemporary survey of OPF.

It is well known that the OPF problem is equivalently reformulated as a rank-constrained problem [5]. As a result, classic convex approximation techniques are applied to handle nonconvexities of the rank constraint, which usually results in a semidefinite program [6] (SDP). SDP relaxations to OPF have gained a lot of attention recently, see [3], [7]–[9], and references therein. The authors in [3] show that SDP relaxation is equivalent to the dual problem of the original OPF. Moreover, sufficient conditions for zero duality and mechanisms to recover the primal solution by the dual problem are given. Thus, [3] identifies a class of OPF problems, where the global optimum is obtained efficiently by using convex tools. Some other classes of OPF problems, for which zero duality holds, are investigated in [7]–[9]. In particular, [7] derives zero duality results for networks that have tree topology where oversatisfaction of loads is allowed. On the other hand, [8] and [9] provide graphically intuitive conditions for the zero duality gap for two-bus networks, which are later generalized to tree topologies.

The aforementioned references suggest that the applicability of SDP approaches is limited to *special network classes*. Of course, SDP relaxations can always be used to compute a lower bound on the optimal value of the primal problem. However, in practice, what is crucial is a network operating point. In general, SDP relaxations fail to provide a network operating point (i.e., a feasible point) due to nonzero duality gap [10]. Another drawback of SDP-based methods is that even when zero duality holds, if the objective functions are nonquadratic, the dual machinery employed in constructing primal feasible solutions is not applied. The authors in [10] and [11] explore limitations of SDP approaches and give practical examples where the sufficient conditions for zero duality do not hold.

Centralized methods for OPF problem, of course, exhibit poor scalability. On the contrary, distributed and scalable OPF solution methods are less investigated, though they are highly desirable in the context of rapidly growing real-world electrical

networks. Unlike centralized methods, distributed OPF solution methods are also appealing in the context of privacy and security, because they do not entail collecting possibly sensitive problem data at a central node. In other words, when solving in a centralized manner the OPF problem in the smart grid, the power companies must rely on private information, such as the load profile of their costumers [12], [13], which might be of interest to a third party. For example, government agencies might inquire the information to profile criminal activity, and insurance companies might be interested in buying the information to determine if an individual is viable for insurance [14]. Therefore, the gathering of private information at a centralized node has raised serious concerns about personal privacy which, in turn, discourages the use of centralized approaches. Interestingly, the sparsity of most electrical networks brings out an appealing decomposition structure and, therefore, it is worth investigating distributed methods for the OPF problem.

Distributed methods for the OPF problem are first studied in [15]–[17], where the transmission network is divided into regions and different decomposition methods, including the auxiliary problem principle, predictor-corrector proximal multiplier method, and alternating direction method, are explored to solve the problem distributively among these regions. The formulation is restricted to *2-region* network decompositions, and border variables cannot be shared among more than two regions. Another approach to decentralize the problem into regions is presented in [18]–[20]. The method is based on solving the Karush–Kuhn–Tucker (KKT) optimality conditions, where a Newton procedure is adapted. The authors provide a sufficient condition for convergent which can be interpreted as a measurement of coupling between regions. However, when the condition is not satisfied, they rely on the generalized minimal residual method to find the Newton direction, which involves a lot of communications between entities. The methods presented in [21] are limited to dc OPF.

More recent distributed algorithms are found in [22]–[28]. The decentralized methods in [22]–[24] capitalize on the SDP relaxation, which still has the drawbacks of being specific to special classes of networks and a lack of flexibility with general objective functions. Another relaxation method is presented in [25], where instead of the original nonconvex constraint sets, the convex hull of those are used. However, the method can result in an infeasible point to the original unrelaxed problem, entailing local methods to help construct good feasible points. Other recent works consider distributed methods for optimal reactive power flow in distribution networks [26], [27]. Both papers first make approximations that yield a convex OPF problem and then distribute the computation by using dual decomposition [26] and ADMM [27]. The recent work in [28] employs ADMM to the general nonconvex OPF problem to devise a scalable algorithm. A major drawback of the method in [28] is that its convergence is very sensitive to the initialization. In fact, the authors of [28] always initialize their algorithm with a point which is close to the optimal solution. However, the optimal solution is not known *a priori*, limiting the scope of the method.

Almost all of the aforementioned methods can be classified as those which are general yet not scalable and those which are scalable yet not general. However, methods, which are *simultaneously* general and scalable, are of crucial importance in theory as well as in practice and, therefore, deserve investigations.

### B. Our Contributions

The main contributions of this paper are as follows.

1) We develop a distributed algorithm for the *general nonconvex* OPF problem. Our approach is not restricted to any special classes of networks, where zero duality holds. It also handles nonquadratic convex objective functions, unlike the SDP-based distributed algorithms.
2) We capitalize on the alternating direction method of multipliers (ADMM) [29] to accomplish the distributed implementation (among electrical network buses) of the proposed algorithm with a *little* coordination of the neighboring entities. Thus, the proposed algorithm is highly *scalable*, which is favorable in practice.
3) In the case of subproblems, we capitalize on sequential approximations, in order to gracefully manipulate the nonconvexity issues. The approach is adopted from an existing algorithm originally proposed in [30] in the context of the centralized OPF problem.
4) The convergent properties of the proposed algorithm are mathematically and numerically substantiated.
5) A number of numerical examples are provided to evaluate the performance of the proposed algorithm.

### C. Organization and Notations

This paper is organized as follows. Section II describes the system model and problem formulation. The solution method is presented in Section III. In Section IV, we discuss some fundamental properties of the algorithm. Numerical results are provided in Section V. Finally, Section VI concludes this paper.

The imaginary unit is denoted by $j$, that is, $j = \sqrt{-1}$. Boldface lowercase and uppercase letters represent vectors and matrices, respectively, and calligraphic letters represent sets. The cardinality of $\mathcal{A}$ is denoted by $|\mathcal{A}|$. $\mathrm{len}(\mathbf{x})$ denotes the length of $\mathbf{x}$. The set of real and complex $n$-vectors are denoted by $\mathbb{R}^n$ and $\mathbb{C}^n$, respectively, and the set of real and complex $m \times n$ matrices are denoted by $\mathbb{R}^{m\times n}$ and $\mathbb{C}^{m\times n}$. We denote the real and imaginary parts of the complex number $z \in \mathbb{C}$ by $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$, respectively. The set of non-negative integers is denoted by $\mathbb{N}$, that is, $\mathbb{N} = \{0, 1, \ldots\}$. The superscript $(\cdot)^{\mathrm{T}}$ stands for transpose. We use parentheses to construct column vectors from comma separated lists, for example, $(\mathbf{a}, \mathbf{b}, \mathbf{c}) = [\mathbf{a}\mathrm{T} \quad \mathbf{b}\mathrm{T} \quad \mathbf{c}\mathrm{T}]\mathrm{T}$. We denote the diagonal block matrix with $\mathbf{A}_1, \cdots, \mathbf{A}_N$ on the diagonal by $\mathrm{diag}(\mathbf{A}_1, \cdots, \mathbf{A}_N)$. The Hadamard product of the matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \circ \mathbf{B}$. We denote by $\|\mathbf{x}\|_2$ the $\ell_2$-norm of the vector $\mathbf{x}$. We denote the gradient of the function $f$ in the point $\mathbf{x}$ by $\nabla_{\mathbf{x}} f$.

## II. System Model and Problem Formulation

Consider an electrical network with $N$ buses with $\mathcal{N} = \{1, 2, \ldots, N\}$ denoting the set of buses and $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ representing the set of flow lines. We let $i_k = i_k^{\mathrm{Re}} + j i_k^{\mathrm{Im}}$ be the current injection and $v_k = v_k^{\mathrm{Re}} + j v_k^{\mathrm{Im}}$ be the voltage at bus

$k \in \mathcal{N}$. Let $p_k^{\mathrm{D}} + jq_k^{\mathrm{D}} \in \mathbb{C}$ and $p_k^{\mathrm{G}} + jq_k^{\mathrm{G}} \in \mathbb{C}$ denote the complex power demand and the complex power generated by bus $k \in \mathcal{N}$, respectively. Thus, the complex power $p_k + jq_k \in \mathbb{C}$ injected to bus $k$ is given by $p_k + jq_k = (p_k^{\mathrm{G}} + jq_k^{\mathrm{G}}) - (p_k^{\mathrm{D}} + jq_k^{\mathrm{D}})$.

For notational compactness, we let $\mathbf{p}^{\mathrm{G}}$, $\mathbf{q}^{\mathrm{G}}$, $\mathbf{p}^{\mathrm{D}}$, $\mathbf{q}^{\mathrm{D}}$, $\mathbf{p}$, $\mathbf{q}$, $\mathbf{i}$, $\mathbf{i}^{\mathrm{Re}}$, $\mathbf{i}^{\mathrm{Im}}$, $\mathbf{v}$, $\mathbf{v}^{\mathrm{Re}}$, and $\mathbf{v}^{\mathrm{Im}}$ denote the vectors $(p_k^{\mathrm{G}})_{k\in\mathcal{N}}$, $(q_k^{\mathrm{G}})_{k\in\mathcal{N}}$, $(p_k^{\mathrm{D}})_{k\in\mathcal{N}}$, $(q_k^{\mathrm{D}})_{k\in\mathcal{N}}$, $(p_k)_{k\in\mathcal{N}}$, $(q_k)_{k\in\mathcal{N}}$, $(i_k)_{k\in\mathcal{N}}$, $(i_k^{\mathrm{Re}})_{k\in\mathcal{N}}$, $(i_k^{\mathrm{Im}})_{k\in\mathcal{N}}$, $(v_k)_{k\in\mathcal{N}}$, $(v_k^{\mathrm{Re}})_{k\in\mathcal{N}}$, and $(v_k^{\mathrm{Im}})_{k\in\mathcal{N}}$, respectively. We denote by $i_{ls}^{\mathrm{Re}} + ji_{ls}^{\mathrm{Im}} \in \mathbb{C}$ the complex current and by $p_{ls} + jq_{ls} \in \mathbb{C}$ the complex power transferred from bus $l$ to the rest of the network through the flow line $(l,s) \in \mathcal{L}$. The admittance matrix $\mathbf{Y} \in \mathbb{C}^{N\times N}$ of the network is given by

$$\mathbf{Y} = \begin{cases} y_{ll} + \sum_{(l,t)\in\mathcal{L}} y_{lt}, & \text{if } l = s, \\ -y_{ls}, & \text{if } (l,s) \in \mathcal{L}, \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $y_{ls} = g_{ls} + jb_{ls} \in \mathbb{C}$ is the admittance in the flow line $(l,s) \in \mathcal{L}$, and $y_{ll} = g_{ll} + jb_{ll} \in \mathbb{C}$ is the admittance to ground at bus $l$. We let $\mathbf{G} \in \mathbb{R}^{N\times N}$ and $\mathbf{B} \in \mathbb{R}^{N\times N}$ denote the real and imaginary parts of $\mathbf{Y}$, respectively. In particular, $[\mathbf{G}]_{ls} = g_{ls}$ and $[\mathbf{B}]_{ls} = b_{ls}$ yield $\mathbf{Y} = \mathbf{G} + j\mathbf{B}$.

### A. Centralized Formulation

For fixed power demands, $\mathbf{p}^{\mathrm{D}}$ and $\mathbf{q}^{\mathrm{D}}$, the goal of the OPF problem is to find the optimal way to tune the variables $\mathbf{p}^{\mathrm{G}}$, $\mathbf{q}^{\mathrm{G}}$, $\mathbf{p}$, $\mathbf{q}$, $\mathbf{i}^{\mathrm{Re}}$, $\mathbf{i}^{\mathrm{Im}}$, $\mathbf{v}^{\mathrm{Re}}$, and $\mathbf{v}^{\mathrm{Im}}$, ensuring that the relationships among the variables are held and system limitations are respected. The objective function differs between applications. In this paper, we consider the minimization of a convex cost function of real power generation. We denote by $f_k^{\mathrm{G}}$, the cost of generating power at bus $k \in \mathcal{G}$, where $\mathcal{G} \subseteq \mathcal{N}$ denotes the set of generator buses. The OPF problem can now be expressed as[1]

$$\min \sum_{k\in\mathcal{G}} f_k^{\mathrm{G}}\left(p_k^{\mathrm{G}}\right) \tag{2a}$$

$$\text{s.t. } \mathbf{i}^{\mathrm{Re}} + j\mathbf{i}^{\mathrm{Im}} = \mathbf{G}\mathbf{v}^{\mathrm{Re}} - \mathbf{B}\mathbf{v}^{\mathrm{Im}} + j(\mathbf{B}\mathbf{v}^{\mathrm{Re}} + \mathbf{G}\mathbf{v}^{\mathrm{Im}}) \tag{2b}$$

$$p_k + jq_k = p_k^{\mathrm{G}} - p_k^{\mathrm{D}} + j\left(q_k^{\mathrm{G}} - q_k^{\mathrm{D}}\right), k \in \mathcal{N}, \tag{2c}$$

$$i_{ls}^{\mathrm{Re}} + ji_{ls}^{\mathrm{Im}} = \left(\mathbf{c}_{ls}^{\mathrm{T}} + j\mathbf{d}_{ls}^{\mathrm{T}}\right)(v_l^{\mathrm{Re}}, v_s^{\mathrm{Re}}, v_l^{\mathrm{Im}}, v_s^{\mathrm{Im}}), (l,s) \in \mathcal{L} \tag{2d}$$

$$\mathbf{p} + j\mathbf{q} = \mathbf{v}^{\mathrm{Re}} \circ \mathbf{i}^{\mathrm{Re}} + \mathbf{v}^{\mathrm{Im}} \circ \mathbf{i}^{\mathrm{Im}} + j\left(\mathbf{v}^{\mathrm{Im}} \circ \mathbf{i}^{\mathrm{Re}} - \mathbf{v}^{\mathrm{Re}} \circ \mathbf{i}^{\mathrm{Im}}\right) \tag{2e}$$

$$p_{ls} + jq_{ls} = v_l^{\mathrm{Re}} i_{ls}^{\mathrm{Re}} + v_l^{\mathrm{Im}} i_{ls}^{\mathrm{Im}} + j(v_l^{\mathrm{Im}} i_{ls}^{\mathrm{Re}} - v_l^{\mathrm{Re}} i_{ls}^{\mathrm{Im}}) \quad (l,s) \in \mathcal{L} \tag{2f}$$

$$p_k^{\mathrm{G,min}} \le p_k^{\mathrm{G}} \le p_k^{\mathrm{G,max}}, k \in \mathcal{N}, \tag{2g}$$

$$q_k^{\mathrm{G,min}} \le q_k^{\mathrm{G}} \le q_k^{\mathrm{G,max}}, k \in \mathcal{N}, \tag{2h}$$

$$(i_{ls}^{\mathrm{Re}})^2 + (i_{ls}^{\mathrm{Im}})^2 \le (i_{ls}^{\max})^2, (l,s) \in \mathcal{L}, \tag{2i}$$

$$p_{ls}^2 + q_{ls}^2 \le (s_{ls}^{\max})^2, (l,s) \in \mathcal{L}, \tag{2j}$$

$$|p_{ls}| \le p_{ls}^{\max}, (l,s) \in \mathcal{L}, \tag{2k}$$

$$\left(v_k^{\min}\right)^2 \le \left(v_k^{\mathrm{Re}}\right)^2 + \left(v_k^{\mathrm{Im}}\right)^2 \le \left(v_k^{\max}\right)^2, k \in \mathcal{N} \tag{2l}$$

[1]Formulation (2) is equivalent to the OPF formulation in [3], and one can easily switch between the two formulation by using simple transformations. We use formulation (2), because it is convenient, in terms of notations, when describing the content in subsequent sections.

where the variables are $\mathbf{p}^{\mathrm{G}}$, $\mathbf{q}^{\mathrm{G}}$, $\mathbf{p}$, $\mathbf{q}$, $\mathbf{i}^{\mathrm{Re}}$, $\mathbf{i}^{\mathrm{Im}}$, $\mathbf{v}^{\mathrm{Re}}$, $\mathbf{v}^{\mathrm{Im}}$, and $i_{ls}^{\mathrm{Re}}$, $i_{ls}^{\mathrm{Im}}$, $p_{ls}$, $q_{ls}$ for $(l,s) \in \mathcal{L}$. Here, constraint (2b) is from $\mathbf{i} = \mathbf{Y}\mathbf{v}$, (2c) is derived from the conservation of power flow holds, (2e) is from $i_{ls}^{\mathrm{Re}} = \mathrm{Re}(y_{ls}(v_l - v_s))$, (2f) is from the complex power being $(\mathbf{v} \circ \mathbf{i}^*)$, and $i_{ls}^{\mathrm{Im}} = \mathrm{Im}(y_{ls}(v_l - v_s))$ with $\mathbf{c}_{ls} = (g_{ls}, -g_{ls}, b_{ls}, -b_{ls})$ and $\mathbf{d}_{ls} = (b_{ls}, -b_{ls}, -g_{ls}, g_{ls})$, and (2f) is from $p_{ls} = \mathrm{Re}(v_l i_{ls}^*)$ and $q_{ls} = \mathrm{Im}(v_l i_{ls}^*)$. Note that (2b)–(2f) correspond to the constraints imposed by the laws of physics associated with the electrical network. In addition, (2g)–(2l) correspond to the constraints imposed by operational limitations, where the lower bound problem data $(\cdot)^{\min}$ and the upper bound problem data $(\cdot)\max$ determine the boundaries of the feasible regions of power, current, as well as voltages in the network. Note that if a bus $k$ is not a generator bus, then there is no power generation at that bus and, thus, $p_k^{\mathrm{G}} + jq_k^{\mathrm{G}} = 0$. Such situations can be easily modeled by letting

$$p_k^{\mathrm{G,min}} = p_k^{\mathrm{G,max}} = q_k^{\mathrm{G,min}} = q_k^{\mathrm{G,max}} = 0, \qquad k \in \mathcal{N} \backslash \mathcal{G}. \tag{3}$$

The constraints (2e), (2f), and (2l) are nonconvex, which, in turn, makes problem (2) nonconvex. In fact, the problem is NP-hard [3]. Thus, it hinders efficient algorithms from achieving optimality. However, in the sequel, we design an efficient algorithm to address problem (2) in a decentralized manner.

### B. Distributed Formulation

In this section, we derive an equivalent formulation of problem (2), where all of the constraints, except for a *single consistency constraint*, are decoupled among the buses. In particular, the resulting formulation is in the form of a general consensus problem [29, Sec. 7.2], where the fully decentralized implementation can be realized, without any coordination of a central authority. More generally, the proposed formulation can be easily adapted to accomplish decoupling among subsets of buses, each of which corresponds to buses located in a given area, for example, multiarea OPF [15].

We start by identifying the coupling constraints of problem (2). From constraint (2b), note that the current injection of each bus is affected by the voltages of its neighbors and by its own voltage. Therefore, constraint (2b) introduces coupling between neighbors. To decouple constraint (2b), we let each bus maintain local copies of the neighbors' voltages and then enforce them to agree by introducing consistency constraints.

To formally express the idea from before, we first denote by $\mathcal{N}_k$ the set of bus $k$ itself and its neighboring buses, thsat is, $\mathcal{N}_k = \{k\} \cup \{n | (k,n) \in \mathcal{L}\}$. Copies of real and imaginary parts of the voltages corresponding to buses in $\mathcal{N}_k$ are denoted by $\mathbf{v}_k^{\mathrm{Re}} \in \mathbb{R}^{|\mathcal{N}_k|}$ and $\mathbf{v}_k^{\mathrm{Im}} \in \mathbb{R}^{|\mathcal{N}_k|}$, respectively. For notational convenience, we let $(\mathbf{v}_k^{\mathrm{Re})_1} = v_k^{\mathrm{Re}}$ and $(\mathbf{v}_k^{\mathrm{Im})_1} = v_k^{\mathrm{Im}}$. We refer to $\mathbf{v}^{\mathrm{Re}}$ and $\mathbf{v}^{\mathrm{Im}}$ as *real and imaginary net variables*, respectively. Note that the copies of either the net variable $v_k^{\mathrm{Re}}$ or $v_k^{\mathrm{Im}}$ are shared among $|\mathcal{N}_k|$ entities, which we call the degree of net variable $v_k^{\mathrm{Re}}$ or $v_k^{\mathrm{Im}}$. The consistency constraints are given by

$$\mathbf{v}_k^{\mathrm{Re}} = \mathbf{E}_k \mathbf{v}^{\mathrm{Re}}, \qquad \mathbf{v}_k^{\mathrm{Im}} = \mathbf{E}_k \mathbf{v}^{\mathrm{Im}} \tag{4}$$

where $\mathbf{E}_k \in \mathbb{R}^{|\mathcal{N}_k| \times N}$ is given by

$$(\mathbf{E}_k)_{ls} = \begin{cases} 1, & \text{if } \left(\mathbf{v}_k^{\mathrm{Re}}\right)_l \text{ is a local copy of } v_s^{\mathrm{Re}} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Note that (4) ensures the agreement of the copies of the net variables and that for any bus $k$, either $\mathbf{v}_k^{\mathrm{Re}}$ or $\mathbf{v}_k^{\mathrm{Im}}$ is local in the sense that they depend only on neighbors.

The constraints (2b)–(2l) of problem (2) can be written by using local variables $\mathbf{v}_k^{\mathrm{Re}}$ and $\mathbf{v}_k^{\mathrm{Im}}$. In particular, we can equivalently list them as follows:

$$i_k^{\mathrm{Re}} + j i_k^{\mathrm{Im}} = \mathbf{g}_k^{\mathrm{T}} \mathbf{v}_k^{\mathrm{Re}} - \mathbf{b}_k^{\mathrm{T}} \mathbf{v}_k^{\mathrm{Im}} + j \left( \mathbf{b}_k^{\mathrm{T}} \mathbf{v}_k^{\mathrm{Re}} + \mathbf{g}_k^{\mathrm{T}} \mathbf{v}_k^{\mathrm{Im}} \right) \quad (6a)$$

$$p_k + j q_k = p_k^{\mathrm{G}} - p_k^{\mathrm{D}} + j \left( q_k^{\mathrm{G}} - q_k^{\mathrm{D}} \right) \quad (6b)$$

$$\bar{\mathbf{i}}_k^{\mathrm{Re}} + j \bar{\mathbf{i}}_k^{\mathrm{Im}} = \mathbf{C}_k \mathbf{v}_k^{\mathrm{Re}} + \mathbf{D}_k \mathbf{v}_k^{\mathrm{Im}} + j \left( \mathbf{D}_k \mathbf{v}_k^{\mathrm{Re}} - \mathbf{C}_k \mathbf{v}_k^{\mathrm{Im}} \right) \quad (6c)$$

$$p_k + j q_k = (\mathbf{v}_k^{\mathrm{Re}})_1 i_k^{\mathrm{Re}} + (\mathbf{v}_k^{\mathrm{Im}})_1 i_k^{\mathrm{Im}} + j \left( (\mathbf{v}_k^{\mathrm{Im}})_1 i_k^{\mathrm{Re}} - (\mathbf{v}_k^{\mathrm{Re}})_1 i_k^{\mathrm{Im}} \right) \quad (6d)$$

$$\bar{\mathbf{p}}_k + j \bar{\mathbf{q}}_k = (\mathbf{v}_k^{\mathrm{Re}})_1 \bar{\mathbf{i}}_k^{\mathrm{Re}} + (\mathbf{v}_k^{\mathrm{Im}})_1 \bar{\mathbf{i}}_k^{\mathrm{Re}} + j \left( (\mathbf{v}_k^{\mathrm{Im}})_1 \bar{\mathbf{i}}_k^{\mathrm{Re}} - (\mathbf{v}_k^{\mathrm{Re}})_1 \bar{\mathbf{i}}_k^{\mathrm{Im}} \right) \quad (6e)$$

$$p_k^{\mathrm{G,min}} \le p_k^{\mathrm{G}} \le p_k^{\mathrm{G,max}}, \quad (6f)$$

$$q_k^{\mathrm{G,min}} \le q_k^{\mathrm{G}} \le q_k^{\mathrm{G,max}}, \quad (6g)$$

$$(\bar{\mathbf{i}}_k^{\mathrm{Re}})_r^2 + (\bar{\mathbf{i}}_k^{\mathrm{Im}})_r^2 \le (\bar{\mathbf{i}}_k^{\mathrm{max}})_r^2, \quad r = 1, \ldots, |\mathcal{N}_k| - 1 \quad (6h)$$

$$(\bar{\mathbf{p}}_k)_r^2 + (\bar{\mathbf{q}}_k)_r^2 \le (\bar{\mathbf{s}}_k^{\mathrm{max}})_r^2, \quad r = 1, \ldots, |\mathcal{N}_k| - 1 \quad (6i)$$

$$|(\bar{\mathbf{p}}_k)_r| \le (\bar{\mathbf{p}}_k^{\mathrm{max}})_r, \quad r = 1, \ldots, |\mathcal{N}_k| - 1, \quad (6j)$$

$$(\mathbf{v}_k^{\mathrm{min}})_r^2 \le (\mathbf{v}_k^{\mathrm{Re}})_r^2 + (\mathbf{v}_k^{\mathrm{Im}})_r^2 \le (\mathbf{v}_k^{\mathrm{max}})_r^2, \quad r = 1, \ldots |\mathcal{N}_k| \quad (6k)$$

where $k \in \mathcal{N}$, $\bar{\mathbf{i}}_k^{\mathrm{Re}} = (i_{kl}^{\mathrm{Re}})_{l \in \mathcal{N}_k \setminus \{k\}}$, $\bar{\mathbf{i}}_k^{\mathrm{Im}} = (i_{kl}^{\mathrm{Im}})_{l \in \mathcal{N}_k \setminus \{k\}}$, $\bar{\mathbf{p}}_k = (p_{kl})_{l \in \mathcal{N}_k \setminus \{k\}}$, and $\bar{\mathbf{q}}_k = (q_{kl})_{l \in \mathcal{N}_k \setminus \{k\}}$, with the order kept preserved as in $(\mathbf{v}_k^{\mathrm{Re}})_{1:|\mathcal{N}_k|}$ and $(\mathbf{v}_k^{\mathrm{Im}})_{1:|\mathcal{N}_k|}$. In addition, $\mathbf{g}_k$ (or $\mathbf{b}_k$) in constraint (6a) are obtained by first extracting the $k$th column of $\mathbf{G}$ (respectively, $\mathbf{B}$) and then extracting the rows corresponding to the buses in $\mathcal{N}_k$, where the order of the components are preserved as in $\mathbf{v}_k^{\mathrm{Re}}$ and $\mathbf{v}_k^{\mathrm{Im}}$. In addition, $\mathbf{C}_k \in \mathbb{R}^{(|\mathcal{N}_k|-1) \times |\mathcal{N}_k|}$ and $\mathbf{D}_k \in \mathbb{R}^{(|\mathcal{N}_k|-1) \times |\mathcal{N}_k|}$ in constraint (6c) are given by

$$\mathbf{C}_k = \begin{pmatrix} (\mathbf{g}_k)_2 & -(\mathbf{g}_k)_2 & \cdots & 0 \\ (\mathbf{g}_k)_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{g}_k)_{|\mathcal{N}_k|-1} & 0 & \cdots & -(\mathbf{g}_k)_{|\mathcal{N}_k|} \end{pmatrix} \quad (7)$$

$$\mathbf{D}_k = \begin{pmatrix} (\mathbf{b}_k)_2 & -(\mathbf{b}_k)_2 & \cdots & 0 \\ (\mathbf{b}_k)_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{b}_k)_{|\mathcal{N}_k|-1} & 0 & \cdots & -(\mathbf{b}_k)_{|\mathcal{N}_k|} \end{pmatrix}. \quad (8)$$

Moreover, $\mathbf{v}_k^{\mathrm{min}}, \mathbf{v}_k^{\mathrm{max}}, \bar{\mathbf{i}}_k^{\mathrm{max}}, \bar{\mathbf{s}}_k^{\mathrm{max}}$, and $\bar{\mathbf{p}}_k^{\mathrm{max}}$ of constraints (6f)–(6k) are chosen in a straightforward manner [cf. (2g)–(2l)].

Finally, for notational convenience, associated with each bus $k$, we denote by

$$\mathbf{z}_k = \left(p_k^{\mathrm{G}}, q_k^{\mathrm{G}}, p_k, q_k, i_k^{\mathrm{Re}}, i_k^{\mathrm{Im}}, \mathbf{v}_k^{\mathrm{Re}}, \mathbf{v}_k^{\mathrm{Im}}, \bar{\mathbf{i}}_k^{\mathrm{Re}}, \bar{\mathbf{i}}_k^{\mathrm{Im}}, \bar{\mathbf{p}}_k, \bar{\mathbf{q}}_k\right) \quad (9)$$

the local variables of bus $k$, by $\boldsymbol{\alpha}_k(\mathbf{z}_k) = 0$ the affine constraints (6a)–(6c), by $\boldsymbol{\lambda}_k(\mathbf{z}_k) = 0$ the nonlinear equality constraint (6d), by $\boldsymbol{\mu}_k(\mathbf{z}_k) = 0$ the nonlinear equality constraint (6e), by $\boldsymbol{\beta}_k(\mathbf{z}_k) \le 0$ the linear convex inequality constraints (6f), (6g), (6j), and by $\boldsymbol{\gamma}_k(\mathbf{z}_k) \le 0$ the nonlinear convex inequality constraints (6h) and (6i) as we will see next.[2]

Now we can express the distributed formulation of problem (2) as

$$\min \quad \sum_{k \in \mathcal{G}} f_k^{\mathrm{G}} \left(p_k^{\mathrm{G}}\right) \quad (10a)$$

$$\text{s.t.} \quad \mathbf{z}_k = \left(p_k^{\mathrm{G}}, q_k^{\mathrm{G}}, p_k, q_k, i_k^{\mathrm{Re}}, i_k^{\mathrm{Im}}, \mathbf{x}_k^{\mathrm{Re}}, \mathbf{x}_k^{\mathrm{Im}}, \bar{\mathbf{i}}_k^{\mathrm{Re}}, \bar{\mathbf{i}}_k^{\mathrm{Im}}, \bar{\mathbf{p}}_k, \bar{\mathbf{q}}_k\right), \quad k \in \mathcal{N}, \quad (10b)$$

$$(\boldsymbol{\alpha}_k(\mathbf{z}_k), \boldsymbol{\lambda}_k(\mathbf{z}_k), \boldsymbol{\mu}_k(\mathbf{z}_k)) = \mathbf{0}, \quad k \in \mathcal{N}, \quad (10c)$$

$$(\boldsymbol{\beta}_k(\mathbf{z}_k), \boldsymbol{\gamma}_k(\mathbf{z}_k)) \le \mathbf{0}, \quad k \in \mathcal{N}, \quad (10d)$$

$$(\mathbf{v}_k^{\mathrm{min}})_r^2 \le (\mathbf{v}_k^{\mathrm{Re}})_r^2 + (\mathbf{v}_k^{\mathrm{Im}})_r^2 \le (\mathbf{v}_k^{\mathrm{max}})_r^2, \quad r = 1, \ldots, |\mathcal{N}_k|, \quad k \in \mathcal{N}, \quad (10e)$$

$$\mathbf{v}_k^{\mathrm{Re}} + j \mathbf{v}_k^{\mathrm{Im}} = \mathbf{E}_k \mathbf{v}^{\mathrm{Re}} + j \mathbf{E}_k \mathbf{v}^{\mathrm{Im}}, \quad k \in \mathcal{N} \quad (10f)$$

where the variables are $p_k^{\mathrm{G}}$, $q_k^{\mathrm{G}}$, $p_k$, $q_k$, $i_k^{\mathrm{Re}}$, $i_k^{\mathrm{Im}}$, $\mathbf{v}_k^{\mathrm{Re}}$, $\mathbf{v}_k^{\mathrm{Im}}$, $\bar{\mathbf{i}}_k^{\mathrm{Re}}$, $\bar{\mathbf{i}}_k^{\mathrm{Im}}$, $\bar{\mathbf{p}}_k$, $\bar{\mathbf{q}}_k$, $\mathbf{z}_k$ for $k \in \mathcal{N}$ and $\mathbf{v}^{\mathrm{Re}}$ and $\mathbf{v}^{\mathrm{Im}}$. Note that (10f) establishes the consistency constraints [cf. (4)], which affirms the consistency among neighbor voltages. The coupling in the original centralized formulation (2) has been subsumed in the consistency constraint (10f), which results in the form of general consensus problem [29, Sec. 7.2], where decomposition methods can be gracefully applied.

## III. Distributed Solution Method

In this section, we present our distributed algorithm to the OPF problem (10). In particular, we use the ADMM as the basis for our algorithm development, where we have fast convergence properties, compared to the dual decomposition [29]. The ADMM is also promising in the sense that it works on many nonconvex problems as a good heuristic [29, Sec. 9]. Once the solution method is established, we investigate the properties in Section IV.

### A. Outline of the Algorithm

For notational simplicity, we let $\mathbf{v}_k$ and $\bar{\mathbf{E}}_k$ denote $(\mathbf{v}_k^{\mathrm{Re}}, \mathbf{v}_k^{\mathrm{Im}})$ and $\mathrm{diag}(\mathbf{E}_k, \mathbf{E}_k)$, respectively, for each $k \in \mathcal{N}$. Moreover, we let $\mathbf{v}$ denote $(\mathbf{v}^{\mathrm{Re}}, \mathbf{v}^{\mathrm{Im}})$. The ADMM essentially minimizes the augmented Lagrangian associated with the problem in an iterative manner. Particularized to our problem (10), the partial augmented Lagrangian with respect to the consistency constraints (10f) (i.e., $\mathbf{v}_k = \bar{\mathbf{E}}_k \mathbf{v}$) is given by

$$L_\rho \left(\mathbf{p}^{\mathrm{G}}, (\mathbf{v}_k)_{k \in \mathcal{N}}, \mathbf{v}, (\mathbf{y}_k)_{k \in \mathcal{N}}\right) = \sum_{k \in \mathcal{G}} f_k^{\mathrm{G}} \left(p_k^{\mathrm{G}}\right) + \sum_{k \in \mathcal{N}} \left( \mathbf{y}_k^{\mathrm{T}} (\mathbf{v}_k - \bar{\mathbf{E}}_k \mathbf{v}) + \frac{\rho}{2} \|\mathbf{v}_k - \bar{\mathbf{E}}_k \mathbf{v}\|_2^2 \right) \quad (11)$$

[2]The functions $\beta_k$ and $\gamma_k$ depend on $p_k^{\mathrm{G,min}}$, $p_k^{\mathrm{G,max}}$, $q_k^{\mathrm{G,min}}$, $q_k^{\mathrm{G,max}}$, $\bar{\mathbf{i}}_k^{\mathrm{max}}$, $\bar{\mathbf{s}}_k^{\mathrm{max}}$, and $\bar{\mathbf{p}}_k^{\mathrm{max}}$ which are intentionally omitted for clarity and space reasons.

where $\mathbf{y}_k$ is the dual variable associated with (10f), and $\rho$ is called the *penalty parameter*. Together, with the separability of (11) among $k \in \mathcal{N}$, the steps of ADMM are formally expressed below.

---

Algorithm 1: ADMM for distributed OPF (ADMM-DOPF)

1) Initialization: Set $n = 0$ and initialize $\mathbf{y}_k^{(n)}$ and $\mathbf{v}^{(n)}$.
2) Private variable update: Set $\mathbf{y}_k = \mathbf{y}_k^{(n)}$ and $\mathbf{v} = \mathbf{v}^{(n)}$. Each bus $k \in \mathcal{N}$ updates $\mathbf{x}_k$ locally, where we let $(\mathbf{z}_k^{(n+1)}, \mathbf{u}_k^{(n+1)})$ be the primal and dual (possibly) optimal variables achieved for the following problem:

$$\min \quad f_k^{\mathrm{G}}(p_k^{\mathrm{G}}) + \mathbf{y}_k^{\mathrm{T}}(\mathbf{v}_k - \bar{\mathbf{E}}_k\mathbf{v}) + \frac{\rho}{2}||\mathbf{v}_k - \bar{\mathbf{E}}_k\mathbf{v}||_2^2 \tag{12a}$$

$$\text{s.t.} \quad \mathbf{z}_k = (p_k^{\mathrm{G}}, q_k^{\mathrm{G}}, p_k, q_k, i_k^{\mathrm{Re}}, i_k^{\mathrm{Im}}, \mathbf{v}_k^{\mathrm{Re}}, \mathbf{v}_k^{\mathrm{Im}} \bar{\mathbf{i}}_k^{\mathrm{Re}}, \bar{\mathbf{i}}_k^{\mathrm{Im}}, \bar{\mathbf{p}}_k, \bar{\mathbf{q}}_k) \tag{12b}$$

$$\boldsymbol{\alpha}_k(\mathbf{z}_k) = \mathbf{0} \tag{12c}$$

$$\boldsymbol{\lambda}_k(\mathbf{z}_k) = \mathbf{0} \tag{12d}$$

$$\boldsymbol{\mu}_k(\mathbf{z}_k) = \mathbf{0} \tag{12e}$$

$$\boldsymbol{\beta}_k(\mathbf{z}_k) \leq \mathbf{0} \tag{12f}$$

$$\boldsymbol{\gamma}_k(\mathbf{z}_k) \leq \mathbf{0} \tag{12g}$$

$$(\mathbf{v}_k^{\min})_r^2 \leq (\mathbf{v}_k^{\mathrm{Re}})_r^2 + (\mathbf{v}_k^{\mathrm{Im}})_r^2 \leq (\mathbf{v}_k^{\max})_r^2 \quad r = 1, \ldots, |\mathcal{N}_k| \tag{12h}$$

where the variables are $p_k^{\mathrm{G}}$, $q_k^{\mathrm{G}}$, $p_k$, $q_k$, $i_k^{\mathrm{Re}}$, $i_k^{\mathrm{Im}}$, $\mathbf{v}_k^{\mathrm{Re}}$, $\mathbf{v}_k^{\mathrm{Im}}$, $\bar{\mathbf{i}}_k^{\mathrm{Re}}$, $\bar{\mathbf{i}}_k^{\mathrm{Im}}$, $\bar{\mathbf{p}}_k$, $\bar{\mathbf{q}}_k$, and $\mathbf{z}_k$. We denote by $\mathbf{v}_k^{(n+1)}$, the part of $\mathbf{z}_k^{(n+1)}$ corresponding to $(\mathbf{v}_k^{\mathrm{Re}}, \mathbf{v}_k^{\mathrm{Im}})$.

3) Net variable update: We let $\mathbf{v}^{(n+1)}$ be the solution to the problem

$$\min \sum_{k \in \mathcal{N}} \mathbf{y}_k^{\mathrm{T}} \left( \mathbf{v}_k^{(n+1)} - \bar{\mathbf{E}}_k\mathbf{v} \right) + \frac{\rho}{2} \left\| \mathbf{v}_k^{(n+1)} - \bar{\mathbf{E}}_k\mathbf{v} \right\|_2^2 \tag{13}$$

where the variable is $\mathbf{v}$.

4) Dual variable update: Each bus $k \in \mathcal{N}$ updates its dual variable $\mathbf{y}_k$ as

$$\mathbf{y}_k^{(n+1)} = \mathbf{y}_k^{(n)} + \rho \left( \mathbf{v}_k^{(n+1)} - \bar{\mathbf{E}}_k\mathbf{v}^{(n+1)} \right). \tag{14}$$

5) Stopping criterion: Set $n := n+1$. If the stopping criterion is not met, go to step 2, otherwise STOP and return $(\mathbf{z}^{(n)}, \mathbf{v}^{(n)}, \mathbf{u}^{(n)}, \mathbf{y}^{(n)}) = ((\mathbf{z}_k^{(n)})_{k \in \mathcal{N}}, \mathbf{v}^{(n)}, (\mathbf{u}_k^{(n)})_{k \in \mathcal{N}}, (\mathbf{y}_k^{(n)})_{k \in \mathcal{N}})$.

---

The first step initializes the net and dual variables. In the second step, each bus solves a nonconvex optimization problem in order to update its private variable (see Section III-B). In the third step, the net variable is updated by solving the unconstrained quadratic optimization problem (13), which has a closed-form solution. The net variable update can be performed in a distributed fashion with a light communication protocol (see Section III-D). The fourth step is the dual variable update, which can be performed locally on each bus (see Section III-D). The fifth step is the stopping criterion. Natural stopping criterions include: 1) running the ADMM-DOPF algorithm for a fixed number of iterations; 2) running the ADMM-DOPF algorithm until the decrement between the local and net variables of each bus $k$ ($\|\mathbf{E}_k\mathbf{v} - \mathbf{v}_k\|_2$) is below a predefined threshold; 3) running the ADMM-DOPF algorithm until the objective value decrement between two successive iterations is below a predefined threshold. In the sequel, we discuss in detail the algorithm steps (12)–(14).

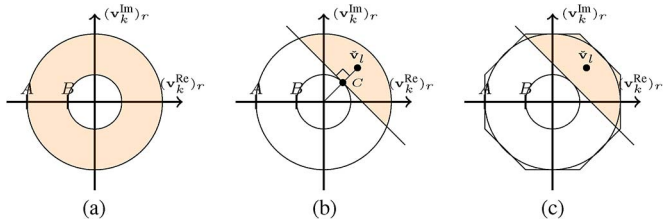


Fig. 1. Feasible set of $((\mathbf{v}_k^{\mathrm{Re}})_r, (\mathbf{v}_k^{\mathrm{Im}})_r)$ where $A = (\mathbf{v}_k^{\max})_r$ and $B = (\mathbf{v}_k^{\min})_r$. (a) $\mathcal{X}_r^k$; (b) $\check{\mathcal{X}}_r^k$; (c) $\check{\mathcal{X}}_r^k$.

### B. Subproblems: Private Variable Update

In this section, problem (12) is considered. Since Problem (12) is NP-hard, only exponentially complex global methods can guarantee its optimality. We capitalize on sequential convex approximations [31] to design an algorithm, which is efficient compared to global methods. Similar techniques are used in [30] for centralized OPF, which we use as the basis for designing our subproblem algorithm.

We start by noting that constraints (12b), (12c), (12f), and (12g) are convex as opposed to constraints (12h), (12d), and (12e), which are clearly nonconvex. The idea is to approximate the nonconvex constraints.

In the case of (12h), we note that for any $r \in \{1, \ldots, |\mathcal{N}_k|\}$, the values of $(\mathbf{v}_k^{\mathrm{Re}})_l$ and $(\mathbf{v}_k^{\mathrm{Im}})_l$ represent a donut, see Fig. 1(a). In other words, the 2-D set

$$\mathcal{X}_r^k = \left\{ \left( (\mathbf{v}_k^{\mathrm{Re}})_r, (\mathbf{v}_k^{\mathrm{Im}})_r \right) \in \mathbb{R}^2 \,\middle|\, (\mathbf{v}_k^{\min})_r^2 \leq (\mathbf{v}_k^{\mathrm{Re}})_r^2 + (\mathbf{v}_k^{\mathrm{Im}})_r^2 \leq (\mathbf{v}_k^{\max})_r^2 \right\}$$

is a donut, which is clearly nonconvex.

We approximate the nonconvex set $\mathcal{X}_r^k$ by considering a convex subset of $\mathcal{X}_r^k$ instead, which we denote by $\check{\mathcal{X}}_r^k$, see Fig. 1(b). To do this, we simply consider the hyperplane tangent to the inner circle of the donut at the point $C$ in Fig. 1(b). Specifically, given a point $((\check{\mathbf{v}}_k^{\mathrm{Re}})_r, (\check{\mathbf{v}}_k^{\mathrm{Im}})_r) \in \mathcal{X}_r^k$, $\check{\mathcal{X}}_r^k$ is the intersection of $\mathcal{X}_r^k$ and the halfspace

$$\left\{ \left( (\mathbf{v}_k^{\mathrm{Re}})_r, (\mathbf{v}_k^{\mathrm{Im}})_r \right) \in \mathbb{R}^2 \,\middle|\, a_r \left( \mathbf{v}_k^{\mathrm{Re}} \right)_r + b_r \left( \mathbf{v}_k^{\mathrm{Im}} \right)_r \geq c_r \right\} \tag{15}$$

where

$$a_r = \operatorname{sign}\left( \check{\mathbf{v}}_k^{\mathrm{Re}} \right)_r \sqrt{\frac{\left( \mathbf{v}_k^{\min} \right)_r^2}{1 + \left( \left( \check{\mathbf{v}}_k^{\mathrm{Im}} \right)_r / \left( \check{\mathbf{v}}_k^{\mathrm{Re}} \right)_r \right)^2}}$$

$$b_r = a_r \left( \frac{\left( \check{\mathbf{v}}_k^{\mathrm{Im}} \right)_r}{\left( \check{\mathbf{v}}_k^{\mathrm{Re}} \right)_r} \right), \qquad c_r = \left( \mathbf{v}_k^{\min} \right)_r^2$$

if $(\check{\mathbf{v}}_k^{\mathrm{Re}})_r \neq 0$ and

$$a_r = 0, \qquad b_r = \operatorname{sign}\left( \check{\mathbf{v}}_k^{\mathrm{Im}} \right)_r, \qquad c_r = \left( \mathbf{v}_k^{\min} \right)_r$$

if $(\check{\mathbf{v}}_k^{\mathrm{Re}})_r = 0$. In the case of nonlinear nonconvex constraints (12d) and (12e), we capitalized on the well-known Taylor's approximation. Specifically, given a point $\hat{\mathbf{z}}_k$, we denote by $\hat{\boldsymbol{\lambda}}_k^{\hat{\mathbf{z}}_k}$ the first-order Taylor's approximation of $\boldsymbol{\lambda}_k$ at $\hat{\mathbf{z}}_k$. Similarly, we denote by $\hat{\boldsymbol{\mu}}_k^{\hat{\mathbf{z}}_k}$, the first-order Taylor's approximation of $\boldsymbol{\mu}_k$ at $\hat{\mathbf{z}}_k$. The approximation is refined in an iterative manner until a stopping criterion is satisfied.

It is worth noting that to construct the functions $\hat{\boldsymbol{\mu}}_k^{\hat{\mathbf{z}}_k}$ and $\hat{\boldsymbol{\lambda}}_k^{\hat{\mathbf{z}}_k}$, one only needs the values of $\hat{\mathbf{v}}_k$, where $\hat{\mathbf{v}}_k$ is the component of $\hat{\mathbf{z}}_k$ corresponding to $(\mathbf{v}_k^{\mathrm{Re}}, \mathbf{v}_k^{\mathrm{Im}})$.[3]

By using the constraint approximations discussed before, we design a subroutine to perform step 2 of the ADMM-DOPF algorithm. The outline of this successive approximation algorithm is given as follows.

---

Algorithm 2: Subroutine for step 2 of the ADMM-DOPF

1) Initialize: Given $\mathbf{v}$ and $\mathbf{y}_k$ from ADMM-DOPF $n$th iteration. Set $(\mathbf{v}^{\mathrm{Re}}, \mathbf{v}^{\mathrm{Im}}) = \mathbf{v}$. For all $r \in \{1, \ldots, |\mathcal{N}_k|\}$, set $((\check{\mathbf{v}}_k^{\mathrm{Re}})_r, (\check{\mathbf{v}}_k^{\mathrm{Im}})_r) = ((\mathbf{E}_k\mathbf{v}^{\mathrm{Re}})_r, (\mathbf{E}_k\mathbf{v}^{\mathrm{Im}})_r)$ and construct $\check{\mathcal{X}}_r^k$. Let $m = 1$ and initialize $\hat{\mathbf{z}}_k$.
2) Solve the approximated subproblem

$$\min \quad f_k^{\mathrm{G}}\left(p_k^{\mathrm{G}}\right) + \mathbf{y}_k^{\mathrm{T}}(\mathbf{v}_k - \bar{\mathbf{E}}_k\mathbf{v}) + \frac{\rho}{2}\|\mathbf{v}_k - \bar{\mathbf{E}}_k\mathbf{v}\|_2^2 \tag{16a}$$

$$\text{s.t.} \quad \mathbf{z}_k = \left(p_k^{\mathrm{G}}, q_k^{\mathrm{G}}, p_k, q_k, i_k^{\mathrm{Re}}, i_k^{\mathrm{Im}}, \mathbf{v}_k^{\mathrm{Re}}, \mathbf{v}_k^{\mathrm{Im}}, \bar{\mathbf{i}}_k^{\mathrm{Re}}, \bar{\mathbf{i}}_k^{\mathrm{Im}}, \bar{\mathbf{p}}_k, \bar{\mathbf{q}}_k\right) \tag{16b}$$

$$\boldsymbol{\alpha}_k(\mathbf{z}_k) = \mathbf{0} \tag{16c}$$

$$\hat{\boldsymbol{\lambda}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k) = \mathbf{0} \tag{16d}$$

$$\hat{\boldsymbol{\mu}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k) = \mathbf{0} \tag{16e}$$

$$\boldsymbol{\beta}_k(\mathbf{z}_k) \leq \mathbf{0} \tag{16f}$$

$$\boldsymbol{\gamma}_k(\mathbf{z}_k) \leq \mathbf{0} \tag{16g}$$

$$\left(\left(\mathbf{v}_k^{\mathrm{Re}}\right)_r, \left(\mathbf{v}_k^{\mathrm{Im}}\right)_r\right) \in \check{\mathcal{X}}_r^k, r = 1, \ldots, |\mathcal{N}_k| \tag{16h}$$

where the variables are $p_k^{\mathrm{G}}$, $q_k^{\mathrm{G}}$, $p_k$, $q_k$, $i_k^{\mathrm{Re}}$, $i_k^{\mathrm{Im}}$, $\mathbf{v}_k^{\mathrm{Re}}$, $\mathbf{v}_k^{\mathrm{Im}}$, $\bar{\mathbf{i}}_k^{\mathrm{Re}}$, $\bar{\mathbf{i}}_k^{\mathrm{Im}}$, $\bar{\mathbf{p}}_k$, $\bar{\mathbf{q}}_k$, and $\mathbf{z}_k$. The solution corresponding to the variable $\mathbf{z}_k$ is denoted by $\mathbf{z}_k^{(m)}$ and all of the dual optimal variables are denoted by $\mathbf{u}_k^{(m)}$.
3) Stopping criterion: If stopping criterion is not met, set $\hat{\mathbf{z}}_k = \mathbf{z}_k^{(m)}$, $m := m + 1$ and go to step 2. Otherwise, STOP and return $(\mathbf{z}_k^{(m)}, \mathbf{u}_k^{(m)})$.

---

The initialization in the first step is done by setting $\hat{\mathbf{v}}_k = (\check{\mathbf{v}}_k^{\mathrm{Re}}, \check{\mathbf{v}}_k^{\mathrm{Im}})$, where $\hat{\mathbf{v}}_k$ is the component of $\hat{\mathbf{z}}_k$ corresponding to the variable $(\mathbf{v}_k^{\mathrm{Re}}, \mathbf{v}_k^{\mathrm{Im}})$. The rest of the vector $\hat{\mathbf{z}}_k$ is then initialized according to (6a)–(6e), which have a unique solution when $\hat{\mathbf{v}}_k$ is given. The second step involves solving a convex optimization problem and the third step is the stopping criterion. A natural stopping criterion is to run the algorithm until the decrement between two successive iterations is below a certain predefined threshold, that is, $\|\mathbf{z}_k^{(m+1)} - \mathbf{z}_k^{(m)}\| < \epsilon$ for a given $\epsilon > 0$. However, since $\mathbf{z}_k$ only depends on $\mathbf{v}_k$, the component related to $(\mathbf{v}_k^{\mathrm{Re}}, \mathbf{v}_k^{\mathrm{Im}})$, we use

$$\left\|\mathbf{v}_k^{(m+1)} - \mathbf{v}_k^{(m)}\right\|_2 < \epsilon^{\mathrm{sub}} \tag{17}$$

where $\mathbf{v}_k^{(m)}$ and $\mathbf{v}_k^{(m+1)}$ are the components of $\mathbf{z}_k^{(m)}$ and $\mathbf{z}_k^{(m+1)}$, respectively, corresponding to the variable $\mathbf{v}_k$ and $\epsilon^{\mathrm{sub}} > 0$ is a given threshold. Furthermore, we do not need to reach the minimum accuracy in every ADMM iteration, but only as the ADMM method progresses. Therefore, it might be practical to set an upper bound on the number of iterations, that is

$$m \geq \texttt{max\_iter} \tag{18}$$

for some $\texttt{max\_iter} \in \mathbb{N}$.

### C. On the Use of Quadratic Programming (QP) Solvers

Problem (16) can be efficiently solved by using general interior-point algorithms for convex problems. However, even higher efficiencies are achieved if problem (16) can be handled by *specific* interior-point algorithms. For example, if the objective function (16a) is quadratic, sophisticated QP solvers can be easily employed. See the Appendix for details.

### D. Net Variables and Dual Variable Updates

Note that the net variable $\mathbf{v}^{(n+1)}$ is the unique solution of the unconstrained convex quadratic optimization problem (13), and is given by

$$\mathbf{v}^{(n+1)} = \left(\sum_{k\in\mathcal{N}} \bar{\mathbf{E}}_k^{\mathrm{T}}\bar{\mathbf{E}}_k\right)^{-1} \sum_{k\in\mathcal{N}} \bar{\mathbf{E}}_k^{\mathrm{T}}\left(\mathbf{v}_k^{(n+1)} + \frac{1}{\rho}\mathbf{y}_k^{(n)}\right) \tag{19}$$

$$= \left(\sum_{k\in\mathcal{N}} \bar{\mathbf{E}}_k^{\mathrm{T}}\bar{\mathbf{E}}_k\right)^{-1} \sum_{k\in\mathcal{N}} \bar{\mathbf{E}}_k^{\mathrm{T}}\mathbf{v}_k^{(n+1)} \tag{20}$$

$$= \mathrm{diag}\left(\left(\sum_{k\in\mathcal{N}} \mathbf{E}_k^{\mathrm{T}}\mathbf{E}_k\right)^{-1}, \left(\sum_{k\in\mathcal{N}} \mathbf{E}_k^{\mathrm{T}}\mathbf{E}_k\right)^{-1}\right) \times \left(\sum_{k\in\mathcal{N}} \mathbf{E}_k^{\mathrm{T}}\mathbf{v}_k^{\mathrm{Re}(n+1)}, \sum_{k\in\mathcal{N}} \mathbf{E}_k^{\mathrm{T}}\mathbf{v}_k^{\mathrm{Im}(n+1)}\right) \tag{21}$$

$$= \Bigg(\underbrace{\mathrm{diag}\left(\frac{1}{|\mathcal{N}_1|}, \ldots, \frac{1}{|\mathcal{N}_N|}\right)\sum_{k\in\mathcal{N}} \mathbf{E}_k^{\mathrm{T}}\mathbf{v}_k^{\mathrm{Re}(n+1)}}_{\mathbf{v}^{\mathrm{Re}(n+1)}}, \underbrace{\mathrm{diag}\left(\frac{1}{|\mathcal{N}_1|}, \ldots, \frac{1}{|\mathcal{N}_N|}\right)\sum_{k\in\mathcal{N}} \mathbf{E}_k^{\mathrm{T}}\mathbf{v}_k^{\mathrm{Im}(n+1)}}_{\mathbf{v}^{\mathrm{Im}(n+1)}}\Bigg) \tag{22}$$

[3]This follows directly from the definition of the first-order Taylor approximation and (6a) and (6c).

where (19) follows trivially from the differentiation of the objective function of problem (13), (20) follows by invoking the optimality conditions for problem (13), that is, $\sum_{k\in\mathcal{N}} \bar{\mathbf{E}}_k \mathrm{T}\mathbf{y}_k = 0$, (21) follows from $\bar{\mathbf{E}}_k = \mathrm{diag}(\mathbf{E}_k, \mathbf{E}_k)$, and (22) follows from $\sum_{k\in\mathcal{N}} \mathbf{E}_k \mathrm{T}\mathbf{E}_k = \mathrm{diag}(|\mathcal{N}_1|, \ldots, |\mathcal{N}_N|)$. From (22), it is not difficult to see that any net variable component update is equivalently obtained by averaging its copies maintained among the neighbor nodes. Such averaging can be accomplished by using fully distributed algorithms, such as gossiping [32]. Therefore, step 3 of the ADMM-DOPF algorithm can be carried out in a fully distributed manner.

The dual variable update (14) can be carried out in a fully distributed manner, where every bus increments the current dual variables by a (scaled) discrepancy between current net variables and its own copies of those net variables.

## IV. Properties of the Distributed Solution Method

Recall that the original problem (2) or, equivalently, problem (10) is nonconvex and NP-hard. Therefore, ADMM-based approaches are not guaranteed to converge [29, Sec. 9], though general convergence results are available for the *convex* case [29, Sec. 3.2]. Nevertheless, in the sequel, we highlight some of the convergence properties of our proposed ADMM-DOPF algorithm. In particular, we first illustrate, by using an example, the possible scenarios that can be encountered by Algorithm 2, that is, step 2 of the ADMM-DOPF algorithm. Then, we capitalized on one of the scenarios, which is empirically observed to be the most dominant, in order to characterize the solutions of the ADMM-DOPF algorithm.

### A. Graphical illustration of Algorithm 2

We start by focussing on step 2, the main ingredient of the ADMM-DOPF algorithm. To get insights into the subroutine (i.e., Algorithm 2) performed at step 2, we first rely on a simple graphical interpretation. Here, instead of problem (12), we consider a small dimensional problem to demonstrate some essential ingredients of the analysis. In particular, we consider the convex objective function $f(p, x)$ in place of (12a). Moreover, instead of the nonconvex constraints (12d) and (12e) [cf. (6d) and (6e)], we consider the constraint

$$p = g(x) \tag{23}$$

where $g$ is a nonconvex function, which resembles right-hand side of (6d) and (6e). Finally, instead of the remaining constraints (12b), (12c), (12f), (12g), and (12h) of problem (12), we consider the constraint

$$(p, x) \in \mathcal{Z} \tag{24}$$

where $\mathcal{Z}$ is not a convex set [cf. (12h)]. Thus, the smaller dimensional problem, which resembles subproblem (12) is given by

$$\begin{aligned} \text{minimize} \quad & f(p, x) \\ \text{subject to} \quad & p = g(x) \\ & (p, x) \in \mathcal{Z} \end{aligned} \tag{25}$$

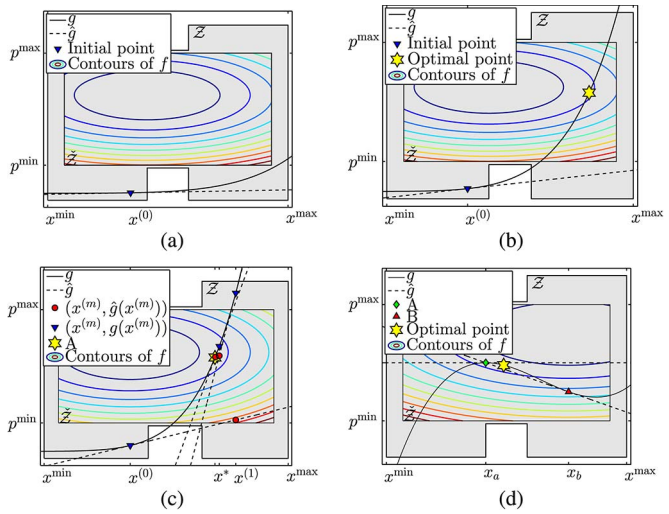


Fig. 2. Graphical illustration of Algorithm 2. (a) Scenario 1, improper approximation of set $\mathcal{Z}$, see (24), makes the approximated problem (26) infeasible. (b) Scenario 2, improper choice of the approximation point $\hat{x} = x^{(0)}$ makes the approximated problem (26) infeasible. (c) Scenario 3, the sequence of approximations eventually converges to a desired point A. (d) Scenario 4, the algorithm jumps between points $A = (x_a, g(x_a))$ and $B = (x_b, g(x_b))$.

where the variables are $p \in \mathbb{R}$ and $x \in \mathbb{R}$. Recall that Algorithm 2 approximates nonconvex functions in constraints (12d) and (12e) of problem (12) by using their first-order Taylor's approximations [see (16d) and (16e)] and the nonconvex constraint (12h) by using a convex constraint [see (16h)]. Particularized to the smaller dimensional problem (25), the approximations pointed above are equivalent to replacing $g$ by its first-order Taylor's approximation $\hat{g}$ and to approximate $\mathcal{Z}$ by some convex set $\check{\mathcal{Z}}$, where $\check{\mathcal{Z}} \subseteq \mathcal{Z}$. The result is the approximated subproblem given by

$$\begin{aligned} \text{minimize} \quad & f(p, x) \\ \text{subject to} \quad & p = \hat{g}(x) = g(\hat{x}) + g'(\hat{x})(x - \hat{x}) \\ & (p, x) \in \check{\mathcal{Z}} \end{aligned} \tag{26}$$

where the variables are $p \in \mathbb{R}$ and $x \in \mathbb{R}$, and $\hat{x}$ represents the point at which the first-order Taylor's approximation is made.

Let us next examine the behavior of Algorithm 2 by considering, instead of problem (16), the representative smaller dimensional problem (26). Recall that the key idea of Algorithm 2 is to iteratively refine the first-order Taylor's approximations $\hat{\boldsymbol{\lambda}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k)$ and $\hat{\boldsymbol{\mu}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k)$ [see steps 2 and 3 of Algorithm 2], until a stopping criterion is satisfied. This behavior is analogously understood from problem (26), by iteratively refining the first-order Taylor's approximation $\hat{g}$ of $g$.

Fig. 2 illustrates the sequential refinement of $\hat{g}$, where the shaded area represents the set $\mathcal{Z}$, the rectangular box represents the convex set $\check{\mathcal{Z}}$, the solid curve represents the function $g$, the dotted lines represent the sequential approximations $\hat{g}$, and the thick solid curves represent the contours of $f$. Note that there are several interesting scenarios, which deserve attention to built intuitively the behavior of Algorithm 2 [see Fig. 2(a)–(d)]. Fig. 2(a) shows the first scenario, where *an improper*

*approximation of set* $\mathcal{Z}$ makes the approximated problem (26) infeasible, irrespective of the choice of $\hat{x}$. In contrast, Fig. 2(b) depicts a scenario, where *an improper choice of the approximation point* $\hat{x}$ makes the approximated problem (26) infeasible. Fig. 2(c) shows a sequence of approximations, which eventually converge to the optimal point "A." Finally, Fig. 2(d) shows a scenario, where a sequence of approximations switch between two points "A" and "B," that is, there is no convergence. Any other scenario can be constructed by combining cases from Fig. 2(a)–(d).

Analogously, the discussion from before suggests that the approximation points $(\hat{\mathbf{z}}_k)_{k\in\mathcal{N}}$ [cf. $\hat{x}$] were used when constructing $\hat{\boldsymbol{\lambda}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k)$ and $\hat{\boldsymbol{\mu}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k)$ [cf. $\hat{g}(x)$] and the approximations used in the set $(\check{\mathcal{X}}_r^k)_{r=1,\ldots,|\mathcal{N}_k|}$ [cf. $\check{\mathcal{Z}}$], can heavily influence the performance of Algorithm 2. Therefore, especially if scenarios 1 and 2 depicted in Fig. 2(a) and (b) occur, during the algorithm iterations, they have to be avoided by changing the initializations. However, extensive numerical experiments show that there are specific choices of $\hat{\mathbf{z}}_k$, and $\check{\mathcal{X}}_r^k$ can make Algorithm 2 often converge to a point as depicted in Fig. 2(c) and barely encounters the scenarios depicted in Fig. 2(a), (b), and (d). See Section V-A for details.

### B. Optimality Properties of the Algorithm 2 Solution

Results obtained in this section are based on the empirical observations (see Section V) that scenario 3 depicted in Fig. 2(c) is more dominant compared to others. In particular, we make the following assumptions.

*Assumption 1:* For any $k\in\mathcal{N}$, there exists $(\mathbf{z}_k^\star,\mathbf{u}_k^\star)$, to which Algorithm 2 can converge. Specifically, there exists $(\mathbf{z}_k^\star,\mathbf{u}_k^\star)$, where $\lim_{m\to\infty}(\mathbf{z}_k^{(m)},\mathbf{u}_k^{(m)})=(\mathbf{z}_k^\star,\mathbf{u}_k^\star)$ for all $k\in\mathcal{N}$. In addition, for all $k\in\mathcal{N}$, the components $(\mathbf{v}_k^{\mathrm{Re}},\mathbf{v}_k^{\mathrm{Im}})$ of $\mathbf{z}_k^\star$, *strictly* satisfy the constraint (16h).

Under Assumption 1, the following assertion can be made:

*Proposition 1:* Suppose Assumption 1 holds. Then, the output $(\mathbf{z}_k^\star,\mathbf{u}_k^\star)$ of Algorithm 2 satisfies Karush-Kuhn-Tucker (KKT) conditions for problem (12).

*Proof:* See Appendix II-A. ■

Combined with our empirical observations that Algorithm 2 almost always converges to a point as depicted in Fig. 2(c) (i.e., Assumption 1 holds), Proposition 1 claims that the point satisfies the first-order necessary conditions for *local optimality*.

### C. Optimality Properties of the ADMM-DOPF Solution

As we already pointed out, there is no guarantee that the eventual output $(\mathbf{z},\mathbf{v},\mathbf{u},\mathbf{y})$ of ADMM-DOPF is optimal, or even feasible to the original problem (10), because the problem is NP-hard. However, Proposition 1, asserts that the eventual output $(\mathbf{z}_k,\mathbf{u}_k)$ of Algorithm 2 is a KKT point for problem (12) solved at step 2 of ADMM-DOPF. One can easily relate this result to characterize the properties of $(\mathbf{z},\mathbf{u})$ of the ADMM-DOPF output, as we will see later. However, the properties of the remaining output $(\mathbf{v},\mathbf{y})$ have yet to be investigated. In this section, combined with the results of Proposition 1, we analyze the optimality properties of the ADMM-DOPF output.

To quantify formally the optimality properties of ADMM-DOPF, we rely on the following definition:

*Definition 1 ($(\delta,\epsilon)$-KKT Optimality):* Consider the possibly nonconvex problem of the form

$$\begin{aligned}\text{minimize}\quad & f_0(\mathbf{x})\\ \text{subject to}\quad & f_i(\mathbf{x})\le 0, i=1,\ldots,q\\ & h_i(\mathbf{x})=0, i=1,\ldots,p\\ & r_i(\mathbf{x})=0,\ i=1,\ldots,s\end{aligned}\tag{27}$$

where $f_0:\mathbb{R}^n\to\mathbb{R}$ is the objective function, $f_i:\mathbb{R}^n\to\mathbb{R},\ i=1,\ldots,q$ are the associated inequality constraint functions, $h_i:\mathbb{R}^n\to\mathbb{R},\ i=1,\ldots,p$ and $r_i:\mathbb{R}^n\to\mathbb{R},\ i=1,\ldots,s$ are the equality constraint functions, and $\mathbf{x}\in\mathbb{R}^n$ is the optimization variable. Moreover, let $\lambda_i$ denote the dual variable associated with constraint $f_i(\mathbf{x})\le 0$, and $\nu_i$ and $\omega_i$ denote the dual variables associated with constraint $h_i(\mathbf{x})=0$ and $r_i(\mathbf{x})=0$, respectively. Then, an arbitrary point $(\mathbf{x}^\star,\lambda_1^\star,\ldots,\lambda_q^\star,\nu_1^\star,\ldots,\nu_p^\star,\omega_1^\star,\ldots,\omega_p^\star)$ is called $(\delta,\epsilon)$-KKT optimal, if

$$f_i(\mathbf{x}^\star)\le 0,\ i=1,\ldots,q \tag{28}$$

$$h_i(\mathbf{x}^\star)=0, i=1,\ldots,q \tag{29}$$

$$(1/s)\sum_{i=1}^{s}\|r_i(\mathbf{x}^\star)\|_2^2=\delta \tag{30}$$

$$\lambda_i^\star\ge 0,\ i=1,\ldots,q \tag{31}$$

$$\lambda_i^\star f_i(\mathbf{x}^\star)=0,\ i=1,\ldots,q \tag{32}$$

$$(1/n)\left\|\nabla_{\mathbf{x}}f_0(\mathbf{x}^\star)+\sum_{i=1}^{q}\lambda_i^\star\nabla_{\mathbf{x}}f_i(\mathbf{x}^\star)+\sum_{i=1}^{p}\nu_i^\star\nabla_{\mathbf{x}}h_i(\mathbf{x}^\star)+\sum_{i=1}^{s}\omega_i^\star\nabla_{\mathbf{x}}r_i(\mathbf{x}^\star)\right\|_2^2=\epsilon. \tag{33}$$

Note that (28)–(33) are closely related to the well-known KKT optimality criterions, see [6, Sec. 5.5.3]. It suggests that the smaller $\delta$ and $\epsilon$ are, better the point $(\mathbf{x}^\star,\lambda_1^\star,\ldots,\lambda_q^\star,\nu_1^\star,\ldots,\nu_p^\star,\omega_1^\star,\ldots,\omega_p^\star)$ to its local optimality. We use Definition 1 to formally analyze the optimality properties of ADMM-DOPF as discussed in the sequel.

Recall that we have used $\mathbf{z}=(\mathbf{z}_k)_{k\in\mathcal{N}}$ to denote the vector of all local primal variables in (9), $\mathbf{v}=(\mathbf{v}^{\mathrm{Re}},\mathbf{v}^{\mathrm{Im}})$ to denote the vector of all net variables, $\mathbf{u}=(\mathbf{u}_k)_{k\in\mathcal{N}}$ to denote the dual variables associated with constraints (10b)–(10e), and, finally, $\mathbf{y}$ to denote the dual variables associated with constraint (10f).

Let us assume that at the termination of ADMM-DOPF, the output corresponding to $\mathbf{v}$ and $\mathbf{y}$ is $\mathbf{v}^\star$ and $\mathbf{y}^\star$, respectively. The output of ADMM-DOPF corresponding to $\mathbf{z}$ and $\mathbf{u}$ are simply the output of Algorithm 2 given by $\mathbf{z}^\star=(\mathbf{z}_k^\star)_{k\in\mathcal{N}}$ and $\mathbf{u}^\star=(\mathbf{u}_k^\star)_{k\in\mathcal{N}}$. However, unlike in convex problems, in the case of problem (10), one cannot take for granted that the consistency constraint (10f) is satisfied (cf. [29, Sec. 3.2.1]). In particular,

$\|\mathbf{v}_k^{(n)} - \bar{\mathbf{E}}_k \mathbf{v}^\star\|_2^2 \to 0$ does not necessarily hold when $n \to \infty$, where $k \in \mathcal{N}$ and $n$ is the ADMM-DOPF iteration index. However, an appropriate choice of the penalty parameter $\rho$ in the ADMM-DOPF algorithm usually allows finding outputs, where the consistency constraints are almost satisfied with a small error floor, which is negligible in real practical implementations as we will see empirically in Section V. For latter use, let us quantify this error floor from $\boldsymbol{\delta}_k$, i.e.,

$$\boldsymbol{\delta}_k = \mathbf{v}_k^\star - \bar{\mathbf{E}}_k \mathbf{v}^\star,\ k \in \mathcal{N}. \tag{34}$$

Now we can formally establish the optimality properties of ADMM-DOPF as follows.

*Proposition 2:* Given Assumption 1 holds, the output $(\mathbf{z}^\star, \mathbf{v}^\star, \mathbf{u}^\star, \mathbf{y}^\star)$ at the termination of ADMM-DOPF is $(a^{-1}\bar{\delta}, b^{-1}\rho^2\bar{\delta})$-KKT optimal, where $\bar{\delta} = \sum_{k\in\mathcal{N}} |\boldsymbol{\delta}_k\|_2^2$, $\rho$ is the penalty parameter used in the ADMM-DOPF iterations, and $a = \text{len}((\boldsymbol{\delta}_k)_{k\in\mathcal{N}})$, $b = \text{len}(\mathbf{z}^\star, \mathbf{v}^\star)$ are normalization factors.

*Proof:* See Appendix II-B. ■

We note that deriving an analytical expression of $\delta$ by using problem (10) data is very difficult. However, we can numerically compute $\delta$ and $\epsilon$ given in Proposition 1 as

$$\delta = a^{-1}\bar{\delta}, \quad \epsilon = b^{-1}\rho^2\bar{\delta}. \tag{35}$$

Extensive numerical experiences show that we usually have very small values for $\delta$. For example, for all considered simulations with $\rho = 10^6$ [see Section V], we have $\delta$ on the order of $10^{-12}$ (or smaller) and $\epsilon$ on the order of $10^{-1}$ (or smaller) after 5000 ADMM-DOPF iterations.

## V. Numerical Results

In this section, we present numerical experiments to illustrate the proposed algorithm. We compare our algorithm with the branch-and-bound algorithm [11], centralized OPF solver provided by Matpower [33], and the SDP relaxation from [3]. In order to study the convergence properties of the algorithm, we evaluate it on four examples that have a (nonzero) duality gap, see Table I, rows 1-4. These four examples come from [10] and [11], and are obtained by making a small modification to standard test examples, see Table I, column 3. It is worth noting that the methods based on the SDP relaxation do not apply here due to the nonzero duality gap [10]. To study the scalability properties of the proposed algorithm, we also evaluate it on two larger examples, see Table I, rows 5-6. The exact specifications of the considered examples are found in Table I and references therein. The objective functions in all examples are quadratic.

The units of real power, reactive power, apparent power, voltage magnitude, and the objective function values are megawatts (MW), MVAr, megavolt-amperes (MVA), per unit (p.u.),[4] and U.S.$/h, respectively. In all six problems, the average power demand of the loads is in the range 10–100 MW and 1–10 MVAr, for the real and reactive powers, respectively.

The simulations were executed in a sequential computational environment, using Matlab version 8.1.0.604 (R2013a) [35].

[4]The voltages base is 400 kV.

TABLE I
Specifications of the Test Problems. The First Column Indicates the Number of Buses. The Second Column Gives the Reference to the Original Problems. The Third Column Show How We Modify the Original Problem ($\bar{p}_k^{\mathrm{D}}$ and $\bar{q}_k^{\mathrm{D}}$ Indicate the Original Problem Data Associated With the Power Demands). The Fourth and Fifth Columns Specify the Number of Generators and Loads, Respectively. The Sixth Column Specifies the Type of Flow Line Limit (FL) Used, If Any, That Is, Which Constraints (2i), (2j), and (2k) Are Included

| $N$ | Original Problem | Modification | # G | # L | FL |
|---|---|---|---|---|---|
| 3 | 3bus [10] | $s_{23}^{\max} = s_{32}^{\max} = 50$ MVA | 3 | 3 | (2j) |
| 9 | Case9 [33] | $q_k^{\mathrm{G,min}} = 10$ MVAr, $\forall k \in \mathcal{G}$<br>$p_k^{\mathrm{D}} = 1.1\bar{p}_k^{\mathrm{D}}$, $\forall k \in \mathcal{N}$ | 3 | 3 | None |
| 14 | IEEE14 [34] | $q_k^{\mathrm{G,min}} = 0$ MVAr, $\forall k \in \mathcal{G}$<br>$p_k^{\mathrm{D}} = 0.1\bar{p}_k^{\mathrm{D}}$, $\forall k \in \mathcal{N}$ | 5 | 11 | None |
| 30 | IEEE30 [34] | $p_k^{\mathrm{D}} = 0.5\bar{p}_{k,k}^{\mathrm{D}}$ $\forall k \in \mathcal{N}$<br>$q_k^{\mathrm{D}} = 0.1\bar{q}_k^{\mathrm{D}}$, $\forall k \in \mathcal{N}$ | 6 | 21 | None |
| 118 | IEEE118 [34] | None | 54 | 99 | None |
| 300 | IEEE300 [34] | None | 69 | 201 | None |

TABLE II
Frequency of the Termination of Algorithm 2 From the Stopping Criteria (18)

| $N$ | max_iter = 10 | max_iter = 20 | max_iter = 1000 | max_iter = 25000 |
|---|---|---|---|---|
| 3 | $2\times10^{-2}\%$ | $5\times10^{-3}\%$ | $5\times10^{-3}\%$ | $5\times10^{-3}\%$ |
| 9 | $0\%$ | $0\%$ | $0\%$ | $0\%$ |
| 14 | $2\times10^{-1}\%$ | $0\%$ | $0\%$ | $0\%$ |
| 30 | $2\times10^{-5}\%$ | $2\times10^{-5}\%$ | $2\times10^{-5}\%$ | $2\times10^{-5}\%$ |
| 118 | $8\times10^{-5}\%$ | $8\times10^{-5}\%$ | $8\times10^{-5}\%$ | $8\times10^{-5}\%$ |
| 300 | $4\times10^{-1}\%$ | $3\times10^{-2}\%$ | $9\times10^{-3}\%$ | $8\times10^{-3}\%$ |

The convex problem (16) is solved with the convex solution method presented in Section III-C, together with the built-in Matlab QP solver quadprog. As a stopping criterion for Algorithm 2, we use $\epsilon^{\mathrm{sub}} = 10^{-10}$ and $\texttt{max\_iter} = 20$, unless stated otherwise. For the ADMM method, we use $\mathbf{v}^{\mathrm{Re}} = (1, \cdots, 1)$, $\mathbf{v}^{\mathrm{Im}} = (0, \cdots, 0)$, and $\mathbf{y} = (0, \cdots, 0)$ as an initial point.

### A. Properties of Algorithm 2

In this subsection, we investigate the convergence properties of Algorithm 2. In particular, we relate the convergence behavior of Algorithm 2 to the analysis in Sections IV-A and IV-B, where four scenarios, or possible outcomes, of Algorithm 2 (Fig. 2) were identified.

During the numerical evaluations, Algorithm 2 was executed 11 060 000 times. Scenarios 1 and 2 [Fig. 2(a) and (b), respectively] are where the approximated subproblem is infeasible and occurred only 6 times ($\approx 0.00008\%$). These occurrences occurred at one of the buses in the 300-bus example, during ADMM iterations 137–143. This suggests even if a particular bus fails to converge with Algorithm 2 in consecutive ADMM iterations, the bus can recover to find a solution in latter ADMM iterations.

To numerically study the occurrences of Scenarios 3 and 4, we run Algorithm 2 with 1) $\texttt{max\_iter} = 20$; 2) $\texttt{max\_iter} = 1000$; and 3) $\texttt{max\_iter} = 25\,000$. In all considered cases, we use $\epsilon^{\mathrm{sub}} = 10^{-10}$ [compare with (17)]. Table II summarizes how frequently the stopping criteria (18) of Algorithm 2 is met.

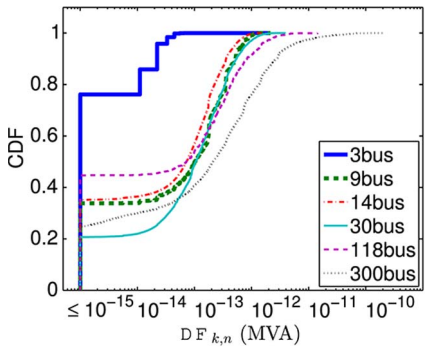


Fig. 3. CDF displaying $\mathrm{DF}_{k,n}$ of (36) for every subproblem $k$ and every ADMM iteration $n$ for each of the four examples.

Note that the entries of Table II suggest an upper bound on the frequencies of Scenario 4. Therefore, Table II shows that the frequencies of Scenario 4 decrease (or unchanged) as `max_iter` is increased. However, the effects are marginal as indicated in the table, especially for `max_iter` $\geq$20. On the other hand, recall that $\epsilon^{\mathrm{sub}} = 10^{-10}$, that is, the decrement of voltages between two successive iterations is below $10^{-10}$ [compare with (17)]. Such an infinitesimal accuracy in the stopping criteria (17) suggests the algorithm's convergence, see Scenario 3, Fig. 2(c). For example, consider the case $N = 3$ and `max_iter` $= 20$ in Table II. From the results, the frequency of termination of Algorithm 2 from the stopping criteria (17) (i.e., Scenario 3) becomes 99.995%. Thus, from Proposition 1, it follows that when `max_iter` $=$ 20, 99.995% of the cases Algorithm 2 converge to a point satisfying the KKT conditions. It is worth noting that for all considered cases, the convergence of the algorithm is in the range 99.99%–100%. The results also suggest that the convergence properties of Algorithm 2 can be improved (see the case $N = 300$) or remain intact (see the cases $N = 3$, 9, 14, 30, and 118) at the expense of the increase in `max_iter`.

Note that the voltages returned by Algorithm 2 are always feasible to problem (12), that is, it satisfies (12h). However, the resulting power injections might be infeasible [compared with (12c)–(12g)]. To measure the feasibility of the returned power injections of *Algorithm 2*, we define the following metric called the degree of feasibility (DF):

$$\mathrm{DF}_{k,n} = \min_{p+jq\in\mathcal{S}_k} \left\| p_k^{(n)} + jq_k^{(n)} - (p+jq) \right\| \qquad (36)$$

where $k$ and $n$ indicate the bus and ADMM iteration, respectively, $p_k^{(n)} + jq_k^{(n)}$ is the returned power injection, and

$$\mathcal{S}_k = \left\{ z\in\mathbb{C} \left\| \begin{array}{l} p_k^{\mathrm{G,min}} - p_k^{\mathrm{D}} \leq \mathrm{Re}(z) \leq p_k^{\mathrm{G,max}} - p_k^{\mathrm{D}} \\ q_k^{\mathrm{G,min}} - q_k^{\mathrm{D}} \leq \mathrm{Im}(z) \leq q_k^{\mathrm{G,max}} - q_k^{\mathrm{D}} \end{array} \right. \right\} \qquad (37)$$

The unit of measurement for $\mathrm{DF}_{k,n}$ is MVA. In order to provide a statistical description of $\mathrm{DF}_{k,n}$ for every execution of Algorithm 2, we consider an empirical cumulative distribution function (CDF) (Fig. 3) and a histogram (Fig. 4), for each example separately. These results suggest that Algorithm 2 returns a feasible solution with high accuracy in all cases, where the worst case accuracy is $2.5 \times 10^{-10}$.

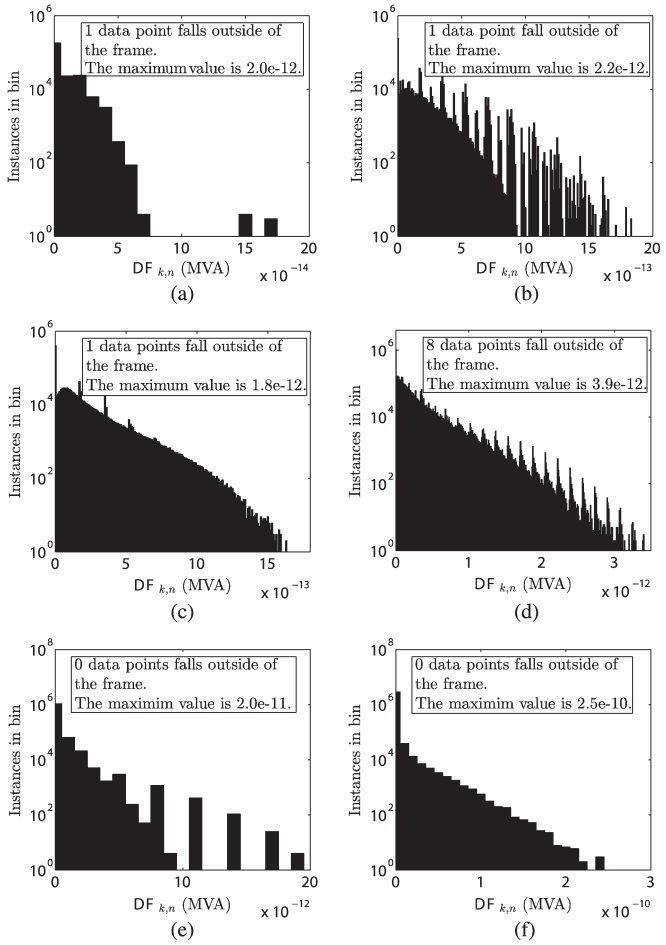


Fig. 4. Histograms displaying $\mathrm{DF}_{k,n}$ in (36) for every subproblem $k$ and every ADMM iteration $n$ for the considered test networks: (a) three-bus networks, (b) nine-bus network, (c) 14-bus network, (d) 30-bus network, (e) 118-bus network, and (f) 300-bus network.

As a consequence of this promising behavior of Algorithm 2, we will proceed under Assumption 1.

### B. Connection to Proposition 2

In this section, we relate the numerical evaluations to Proposition 2. In particular, we inspect the behavior of $\delta$ [compared with (30) and (35)], and $\epsilon$ [compared with (33) and (35)] with respect to $\rho$, which are defined in Section IV-C. The unit of measurements for $\delta$ is p.u.$^2$ and the unit of $\epsilon$ can be interpreted as the square of the decrease/increase in U.S.$/hour with respect to a small perturbation in the variable $\mathbf{z} = (\mathbf{z}_k)_{k\in\mathcal{N}}$.

Fig. 5 depicts $\delta$ at every 500 ADMM iterations, for $\rho = 10^6, \cdots, 10^{13}$. In the 30-bus example, the results are almost identical for $\rho = 10^9, 10^{10}, \cdots, 10^{13}$ and, accordingly, we only include the results for $\rho = 10^6, 10^7, 10^8, 10^{13}$. Since $\delta$ measures the inconsistency between the subproblems, the point returned by ADMM-DOPF can only be considered feasible when $\delta$ has reached acceptable accuracy, that is, $\delta < \gamma$ for some $\gamma > 0$. We do not consider any particular threshold $\gamma$, since we are only interested in observing the convergence behavior. In this aspect, the result shows a promising behavior, as $\delta$ has a

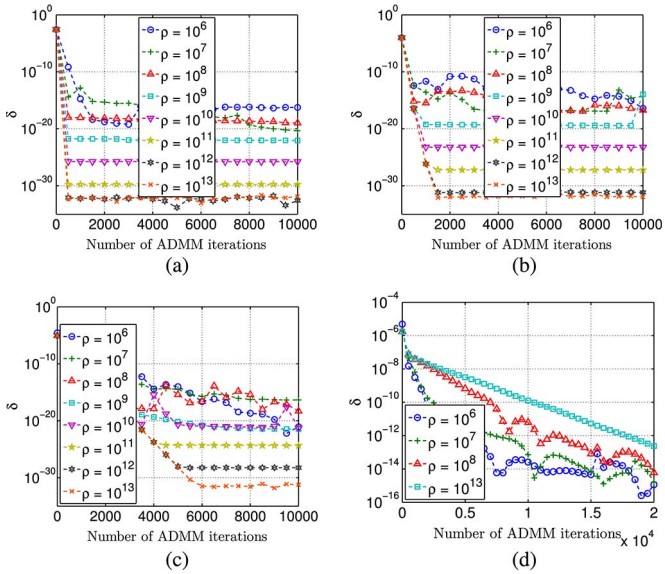


Fig. 5. $\delta$ versus the number of ADMM iterations: (a) three-bus network, (b) nine-bus network, (c) 14-bus network, and (d) 30-bus network.

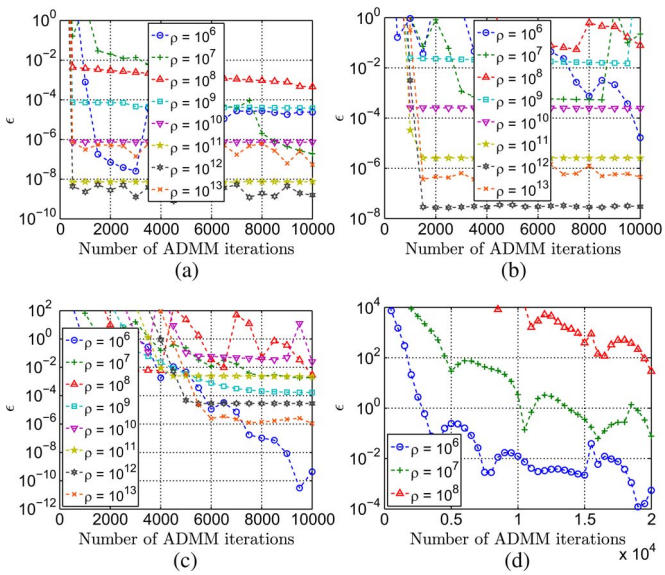


Fig. 6. $\epsilon$ versus the number of ADMM iterations: (a) three-bus network, (b) nine-bus network, (c) 14-bus network, and (d) 30-bus network.

decreasing trend in all cases. Furthermore, for the 3-, 9-, and 14-bus examples, $\delta$ converges to a fixed error floor for the larger values of the penalty parameter $\rho$. In particular, as $\rho$ increases, $\delta$ converges to a point closer to zero, which suggests a negative relationship between $\delta$ and $\rho$. Therefore, this indicates that increasing the penalty parameter enforces higher accuracy of consistency among the subproblems. On the contrary to the 3-, 9-, and 14-bus examples, $\delta$ decreases more slowly when the penalty parameter increases in the case of the 30-bus example. However, in the case of the 30-bus example, $\delta$ is still decreasing after the last iteration considered when $\rho = 10^9, \cdots, 10^{13}$.

Fig. 6 depicts $\epsilon$ at every 500th ADMM iterations, for different $\rho$'s. In contrast to $\delta$, the decreasing trend in $\epsilon$ is not necessary to obtain a feasible solution to the problem. However, under Assumption 1, as $\delta$ and $\epsilon$ go to zero, the algorithm converges

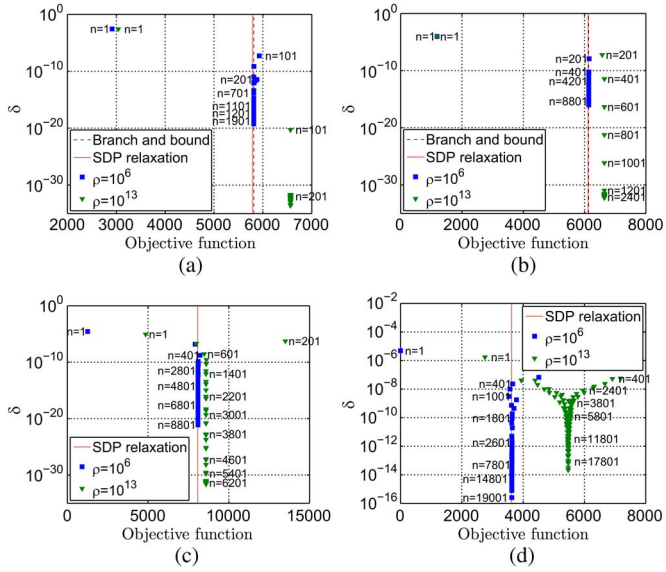


Fig. 7. $\delta$ versus the objective function value: (a) three-bus network; (b) nine-bus network, (c) 14-bus network, and (d) 30- bus network.

to the KKT optimal point. Therefore, the decreasing trend in $\epsilon$, which is observed from the results, is desired. In the case of the 3-, 9-, and 14-bus examples, $\epsilon$ reaches values between $10^{-2}$ and $10^{-11}$ in almost every case. However, in the 30-bus example, only when $\rho = 10^6$ does epsilon reach a value below $10^{-2}$.

### C. Convergence and Scalability Properties

By convention, the objective value of problem (10) is $\infty$ if the problem is infeasible and is given by (10a) if it is feasible. Therefore, when computing the objective function of the problem, one has to verify whether the constraints (10b)–(10f) are feasible or not. Based on Fig. 4, the feasibility of the subproblem variables, including $p_k^{\mathrm{G}}$, is on the order of $10^{-10}$ in the worst case for every ADMM iteration [compare with (36)]. In other words, $(p_k^{\mathrm{G}})_{k\in\mathcal{G}}$ returned by Algorithm 2 in every ADMM iteration is feasible (with very high precision). Therefore, our proposed Algorithm 1, which includes Algorithm 2 as a subroutine, ensures the feasibility of constraints (10b)–(10f) (with very high precision). However, the feasibility of the remaining constraint (10f) has to be verified in order to compute a sensible operating point. In the sequel, we numerically analyze the feasibility of the constraint (10f) together with the objective value computed by using (10a).

Fig. 7 shows the objective value versus $\delta$, at every 100 or 200' ADMM iterations. In the case of the 3- and 9-bus examples, we compare the objective value with the branch-and-bound algorithm from [11] where the relative tolerance, the difference between the best upper and lower bounds relative to the best upper bound, is 0.001. The upper bound is obtained from Matpower, and the lower bound is obtained by using the Matlab toolbox YALMIP [36] and the solver SEDUMI [37] to solve the dual SDP relaxation. In the case of the 14- and 30-bus examples, the branch-and-bound algorithm failed due to memory errors. In all cases, we compare our results with the SDP relaxation from [3]. The results show that the algorithm converges to some objective value in relatively few iterations, which can

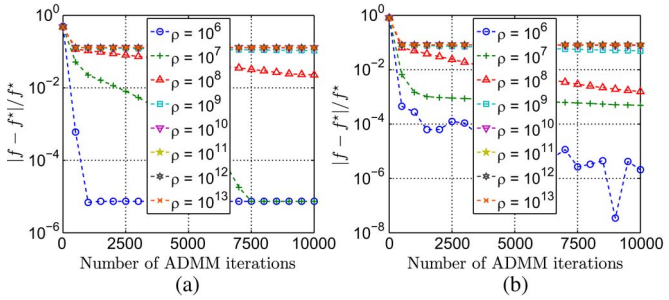


Fig. 8. Relative objective function: (a) three-bus example and (b) nine-bus example.

even be optimal with an appropriate choice of $\rho$. For example, for the considered cases, $\rho = 10^6$ yields almost optimal objective values. Moreover, the desired consistency metric $\delta$ is driven toward zero as the number of ADMM iterations increases.

Fig. 8 depicts the relative objective function ($|f - f^\star|/f^\star$) for the 3- and 9-bus examples. The results are consistent with those presented in Fig. 7. For example, in the case of $\rho = 10^6$, the relative objective function value is on the order of $10^{-6}$. The results suggest that a proper choice of $\rho$ is beneficial to achieve a good network operating point.

To study the scalability properties of the proposed algorithm, we compute the CPU time, relative objective value, $\delta$, and $\epsilon$ of the algorithm for all of the considered examples. In the case of 3-, 9-, 14-, and 30-bus examples, we choose $\rho = 10^6$ and in the case of 118- and 300-bus examples, we chose $\rho = 10^7$.

Fig. 9(a) shows the parallel running times $T_p$ versus ADMM iterations. In particular, we define $T_p = T_s/|\mathcal{N}|$, where $T_s$ is the sequential CPU time. The behavior of the plots in the case of 9-, 14-, 30-, 118-, and 300-bus examples are very similar. In other words, the parallel running time $T_p$ is independent of the number of buses, indicating the promising scalability of the proposed algorithm. Note that the 3-bus example has to handle more variables per subproblem, compared with the other examples, see column 6 of Table I. This is clearly reflected in the plot of the three-bus example, as an increase of the associated parallel running time $T_p$.

Fig. 9(b) depicts the relative objective function $|f - f^\star|/f^\star$. In the case of 3-, 9-, 118-, and 300-bus examples, the global optimum $f^\star$ is found by a branch-and-bound algorithm. However, in the case of 14- and 30-bus examples, the branch-and-bound algorithm failed and, therefore, the best known objective value found by Matpower was considered as $f^\star$.[5] The results show that for large and relatively large test examples (e.g., $N =$ 30, 118, 300), the relative objective function value is on the order of $10^{-3}$ and is not affected by network size. A similar independence of the performance is observed for very small test examples as well (e.g., $N = 3$, 9, 14) with relative objective function values on the order of $10^{-6}$. The reduction of the relative objective function values of smaller networks compared with larger network examples are intuitively expected due to substantial size differences of those networks.

[5]Since the 118- and 300-bus examples have *zero duality gap*, the branch-and-bound algorithm worked efficiently. However, in the case of 14- and 30-bus examples, where there is *nonzero duality gap*, the branch-and-bound algorithm failed.

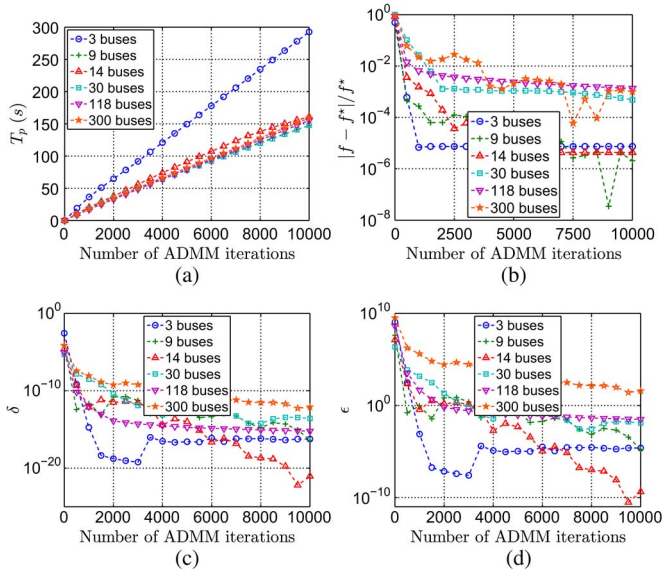


Fig. 9. $T_p$, $|f - f^\star|/f^\star$, $\delta$, and $\epsilon$ as a function of ADMM iterations. (a) $T_p$; (b) $|f - f^\star|/f^\star$; (c) $\delta$; (d) $\epsilon$.

Fig. 9(c) and (d) depicts $\epsilon$ and $\delta$ as a function of ADMM iterations. The results show that irrespective of the number of buses, the metrics $\delta$ and $\epsilon$ decrease as desired. Results further suggest that those values are driven toward small values as ADMM iterations increase.

Table III shows the running time and the objective value obtained by different approaches. As benchmarks, we consider the centralized algorithms, SDP relaxation [3], branch and bound [11], and Matpower [33]. Table III shows that our proposed method yields *network operating points*, which are almost optimal, where the discrepancy with respect to the optimal is on the order of 0.1% (respectively, 1%) or less with 10 000 (respectively, 3000) ADMM iterations. Note that the running time of ADMM-DOPF is insensitive to the network size, see Fig. 9(a) for more details. However, even in small networks (e.g., the case with $N = 14$, $N = 30$), the running time of the branch-and-bound algorithm can explode. This is expected because the worst case complexity of the branch-and-bound algorithm grows exponentially with the problem size. Results further suggest that the running time of the centralized algorithm SDP relaxation increases as the network size grows, unlike the proposed ADMM-DOPF. Note that the running time of ADMM-DOPF is large compared to the centralized Matpower. However, those values can be further reduced if ADMM-DOPF is deployed in a parallel computation environment, where every subproblem is handled at a dedicated set of resources, including processors and memory among others. In addition to the centralized benchmarks, we also consider the decentralized one proposed in [28], which employs ADMM for general nonconvex OPF. However, the results of [28] are not documented in Table III, because for all considered examples, the algorithm therein did not converge. This agrees with the numerical results of [28], where the authors mentioned that the convergence of their algorithm is more sensitive to the initial point in the case of meshed networks [28, p. 5]. We note that their method converges if it is initialized close to the optimal

TABLE III
COMPARISON OF THE OUTPUT OF ADMM-DOPF WHEN AT ADMM ITERATION $n = 3000$ AND $n = 10\,000$ WITH THE SDP RELAXATION [3], BRANCH AND BOUND [11], AND MATPOWER [33]. $T_s$, $T_p$, OBJ, AND # ITER INDICATE THE SEQUENTIAL AND PARALLEL RUNNING TIMES IN SECONDS, THE OBJECTIVE VALUE IN U.S.$/h, AND THE NUMBER OF ITERATIONS, RESPECTIVELY

| Example | SDP | | Branch & Bound | | Matpower | | ADMM-DOPF, n=3000 | | ADMM-DOPF, n=10000 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | obj | $T_s$ | obj | $T_s$ | obj | $T_s$ | obj | $T_p$ | obj | $T_p$ |
| 3 buses | 5789.9 | 5.09 | 5812.6 | 8 | 5812.6 | 0.08 | 5812.6 | 91.73 | 5812.6 | 292.54 |
| 9 buses | 6113.2 | 0.48 | 6135.2 | $1.6 \times 10^5$ | 6135.2 | 0.08 | 6135.9 | 49.19 | 6135.2 | 147.16 |
| 14 buses | 8079.6 | 0.44 | – | – | 8092.4 | 0.1 | 8092.9 | 56.12 | 8092.4 | 160.71 |
| 30 buses | 3624.0 | 0.85 | – | – | 3630.7 | 0.12 | 3634.9 | 51.53 | 3632.5 | 152.95 |
| 118 buses | 129654.1 | 10.30 | 129660.6 | $1.8 \times 10^1$ | 129660.6 | 0.2 | 130094.3 | 48.62 | 129835.2 | 155.29 |
| 300 buses | 719711.7 | 154.35 | 719725.1 | $4.6 \times 10^3$ | 719725.1 | 0.42 | 732629.1 | 51.19 | 720449.4 | 158.97 |

solution. However, in practice, such an initialization point is unknown, thus limiting dramatically the applicability of the method in [28].

Finally, from all of our numerical experiments discussed before, we note that the power losses in the flow lines are typically on the order of 4.4% (or less) of the total power flow in the line. Since the losses are not negligible, approximations, such as the linearization of power-flow equations, can be less applicable to compute better network operating points.

## VI. CONCLUSION

We proposed a distributed algorithm for the OPF, by decomposing the OPF problem among the buses that compose the electrical network. A light communication protocol among neighboring buses is needed during the algorithm, resulting in high scalability properties. The subproblems related to each bus capitalize on sequential convex approximations to gracefully manipulate the nonconvexity of the problem. We showed the convergence of subproblem solutions to the first-order necessary condition for local optimality, under mild conditions. Furthermore, by using the local optimality results associated with the subproblems, we quantified the optimality of the overall algorithm. We evaluated the proposed algorithm on a number of test examples to demonstrate its convergence properties and to compare it with the global optimal method. In all considered cases, the proposed algorithm achieved close to optimal solutions. Moreover, the proposed algorithm showed appealing scalability properties when tested on larger examples.

## APPENDIX I
## ON THE USE OF QUADRATIC PROGRAMMING QP SOLVERS

Note that not all of the constraints of problem (16) are affine (or linear). In particular, constraints (16g) and (16h) are not affine. Therefore, QP solvers are not directly applied to solve the problem. However, if constraints (16g) and (16h) are approximated by using affine constraints, then QP is readily applied to the modified problem.

Let us start by considering the feasible regions defined by (16h), which accounts for $\check{\mathcal{X}}_r^k$, $r \in \{1, \ldots, |\mathcal{N}_k|\}$, see Fig. 1(b). Next, we approximate the nonlinear boundary of $\check{\mathcal{X}}_r^k$ by affine functions as depicted in Fig. 1(c). We denote by $\check{\mathcal{Y}}_r^k$ the approximated polyhedral set. We can apply similar ideas to approximate the feasible regions specified by (16g) [cf. (6h)–(6i)], where we use $\check{\gamma}_k$ to denote the resulting affine function. Finally, the idea is to find the desired optimal solution of problem (16) by constructing a series of sets of the form $\check{\mathcal{Y}}_r^k$ and affine functions of the form $\check{\gamma}_k$ that approximate the feasible set specified by (16g) and (16h) in an *increasing precision*. The QP-based algorithm to solve problem (16) can be summarized as follows.

---

Algorithm 3: QP to solve Problem (16)

1) Initialize: Given the initial approximated set $\check{\mathcal{Y}}_r^k$ and affine function $\check{\gamma}_k$. Let $\bar{m} = 1$.
2) Solve the QP

$$\min \quad f_k^{\mathrm{G}}(p_k^{\mathrm{G}}) + \mathbf{y}_k^{\mathrm{T}}(\mathbf{v}_k - \hat{\mathbf{E}}_k \mathbf{v}) + \frac{\rho}{2}\|\mathbf{v}_k - \hat{\mathbf{E}}_k \mathbf{v}\|_2^2 \quad (38a)$$

$$\text{s.t.} \quad \mathbf{z}_k = (p_k^{\mathrm{G}}, q_k^{\mathrm{G}}, p_k, q_k, i_k^{\mathrm{Re}}, i_k^{\mathrm{Im}}, \mathbf{v}_k^{\mathrm{Re}}, \mathbf{v}_k^{\mathrm{Im}}, \quad (38b)$$

$$\hat{\mathbf{i}}_k^{\mathrm{Re}}, \hat{\mathbf{i}}_k^{\mathrm{Im}}, \hat{\mathbf{p}}_k, \hat{\mathbf{q}}_k) \quad (38c)$$

$$(\boldsymbol{\alpha}_k(\mathbf{z}_k), \hat{\boldsymbol{\lambda}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k), \hat{\boldsymbol{\mu}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k)) = \mathbf{0} \quad (38d)$$

$$(\boldsymbol{\beta}_k(\mathbf{z}_k), \check{\boldsymbol{\gamma}}_k(\mathbf{z}_k)) \leq \mathbf{0} \quad (38e)$$

$$((\mathbf{v}_k^{\mathrm{Re}})_r, (\mathbf{v}_k^{\mathrm{Im}})_r) \in \check{\mathcal{Y}}_r^k, r = 1, \ldots, |\mathcal{N}_k| \quad (38f)$$

where the variables are $p_k^{\mathrm{G}}$, $q_k^{\mathrm{G}}$, $p_k$, $q_k$, $i_k^{\mathrm{Re}}$, $i_k^{\mathrm{Im}}$, $\mathbf{v}_k^{\mathrm{Re}}$, $\mathbf{v}_k^{\mathbf{Im}}$, $\hat{\mathbf{i}}_k^{\mathrm{Re}}$, $\hat{\mathbf{i}}_k^{\mathrm{Im}}$, $b\bar{f}p_k$, $\bar{\mathbf{q}}_k$, and $\mathbf{z}_k$. The solution corresponding to the variables $\mathbf{z}_k$, $((\mathbf{v}_k^{\mathrm{Re}})_r, (\mathbf{v}_k^{\mathrm{Im}})_r)$ are denoted by $\mathbf{z}_k^{(\bar{m})}$, $\mathbf{v}_r^{(\bar{m})}$, respectively, and all of the dual optimal variables are denoted by $\mathbf{u}_k^{(\bar{m})}$.
3) Stopping criterion: If $\boldsymbol{\gamma}_k(\mathbf{z}_k^{(\bar{m})}) \leq 0$ and $\mathbf{v}_r^{(\bar{m})} \in \check{\mathcal{X}}_r^k$ for all $r \in \{1, \ldots, |\mathcal{N}_k|\}$, STOP and return $(\mathbf{z}_k^{(\bar{m})}, \mathbf{u}_k^{(\bar{m})})$. Otherwise, increase the precession of set $\check{\mathcal{Y}}_r^k$ and function $\check{\gamma}_k$ by adding a hyperplane and an affine function, respectively, set $\bar{m} := \bar{m} + 1$ and go to step 2.

---

The set $\check{\mathcal{Y}}_r^k$ is initialized in the first step by approximating the exterior boundary of the donut $\mathcal{X}_r^k$ [Fig. 1(a)] by an equilateral octagon as shown in Fig. 1(c), and $\check{\gamma}_k$ is initialized correspondingly. The second step simply involves solving a QP programming problem. The algorithm terminates in the third step if $\boldsymbol{\gamma}_k(\mathbf{z}^{(\bar{m})}) \leq 0$ and $(\mathbf{v}_k^{(\bar{m})})_r \in \check{\mathcal{X}}_r^k$ for all $r \in \{1, \ldots, |\mathcal{N}_k|\}$. However, if $(\mathbf{v}_k^{(\bar{m})})_r \in \check{\mathcal{Y}}_r^k \setminus \check{\mathcal{X}}_r^k$, we increase the precession of $\check{\mathcal{Y}}_r^k$ by adding a hyperplane on the exterior boundary of the donut $\mathcal{X}_r^k$, so that $(\mathbf{v}_k^{(\bar{m})})_r \notin \check{\mathcal{Y}}_r^k$. In particular, we set $\check{\mathcal{Y}}_r^k = \check{\mathcal{Y}}_r^k \cap \mathcal{W}$ where $\mathcal{W}$ is the halfspace

$$\mathcal{W} = \left\{((\mathbf{v}_k^{\mathrm{Re}})_r, (\mathbf{v}_k^{\mathrm{Im}})_r) \in \mathbb{R}^2 \,\middle|\, \alpha_r(\mathbf{v}_k^{\mathrm{Re}})_r + \beta_r(\mathbf{v}_k^{\mathrm{Im}})_r \leq \gamma_r\right\} \quad (39)$$

where

$$\alpha_r = \text{sign}(\text{Re}((\mathbf{v}_k^{(\bar{m})})_r)) \sqrt{\frac{(\mathbf{v}_k^{\max})_r^2}{1+((\text{Im}((\mathbf{v}_k^{(\bar{m})})_r)/(\text{Re}((\mathbf{v}_k^{(\bar{m})})_r))^2}}$$

$$\beta_r = a_r \left( \frac{(\text{Im}((\mathbf{v}_k^{(\bar{m})})_r)}{(\text{Re}((\mathbf{v}_k^{(\bar{m})})_r)} \right), \quad \gamma_r = (\mathbf{v}_k^{\max})_r^2$$

if $\text{Re}(x_r^\star) \neq 0$ and

$$\alpha_r = 0, \quad \beta_r = \text{sign}(\text{Re}((\mathbf{v}_k^{(\bar{m})})_r)), \quad \gamma_r = (\mathbf{v}_k^{\max})_r.$$

$\check{\gamma}_k$ can be treated identically.

## APPENDIX II
## PROOFS

### A. Proof of Proposition 1

*Proof:* Obviously, problem (16) is convex and in any iteration $m$ of Algorithm 2, $(\mathbf{z}_k^{(m)}, \mathbf{u}_k^{(m)})$ [so is $(\mathbf{z}_k^\star, \mathbf{u}_k^\star)$] are primal and dual optimal, with zero duality gap. Thus, $(\mathbf{z}_k^\star, \mathbf{u}_k^\star)$ satisfies KKT conditions for problem (16) [6, Sec. 5.5.3]. However, in order to show that $(\mathbf{z}_k^\star, \mathbf{u}_k^\star)$ satisfies KKT conditions for problem (12), we need to show: 1) $\mathbf{z}_k^\star$ is primal feasible; 2) $\mathbf{u}_k^\star$ is dual feasible; 3) $\mathbf{z}_k^\star$ and $\mathbf{u}_k^\star$ satisfy complementary slackness conditions; and 4) derivative of the Lagrangian of problem (12) vanishes with $\mathbf{z}_k^\star$ and $\mathbf{u}_k^\star$ [6, Sec. 5.5.3].

We start by noting that the original functions definitions $\boldsymbol{\lambda}_k(\mathbf{z}_k)$ and $\boldsymbol{\mu}_k(\mathbf{z}_k)$ [see (12d) and (12e)] are characterized by using the basic form

$$h(\underbrace{p, x_1, x_2, y_1, y_2}_{\mathbf{z}}) = p - x_1 y_1 - x_2 y_2 \tag{40}$$

where $p \in \mathbb{R}$ represents power, $x_1, x_2 \in \mathbb{R}$ represent currents, $y_1, y_2 \in \mathbb{R}$ represent voltages, and we have denoted $(p, x_1, x_2, y_1, y_2)$ compactly by $\mathbf{z}$. Let $\hat{h}^{\hat{\mathbf{z}}}$ denote the first-order Taylor's approximation of $h$ at $\hat{\mathbf{z}}$. That is, $\hat{h}^{\hat{\mathbf{z}}}$ characterizes the basic form of the first-order Taylor's approximation of function definitions $\hat{\boldsymbol{\lambda}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k)$ and $\hat{\boldsymbol{\mu}}_k^{\hat{\mathbf{z}}_k}(\mathbf{z}_k)$, see (16d) and (16e). Therefore, without loss of generality, we make our assertions based on $h$ and $\hat{h}^{\hat{\mathbf{z}}}$, together with the assumption $\lim_{m\to\infty} \mathbf{z}^{(m)} = \mathbf{z}^\star$, where $\mathbf{z}^\star$ plays the role of $\mathbf{z}_k^\star$ and $\mathbf{z}^{(m)}$ plays the role of $\mathbf{z}_k^{(m)}$.

Let us next summarize some intermediate results, which will be useful later.

*Lemma 1:* Given the function $h$ on the form (40), and $\lim_{m\to\infty} \mathbf{z}^{(m)} = \mathbf{z}^\star$, we have 1) $\lim_{m\to\infty} \hat{h}^{\mathbf{z}^{(m)}}(\mathbf{z}^\star) = h(\mathbf{z}^\star)$ and 2) $\lim_{m\to\infty} \nabla_{\mathbf{z}} \hat{h}^{\mathbf{z}^{(m)}}(\mathbf{z}^\star) = \nabla_{\mathbf{z}} h(\mathbf{z}^\star)$.

*Proof:* See Appendix B-C. ■

From Lemma 1 above, we conclude that

$$h(\mathbf{z}^\star) = \hat{h}^{\mathbf{z}^\star}(\mathbf{z}^\star) \qquad \text{and} \qquad \nabla_{\mathbf{z}} h(\mathbf{z}^\star) = \nabla_{\mathbf{z}} \hat{h}^{\mathbf{z}^\star}(\mathbf{z}^\star). \tag{41}$$

By relating the result (41) to our original problems (12) and (16), we can deduce that

$$\boldsymbol{\lambda}_k(\mathbf{z}_k^\star) = \hat{\boldsymbol{\lambda}}_k^{\mathbf{z}_k^\star}(\mathbf{z}_k^\star), \qquad \boldsymbol{\mu}_k(\mathbf{z}_k^\star) = \hat{\boldsymbol{\mu}}_k^{\mathbf{z}_k^\star}(\mathbf{z}_k^\star) \tag{42}$$

and

$$\bar{\nabla}_{\mathbf{z}_k} \boldsymbol{\lambda}_k(\mathbf{z}_k^\star) = \bar{\nabla}_{\mathbf{z}_k} \hat{\boldsymbol{\lambda}}_k^{\mathbf{z}_k^\star}(\mathbf{z}_k^\star), \bar{\nabla}_{\mathbf{z}_k} \boldsymbol{\mu}_k(\mathbf{z}_k^\star) = \bar{\nabla}_{\mathbf{z}_k} \hat{\boldsymbol{\mu}}_k^{\mathbf{z}_k^\star}(\mathbf{z}_k^\star) \tag{43}$$

where $\bar{\nabla}$ is used to represent the component-wise differentiation of associated functions.

Now we can easily conclude that $\mathbf{z}_k^\star$ is primal feasible for problem (12). This follows from (44), the fact that constraints (12b), (12c), (12f), and (12g) are identical to (16b), (16c), (16f), and (16g), respectively, and that $\check{\mathcal{X}}_r^k \subseteq \mathcal{X}_r^k$.

Dual feasibility of $\mathbf{u}_k^\star$, associated with constraints (16f) and (16g), affirms the dual feasibility of $\mathbf{u}_k^\star$ associated with identical constraints (12f) and (12g). In the case of constraint (16h), the recall from (15) that $\check{\mathcal{X}}_r^k$ is characterized by $((\mathbf{v}_k^{\text{Re}})_r, (\mathbf{v}_k^{\text{Im}})_r) \in \mathbb{R}^2$ such that

$$c_r \leq a_r (\mathbf{v}_k^{\text{Re}})_r + b_r (\mathbf{v}_k^{\text{Im}})_r \text{ and } (\mathbf{v}_k^{\text{Re}})_r^2 + (\mathbf{v}_k^{\text{Im}})_r^2 \leq (\mathbf{v}_k^{\max})_r^2. \tag{44}$$

Thus, dual feasibility of $\mathbf{u}_k^\star$ components associated with the first (respectively, second) constraint above ensures the dual feasibility of the same $\mathbf{u}_k^\star$ components associated with $(\mathbf{v}_k^{\min})_r^2 \leq (\mathbf{v}_k^{\text{Re}})_r^2 + (\mathbf{v}_k^{\text{Im}})_r^2$ (respectively, $(\mathbf{v}_k^{\text{Re}})_r^2 + (\mathbf{v}_k^{\text{Im}})_r^2 \leq (\mathbf{v}_k^{\max})_r^2$) of (12h). Thus, we conclude $\mathbf{u}_k^\star$ is dual feasible for problem (12).

From (42), the fact that constraints (12b), (12c), (12f), and (12g) are identical to (16b), (16c), (16f), and (16g), respectively, and that the components $(\mathbf{v}_k^{\text{Re}}, \mathbf{v}_k^{\text{Im}})$ of $\mathbf{z}_k^\star$, *strictly* satisfy the constraint (16h) [see Assumption 1], it follows that $\mathbf{z}_k^\star$ and $\mathbf{u}_k^\star$ satisfy complementary slackness conditions for problem (12). In addition, Assumption 1, together with the complementary slackness condition, ensures that the components of $\mathbf{u}_k^\star$ associated with constraints (16h) are *identically zero*.

Finally, recall that $(\mathbf{z}_k^\star, \mathbf{u}_k^\star)$ are optimal primal and dual variables for problem (16). Therefore, the derivative of the Lagrangian $\hat{L}_k(\mathbf{z}_k, \mathbf{u}_k)$ associated with problem (16) vanishes at $(\mathbf{z}_k^\star, \mathbf{u}_k^\star)$, that is, $\nabla_{\mathbf{z}_k} \hat{L}_k(\mathbf{z}_k^\star, \mathbf{u}_k^\star) = 0$. This result, combined with (43), and the fact that constraints (12b), (12c), (12f), and (12g) are identical to (16b), (16c), (16f), and (16g), respectively, and the fact that the components of $\mathbf{u}_k^\star$ associated with constraints (16h) are identically zero, affirms that the derivative of the Lagrangian $L_k(\mathbf{z}_k, \mathbf{u}_k)$ associated with problem (12) vanishes at $\mathbf{z}_k^\star$ and $\mathbf{u}_k^\star$, i.e.,

$$\nabla_{\mathbf{z}_k} L_k(\mathbf{z}_k^\star, \mathbf{u}_k^\star) = 0 \tag{45}$$

which concludes the proof. ■

### B. Proof of Proposition 2

*Proof:* Given that Assumption 1 holds, Proposition 1 asserts that all constraints, but (10f) of problem (10) are primal feasible. Combined with (34), it trivially follows that $\delta = a^{-1}\bar{\delta}$, where $a = \text{len}((\boldsymbol{\delta}_k)_{k\in\mathcal{N}})$ and $\bar{\delta} = \sum_{k\in\mathcal{N}} \|\boldsymbol{\delta}_k\|_2^2$ [cf. (30)]. To show that $\epsilon = b^{-1}\rho^2\bar{\delta}$ [cf. (33)], let us consider the Lagrangian $L(\mathbf{z}, \mathbf{v}, \mathbf{u}, \mathbf{y})$ associated with problem (12). Note that $L(\mathbf{z}, \mathbf{v}, \mathbf{u}, \mathbf{y})$ is related to the Lagrangian $L_k(\mathbf{z}_k, \mathbf{u}_k | \mathbf{y}_k)$ of problem (12) as

$$L(\mathbf{z}, \mathbf{v}, \mathbf{u}, \mathbf{y}) = \sum_{k\in\mathcal{N}} \Big( L_k(\mathbf{z}_k, \mathbf{u}_k | \mathbf{y}_k) - (\rho/2) \|\mathbf{x}_k - \hat{\mathbf{E}}_k \mathbf{v}\|_2^2 \Big).$$

$$\nabla_{\mathbf{z},\mathbf{v}} L(\mathbf{z}^\star,\mathbf{v}^\star,\mathbf{u}^\star,\mathbf{y}^\star) = \begin{bmatrix} \nabla_{\mathbf{z}_1} L_1(\mathbf{z}_1^\star,\mathbf{u}_1^\star|\mathbf{y}_k^\star) - \rho\bar{\mathbf{z}}_1 \\ \vdots \\ \nabla_{\mathbf{z}_N} L_N(\mathbf{z}_N^\star,\mathbf{u}_N^\star|\mathbf{y}_k^\star) - \rho\bar{\mathbf{z}}_N \\ \nabla_{\mathbf{v}} \left( \sum_{k\in\mathcal{N}} -\mathbf{y}_k^{\mathrm{T}} \hat{\mathbf{E}}_k \mathbf{v} \right) \end{bmatrix} = \begin{bmatrix} -\rho\bar{\mathbf{z}}_1 \\ \vdots \\ -\rho\bar{\mathbf{z}}_N \\ \mathbf{0} \end{bmatrix} \tag{46}$$

$$\bar{\mathbf{z}}_k = \left( 0,0,0,0,0,0, \underbrace{(\mathbf{x}_k^{\mathrm{Re}\star} - \mathbf{E}_k \mathbf{v}^{\mathrm{Re}\star}), (\mathbf{x}_k^{\mathrm{Im}\star} - \mathbf{E}_k \mathbf{v}_k^{\mathrm{Im}\star})}_{\mathbf{x}_k^\star - \bar{\mathbf{E}}_k \mathbf{v}^\star = \boldsymbol{\delta}_k}, \mathbf{0},\mathbf{0},\mathbf{0},\mathbf{0} \right) \tag{47}$$

Note the notation used when passing the parameters to $L_k$, where we have highlighted the dependence of $L_k$ on $\mathbf{y}_k$ [cf. (12a)]. Let us now inspect the derivative of the Lagrangian $L(\mathbf{z},\mathbf{v},\mathbf{u},\mathbf{y})$, evaluated at $(\mathbf{z}^\star,\mathbf{v}^\star,\mathbf{u}^\star,\mathbf{y}^\star)$. In particular, we have (46), shown at the top of the page, where $\bar{\mathbf{z}}_k$ is given in (47), shown at the top of the page [compared with (34)]. Here, the first equality follows from the standard derivation combined with Proposition 1, the second equality follows from (45), and by invoking the optimality conditions for problem (13), that is, $\sum_{k\in\mathcal{N}} \bar{\mathbf{E}}_k^{\mathrm{T}} \mathbf{y}_k^\star = 0$. From (46) and (47), we conclude that $\epsilon = b^{-1}\rho^2\bar{\delta}$ [cf. (33)], where $b = \mathrm{len}(\mathbf{z}^\star,\mathbf{v}^\star)$. Finally, conditions (28), (29), (31), and (32), associated with problem (10) follow from straightforward arguments, which concludes the proof. ■

### C. Proof of Lemma 1

Let $\mathbf{H}$ denote the Hessian of function $h$. Note that $\mathbf{H}$ is a matrix with constant entries and, thus, does not depend on $\mathbf{z}$. From the definition of the Taylor series expansion at $\mathbf{z}^m$, we have

$$h(\mathbf{z}) - \hat{h}^{\mathbf{z}^{(m)}}(\mathbf{z}) = (1/2)(\mathbf{z}^{(m)} - \mathbf{z})^{\mathrm{T}} \mathbf{H} (\mathbf{z}^{(m)} - \mathbf{z}). \tag{48}$$

Moreover, the differentiation of (48) yields

$$\nabla h(\mathbf{z}) - \nabla \hat{h}^{\mathbf{z}^{(m)}}(\mathbf{z}) = \mathbf{H}(\mathbf{z}^{(m)} - \mathbf{z}). \tag{49}$$

To show case 1 of the proposition, we consider the following relations:

$$h(\mathbf{z}^\star) - \hat{h}^{\mathbf{z}^{(m)}}(\mathbf{z}^\star) = (1/2)(\mathbf{z}^{(m)} - \mathbf{z}^\star)^{\mathrm{T}} \mathbf{H} (\mathbf{z}^{(m)} - \mathbf{z}^\star) \tag{50}$$

$$\begin{aligned} (1/2)\lambda_{\min}(\mathbf{H}) \|\mathbf{z}^{(m)} - \mathbf{z}^\star\|_2^2 &\le h(\mathbf{z}^\star) - \hat{h}^{\mathbf{z}^{(m)}}(\mathbf{z}^\star) \\ &\le (1/2)\lambda_{\max}(\mathbf{H}) \|\mathbf{z}^{(m)} - \mathbf{z}^\star\|_2^2 \end{aligned} \tag{51}$$

where (50) follows from (48) with $\mathbf{z} = \mathbf{z}^\star$, and (51) follows from (50) and basics of linear algebra. By letting $m \to \infty$ in (51), we conclude $\lim_{m\to\infty} \hat{h}^{\mathbf{z}^{(m)}}(\mathbf{z}^\star) = h(\mathbf{z}^\star)$, since $\lim_{m\to\infty} \mathbf{z}_k^{(m)} = \mathbf{z}_k^\star$. Similarly, by using (49) and that $\mathbf{H}(\mathbf{z}^{(m)} - \mathbf{z}^\star) \le \lambda_{\max}(\mathbf{H}^{\mathrm{T}}\mathbf{H}) \|\mathbf{z}^{(m)} - \mathbf{z}^\star\|_2^2$ and $\mathbf{H}(\mathbf{z}^{(m)} - \mathbf{z}^\star) \ge \lambda_{\min}(\mathbf{H}^{\mathrm{T}}\mathbf{H}) \|\mathbf{z}^{(m)} - \mathbf{z}^\star\|_2^2$, we conclude $\lim_{m\to\infty} \nabla \hat{h}^{\mathbf{z}^{(m)}}(\mathbf{z}^\star) = \nabla h(\mathbf{z}^\star)$.

## ACKNOWLEDGMENT

The authors would like to thank A. Forsgren for valuable comments.

## REFERENCES

[1] J. Carpentier, "Contribution à l'étude du dispatching économique," *Bull. Soc. Francaise Elect.*, vol. 3, no. 8, pp. 431–447, 1962.
[2] S. Frank, I. Steponavice, and S. Rebennack, "Optimal power flow: A bibliographic survey I," *Energy Syst.*, vol. 3, no. 3, pp. 221–258, Sep. 2012.
[3] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 92–107, Feb. 2012.
[4] S. Frank, I. Steponavice, and S. Rebennack, "Optimal power flow: A bibliographic survey II," *Energy Syst.*, vol. 3, no. 3, pp. 259–289, Sep. 2012.
[5] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, "Semidefinite programming for optimal power flow problems," *Int. J. Elect. Power Energy Syst.*, vol. 30, no. 6–7, pp. 383–392, Jul. 2008.
[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, USA: Cambridge University Press, 2004.
[7] S. Bose, D. Gayme, S. H. Low, and K. M. Chandy, "Optimal power flow over tree networks," presented at the 49th Annu. Allerton Conf. Commun., Control, Comput., Monticello, IL, USA, 2011.
[8] B. Zhang and D. Tse, "Geometry of injection regions of power networks," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 788–797, May 2013.
[9] J. Lavaei, D. Tse, and B. Zhang, "Geometry of power flows and optimization in distribution networks," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 572–583, Mar. 2014.
[10] B. C. Lesieutre, D. K. Molzahn, A. R. Borden, and C. L. DeMarco, "Examining the limits of the application of semidefinite programming to power flow problems," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput.*, 2011, pp. 1492–1499.
[11] A. Gopalakrishnan, A. U. Raghunathan, D. Nikovski, and L. T. Biegler, "Global optimization of optimal power flow using a branch and bound algorithm," in *Proc. 50th Annu. Allerton Conf. Commun., Control, Comput.*, 2012, pp. 609–616.
[12] E. L. Quinn, "Privacy and the new energy infrastructure," Center for Energy and Environmental Security (CEES), University of Colorado, CO 80309, USA, Tech. Rep. 09-001, Feb. 2009. [Online]. Available: http://ssrn.com/abstract=1370731 or http://dx.doi.org/10.2139/ssrn.1370731
[13] A. Cavoukian, J. Polonetsky, and C. Wolf, "Smartprivacy for the smart grid: Embedding privacy into the design of electricity conservation," *Identity Inf. Soc.*, vol. 3, no. 2, pp. 275–294, 2010.
[14] S. Salinas, M. Li, and P. Li, "Privacy-preserving energy theft detection in smart grids: A P2P computing approach," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 257–267, Sep. 2013.
[15] 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.
[16] 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, Jul. 1999.
[17] B. H. Kim and R. Baldick, "A comparison of distributed optimal power flow algorithms," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 599–604, May 2000.

[18] A. J. Conejo, F. J. Nogales, and F. J. Prieto, "A decomposition procedure based on approximate Newton directions," *Math. Program.*, vol. 93, no. 3, pp. 495–515, 2002.
[19] F. J. Nogales, F. J. Prieto, and A. J. Conejo, "A decomposition methodology applied to the multi-area optimal power flow problem," *Ann. Oper. Res.*, vol. 120, no. 1–4, pp. 99–116, 2003.
[20] G. Hug-Glanzmann and G. Andersson, "Decentralized optimal power flow control for overlapping areas in power systems," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 327–336, Feb. 2009.
[21] A. G. Bakirtzis and P. N. Biskas, "A decentralized solution to the DC-OPF of interconnected power systems," *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1007–1013, Aug. 2003.
[22] A. Y. S. Lam, B. Zhang, and D. Tse, "Distributed algorithms for optimal power flow problem," in *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, Dec. 2012, pp. 430–437.
[23] 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.
[24] B. Zhang, A. Y. S. Lam, A. Dominguez-Garcia, and D. Tse, "Optimal distributed voltage regulation in power distribution networks," *ArXiv e-prints*, Apr. 2012. [Online]. Available: http://adsabs.harvard.edu/abs/2012arXiv1204.5226Z
[25] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, "Dynamic network energy management via proximal message passing," *Found. Trends Optimiz.*, vol. 1, no. 2, pp. 73–126, 2014.
[26] S. Bolognani, G. Cavraro, R. Carli, and S. Zampieri, "A distributed feedback control strategy for optimal reactive power flow with voltage constraints," *ArXiv e-prints*, Mar. 2013.
[27] P. Šul, S. Backhaus, and M. Chertkov, "Optimal distributed control of reactive power via the alternating direction method of multipliers," *IEEE Transactions on Energy Conversion*, vol. 29, no. 4, pp. 968–977, Dec. 2014. [Online]. Available: http://dx.doi.org/10.1109/TEC.2014.2363196.
[28] A. X. Sun, D. T. Phan, and S. Ghosh, "Fully decentralized ac optimal power flow algorithms," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2013, pp. 1–5.
[29] 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. Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
[30] R. O'Neil, A. Castillo, and M. B. Cain, "The IV formulation and linear approximation of the AC optimal power flow problem," 2012. [Online]. Available: http://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-2-iv-linearization.pdf
[31] S. Boyd, "Sequential convex programming," University lecture, 2013. [Online]. Available: http://web.stanford.edu/class/ee364b/lectures/seq_slides.pdf
[32] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
[33] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
[34] R. D. Christie, *Power systems test case archive*. Seattle, Washington, USA, Aug. 1999. [Online]. Available: http://www.ee.washington.edu/research/pstca/.
[35] MATLAB, ver. 8.1.0.604 (R2013a), The MathWorks Inc., Natick, MA, USA, 2013.
[36] J. Löfberg, "Yalmip: A toolbox for modeling and optimization in MATLAB," presented at the CACSD Conf., Taipei, Taiwan, 2004. [Online]. Available: http://users.isy.liu.se/johanl/yalmip
[37] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimiz. Meth. Softw.*, vol. 11–12, pp. 625–653, 1999. [Online]. Available: http://plato.asu.edu/ftp/usrguide.pdf

**Sindri Magnússon** received the B.Sc. degree in mathematics from the University of Iceland, Reykjavík, Iceland, in 2011 and the M.Sc. degree in mathematics from KTH Royal Institute of Technology, Stockholm, Sweden, in 2013, where he is currently pursuing the Ph.D. degree in automatic control at the School of Electrical Engineering and ACCESS Linnaeus Center.

His research interests include distributed optimization, in theory and applications.

**Pradeep Chathuranga Weeraddana** (S'08–M'11) received the M.Eng. degree in telecommunication from the School of Engineering and Technology, Asian Institute of Technology, Khlong Luang, Thailand, in 2007 and the Ph.D. degree from the University of Oulu, Oulu, Finland, in 2011.

Currently, he is a Postdoctoral Researcher with the Automatic Control Lab, Electrical Engineering Department and ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include the application of optimization techniques in various application domains, such as signal processing, wireless communications, smart grids, privacy, and security.

**Carlo Fischione** (M'05) received the Dr.Eng. degree in electronic engineering (Hons.) and the Ph.D. degree in electrical and information engineering from the University of L'Aquila, L'Aquila, Italy, in 2001 and 2005, respectively.

Currently, he is a tenured Associate Professor at KTH Royal Institute of Technology, Electrical Engineering and ACCESS Linnaeus Center, Automatic Control Lab, Stockholm, Sweden. He held research positions at the University of California at Berkeley, Berkeley, CA, USA, (Visiting Scholar from 2004 to 2005 and Research Associate from 2007 to 2008) and Royal Institute of Technology, Stockholm (Research Associate from 2005 to 2007). He has co-authored more than 100 publications, including books, book chapters, international journals and conferences, and an international patent. He has also offered his advice as a Consultant to numerous technology companies, such as Berkeley Wireless Sensor Network Lab, Ericsson Research, Synopsys, and United Technology Research Center. He is co-funder and CTO of the sensor networks start-up companies Lokkupp and MIND. His research interests include optimization and parallel computation with applications to wireless-sensor networks, networked control systems, and wireless networks.

Dr. Fischione received a number of awards, including the best paper award from the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS of 2007, the best paper awards at the IEEE International Conference on Mobile Ad-hoc and Sensor System 2005 and 2009 (IEEE MASS 2005 and IEEE MASS 2009); the Best Paper Award of the IEEE Sweden VT-COM-IT Chapter of 2014; the Best Business Idea award from VentureCup East Sweden in 2010; the Ferdinando Filauro award from the University of L'Aquila, Italy, in 2003; the Higher Education award from Abruzzo Region Government, Italy, in 2004; and the Junior Research award from Swedish Research Council, 2007; and the Silver Ear of Wheat award in history from the Municipality of Tornimparte, Italy, 2012. He has chaired or served as a technical member of program committees of several international conferences and is serving as referee for technical journals. He is Ordinary Member of the academy of history Deputazione Abruzzese di Storia Patria.