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

S. Magnússon, P. C. Weeraddana, Member, IEEE, and C. 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. In this paper, an efficient novel method to address the general OPF problem is investigated. Unlike the semi-definite programming relaxation based approaches, our approach is not restricted to any special classes of electrical networks. The proposed method is based on alternating direction method of multipliers combined with sequential convex approximations. The global OPF problem is decomposed into smaller problem associated to 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—Optimal power flow, distributed optimization, smart grid.

I. INTRODUCTION

The optimal power flow (OPF) problem in electrical networks determines optimally 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.

A. Previous work

The problem was originally presented by Carpentier in the sixties [1], and has been extensively studied since then and become of great importance in efficient operation of power systems [2]. The set of feasible operation points of the OPF problem is nonconvex due to 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. One simple approximation of the OPF problem is the direct current (DC)-approximation. In this case, multiple assumptions are made to simplify the problem by linearizing the feasibility set, which results in a convex formulation. Though the resulting convex problem is solved very efficiently, the simplicity of the model can result in errors [4]. Instead of having a single crude linearized approximation, one can employ sequential convex approximation techniques, so that sophisticated convex techniques can be applied to compute efficiently a better feasible point, see [5]–[7]. Numerous other methods have been proposed for the centralized OPF problem, e.g. Newton’s method [8]–[10], interior point methods [11], [12], and stochastic methods [13]. We refer the reader to [2], [14] for a contemporary survey of OPF.

It is well known that the OPF problem can be equivalently reformulated as a rank constrained problem [15]. As a result, classic convex approximation techniques can be applied to handle nonconvexities of the rank constraint, which usually results in a semidefinite program (SDP). SDP based approaches to OPF has gained a lot of attention recently, see [3], [17]–[19] and references therein. 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 can be obtained efficiently by using convex tools. Some other classes of OPF problems, for which zero duality holds, are investigated in [17]–[19]. In particular, [17] derives zero duality results for networks that have tree topology where over satisfaction of loads is allowed. On the other hand, [18], [19] provide a graphically intuitive conditions for the zero duality gap for 2-bus networks, which are later generalized to tree topologies.

Even in the special classes of networks, where zero duality holds, the solution approaches in [3], [17]–[19] rely on that the objective is quadratic (or linear). Therefore, one of the drawback of SDP based methods is that, in the case of non-quadratic objective functions, the dual machinery employed in constructing primal feasible solutions is not applied. In addition, the applicability of SDP based approaches are limited to special network classes, and thus, in general, they fail to provide a primal feasible solution due to nonzero duality gap. Authors in [20], [21] investigate limitations of SDP based approaches and provide examples scenarios of practical interest where the sufficient conditions for zero duality does not hold. In these cases, by employing SDP, only lower bounds on the optimal value of the primal OPF problem are achieved.
The focal point of OPF study has been on centralized methods, which exhibit a poor scalability performance. 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 customers, 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 an insurance policy [24]. Therefore, 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 almost all electrical networks brings out an appealing decomposition structure, and therefore it is worth studying distributed methods for the problem.

Distributed methods for the OPF problem have been first studied in late nineties in [25]–[27], where the transmission network is divided into regions and different decomposition methods, including 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, where there cannot be border variables shared among more than 2 regions. Another approach to decentralize the problem into regions is presented in [28]–[30]. The method is based on solving the Karush-Kuhn-Tucker (KKT) optimality condition of the problem by using a modified Newton procedure where the Newton direction in each step is approximated by the local KKT conditions of all the regions. 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. Methods presented in [31] are limited to DC OPF. Recently, several decentralized methods capitalized on the SDP relaxation are designed in [32]–[34], which still have the drawbacks of being specific to special classes of networks and lack of flexibility with general objective functions.

B. Our Contributions

The main contributions of this paper are as follows:

1) We develop a distributed algorithm for the general OPF problem. We do not rely on SDP relaxation, and therefore our approach is not restricted to any special classes of networks. In addition, our method can handle non-quadratic convex objective functions unlike the SDP based distributed algorithms.

2) We capitalize on alternating direction method of multipliers (ADMM) [35] to accomplish the distributed implementation (among electrical network buses) of the proposed algorithm with a little coordination of the involved entities. In particular, the algorithm consists of 3 main steps that should be performed in an iterative manner: 1) subproblem optimization for every bus, 2) an averaging of local parameter, and 3) dual variable update. Only step 2 requires message exchanges among neighboring buses, and therefore the proposed algorithm has rich scalability properties, which is favorable in practice. Our decomposition methodology is general compared with the 2-region network decompositions adopted in [25]–[27].

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 [9] in the context of 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. In addition, the proposed algorithm is compared with an optimal solution approach based on branch and bound techniques, and MatPower [36]. Even though the optimality of the proposed method is not guaranteed due to the nonconvexity of general OPF problem, our algorithm yields optimal solutions for all the considered simulation setups. This reaffirms the promising behavior of ADMM for nonconvex problems as discussed in [35]–[9].

Thus, our solution approach for the general OPF problem is distributed, scalable, and is easily deployed with a little coordination of involved entities.

C. Organization and Notations

The 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 the paper.

The imaginary unit is denote by \( j \), i.e., \( j = \sqrt{-1} \). Boldface lower case and upper case letters represent vectors and matrices, respectively, and calligraphy letters represent sets. The cardinality of \( \mathcal{A} \) is denoted by \( |\mathcal{A}| \). len(x) denotes the length of 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 Re(z) and Im(z), respectively. The set of nonnegative integers is denoted by \( \mathbb{N} \), i.e., \( \mathbb{N} = \{0, 1, \ldots\} \). The superscript \( (\cdot)^T \) stands for transpose. We use parentheses to construct column vectors from comma separated lists, e.g., \( (a, b, c) = [a^\top \ b^\top \ c^\top]^\top \). We denote the diagonal block matrix with \( A_1, \ldots, A_N \) on the diagonal by \( \text{diag}(A_1, \ldots, A_N) \). The Hadamard product of the matrices A and B is denoted by \( A \odot B \). We denote by \( ||x||_2 \) the \( \ell_2 \)-norm of the vector x. We denote the gradient of the function \( f \) in the point x by \( \nabla_x f \).

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 \( \mathcal{L} \subseteq \mathcal{N} \times \mathcal{N} \).
the set of flow lines. Let \( i_k = v_k^i + j v_k^i \) be the current injection and \( v_k = v_k^e + j v_k^e \) be the voltage at bus \( k \in \mathcal{N} \). Let \( p_k^e + j q_k^e \in \mathbb{C} \) and \( p_k^q + j q_k^q \in \mathbb{C} \) denote the complex power demand and the complex power generated by bus \( k \in \mathcal{N} \), respectively. Thus, the complex power \( p_k + j q_k \in \mathbb{C} \) injected to bus \( k \) is given by \( p_k + j q_k = (p_k^e + j q_k^e) - (p_k^q + j q_k^q) \).

For notational compactness, we let \( p^i, q^i, p^q, q^q, i^p, i^q, i^v, v^e, v^q, v^i \), and \( v^m \) denote the vectors \((p^i_k)_{k \in \mathcal{N}}, (q^i_k)_{k \in \mathcal{N}}, (p^q_k)_{k \in \mathcal{N}}, (q^q_k)_{k \in \mathcal{N}}, (i^p_k)_{k \in \mathcal{N}}, (i^q_k)_{k \in \mathcal{N}}, (v^e_k)_{k \in \mathcal{N}}, (v^q_k)_{k \in \mathcal{N}}, (v^i_k)_{k \in \mathcal{N}}, (v^m_k)_{k \in \mathcal{N}}, \) respectively. We denote by \( p^l = y^l \in \mathbb{C} \) the complex current and by \( p_{ls} + j q_{ls} \in \mathbb{C} \) the complex power transferred from bus \( l \) to the rest of the network through the flow line \((l, s) \in \mathcal{L})

\[
Y = \begin{cases}
  yt + \sum_{(l, s) \in \mathcal{L}} y_{ls}, & \text{if } l = s, \\
  -y_{ls}, & \text{if } (l, s) \in \mathcal{L}, \\
  0, & \text{otherwise},
\end{cases}
\]

where \( y_{ls} = g_{ls} + j b_{ls} \in \mathbb{C} \) is the admittance in the flow line \((l, s) \in \mathcal{L})

\[
y_{ls} = g_{ls} + j b_{ls} \in \mathbb{C}.
\]

The admittance matrix \( Y \in \mathbb{C}^{N \times N} \) of the network is given by

\[
Y = \begin{pmatrix} y_{11} & \cdots & y_{1N} \\
          \vdots & \ddots & \vdots \\
          y_{N1} & \cdots & y_{NN} \end{pmatrix}
\]

A. Centralized formulation

For fixed power demands, \( p^d \) and \( q^d \), the goal of the OPF problem is to find the optimal way to tune the variables \( p^i, q^i, p^q, q^q, i^p, i^q, i^v, v^e, v^q, v^m \), ensuring that the fundamental relationships among power, current, and voltage parameters, imposed by laws of physics and the operational limitations imposed by the practical electrical network are held. The objective function differs between applications. In this paper we consider the minimization of a convex cost function of real power generation in the system. In particular, we denote by \( f_k^0 \) the cost of generating power at bus \( k \in \mathcal{G} \), where \( \mathcal{G} \subseteq \mathcal{N} \) denotes the set of generator buses. We can now express formally the OPF problem as

\[
\min \sum_{k \in \mathcal{G}} f_k^0(p_k^e) \quad \text{(2a)}
\]

subject to

\[
p_k^e + q_k^e = p_k^q + q_k^q, \quad k \in \mathcal{N}, \quad \text{(2b)}
\]

\[
\begin{align*}
  v_k^e &= v_k^q, \quad k \in \mathcal{N}, \\
  p_k^e + j q_k^e &= (e_k^T + j d_k^T) (v_k^e, v_k^q, v_k^i, v_k^m), \\
  \text{for } (l, s) \in \mathcal{L}.
\end{align*}
\]

\[
\begin{align*}
  p + j q &= v^m \ominus (v^e - v^q), \\
  p_{ls} + j q_{ls} &= (v_{ls}^e - v_{ls}^q), \\
  \text{for } (l, s) \in \mathcal{L}. 
\end{align*}
\]

where \( \ominus \) denotes the complex conjugate of a complex number. The constraints (2c), (2d), and (2e) are nonconvex, which in turn make problem (2) nonconvex. In fact, the problem is NP-hard. Thus, it hinders efficient algorithms for achieving optimality. However, in the sequel, we design an efficient algorithm to address problem (2) in a decentralized manner, where we first introduce new variables to unveil the rich decomposition structure of the problem and then rely on ADMM and sequential approximation techniques for the algorithm development.

B. Distributed formulation

In this section, we derive an equivalent formulation of problem (2), where all the constraints except for a single consistency constraint are decoupled among the buses. In particular, the resulting formulation is in the form of general consensus problem [35, §7.2], where fully decentralized implementation can be realized, without any coordination of a central authority. More generally, the proposed formulation can be easily adopted to accomplish decoupling among subsets of buses, each of which corresponds to buses located in a given area, e.g., multi-area OPF [25].

We start by identifying the coupling constraints of problem (2). One can easily see from constraint (2b) that the current injection of each bus is affected by the voltages of all its neighbors and by its own voltage. Therefore, constraint (2b) introduces coupling between neighbors in the network. 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 above, we first denote by \( \mathcal{N}_k \) the set of bus \( k \) itself and its neighboring buses, i.e., \( \mathcal{N}_k = \{k\} \cup \{n(k, n) \in \mathcal{L}\} \). Copies of real and imaginary parts of the voltages corresponding to buses in \( \mathcal{N}_k \) are denoted by \( x_k^e \in \mathbb{R}^{\left|\mathcal{N}_k\right|} \) and \( x_k^i \in \mathbb{R}^{\left|\mathcal{N}_k\right|} \) respectively. For notational convenience, we let \( \{x_k^e\}_1 = v_k^e \) and \( \{x_k^i\}_1 = v_k^i \). We refer to \( v^e \) and \( v^i \) as real and imaginary net variables, respectively. Note that the copies of either the net variable \( v_k^e \) or \( v_k^i \) are shared among \( |\mathcal{N}_k| \) entities, which is the degree of net variable \( v_k^e \) or \( v_k^i \). The consistency constraints are given by \( x_k = E_k v_k^e, \quad x_k = E_k v_k^i \), where \( E_k \in \mathbb{R}^{\left|\mathcal{N}_k\right| \times \left|\mathcal{N}_k\right|} \) is given by

\[
E_k = \begin{pmatrix} 1 & \text{if } (x_k^e) \text{ is a local copy of } v_k^e \\
0 & \text{otherwise}
\end{pmatrix}
\]

Note that (4) ensures the agreement of the copies of the net variables and that, for any bus \( k \), either \( x_k^e \) or \( x_k^i \) 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 \(x_k^m\) and \(x_k^n\). In particular, we can equivalently list them as follows:

\[ \begin{align*}
\phi_k^m + \phi_k^n &= g_k^m x_k^m - b_k^m x_k^n + j (b_k^n x_k^m + g_k^n x_k^n), \\
\beta_k^m + \beta_k^n &= p_k^m - \phi_k^n + j (\beta_k^n + \beta_k^m), \\
\beta_k^m + j \beta_k^n &= C_k x_k^m + D_k x_k^n + j (D_k x_k^m - C_k x_k^n), \\
\rho_k^m + j \rho_k^n &= (x_k^m)^2 + (x_k^n)^2 + j ((x_k^m)^2 - (x_k^n)^2) i_k^m, \\
\bar{p}_k + j \bar{q}_k &= (x_k^m)^2 + (x_k^n)^2 + j ((x_k^m)^2 - (x_k^n)^2) i_k^m, \\
p_k^m \leq \bar{p}_k \leq \bar{q}_k^m, \\
q_k^m \leq q_k \leq \bar{q}_k^m, \\
(\bar{p}_k)^2 + (\bar{q}_k)^2 \leq (\bar{p}_k)^2, \\
(\bar{p}_k)^2 + (\bar{q}_k)^2 \leq (\bar{p}_k)^2, \\
|\bar{p}_k|_r \leq 1, \ldots, |N_k| - 1, \\
|\bar{q}_k|_r \leq 1, \ldots, |N_k| - 1, \\
(x_k^m)^2 \leq (x_k^m)^2 + (x_k^n)^2 \leq (x_k^m)^2, \\
q_k \in \mathbb{R}(|N_k| - 1) \\
\end{align*} \]

where the variables are \(p_k, q_k, \rho_k, \beta_k, \gamma_k, \bar{p}_k, \bar{q}_k, x_k^m, x_k^n, i_k^m, i_k^n, \bar{i}_k^m, \bar{i}_k^n, \bar{p}_k, \bar{q}_k, z_k\) for \(k \in N\) and \(v^n, v^m\). Note that (10f) establishes the consistency constraints [cf. (3)], 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 [35, § 7.2], where decomposition methods can gracefully be applied.

III. DISTRIBUTED SOLUTION METHOD

In this section, we present our distributed algorithm to the OPF problem (10). In particular, we use the ADMM method as basis for our algorithm development, where we have fast convergence properties, compared to the dual decomposition [37]. The use of ADMM method is promising in the sense that it works on many nonconvex problems as a good heuristic [37, § 9]. Once the solution method is established, we investigate the properties in Section IV.

A. Outline of the algorithm

For notational simplicity, we let \(x_k\) and \(\bar{E}_k\) denote \((x_k^m, x_k^n)\) and \(\operatorname{diag}(E_k, v_k)\), respectively, for each \(k \in N\). Moreover, we let \(v\) denote \((v^n, v^m)\). 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., \(x_k = \bar{E}_k v\)) is given by

\[ L_p(x_k, y_k) = \sum_{k \in G} f_k^p(p_k^m) + \frac{\rho}{2} \|x_k - \bar{E}_k v\|_2^2 + \sum_{k \in N} y_k^m(x_k - \bar{E}_k v)^2 + \frac{\rho}{2} \|x_k - \bar{E}_k v\|_2^2, \]

where \(y_k\) is dual variable associated with (10f) and \(\rho\) is called the penalty parameter. Together with the separability of (10f) among \(k \in N\), steps of ADMM is formally expressed below.

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

1) Initialization: Set \(n = 0\) and initialize \(y_k^{(n)}\) and \(v^{(n)}\).
2) Private variable update: Set \(y_k^{(n+1)} = y_k^{(n)}\) and \(v = v^{(n)}\). Each bus \(k \in N\) updates \(x_k\) locally, where we let \((z_k^{(n+1)}, u_k^{(n+1)})\) be the primal and dual (possibly) optimal variables achieved for the following problem:

\[ \min \left\{ f_k^p(p_k^m) + y_k^m(x_k - \bar{E}_k v)^2 + \frac{\rho}{2} \|x_k - \bar{E}_k v\|_2^2 \right\}, \]

s. t.

\[ \begin{align*}
\alpha_k(z_k) &= 0, \quad \lambda_k(z_k) = 0, \\
\beta_k(z_k) &= 0, \\
\gamma_k(z_k) &= 0, \\
(x_k^m)^2 + (x_k^n)^2 &\leq (x_k^m)^2, \\
r &= 1, \ldots, |N_k|, \\
k &\in N, \\
(\bar{i}_k^m, \bar{i}_k^n, \bar{p}_k, \bar{q}_k, z_k) &\in (\bar{i}_k^m, \bar{i}_k^n, \bar{p}_k, \bar{q}_k, z_k) \quad \text{corresponding to } (x_k^m, x_k^n). \\
\end{align*} \]
3) Net variable update: We let $v^{(n+1)}$ be the solution to the problem
\[
\min \sum_{k \in \mathcal{N}} y_k^{(n)} - E_k v + \frac{\rho}{2} \|x_k^{(n+1)} - E_k v\|^2_2 \tag{13}
\]
where the variable is $v$.

4) Dual variable update: Each bus $k \in \mathcal{N}$ updates its dual variable $y_k$ as
$$y_k^{(n+1)} = y_k^{(n)} + \rho (x_k^{(n+1)} - E_k v)^{\odot} \tag{14}$$

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

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 close form solution. The net variable update can be done in a distributed fashion with a light communication protocol (see Section III-D). The forth step is the dual variable update, which can be done locally on each bus (see Section III-D). The fifth step is the stopping criterion. Natural stopping criteria include 1) running the ADMM-DOPF algorithm for a fixed number of iterations, 2) running the ADMM-DOPF algorithm till the decrement between the local- and net variables of each bus $k$ ($\|E_k v - x_k\|_2$) is below a predefined threshold 3) running the ADMM-DOPF algorithm till 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).

B. The subproblems: Private variable update

In this section, we present our method to address problem (12). Note that such a problem inherits the nonconvexity of the original problem. In fact, problem (12) also is NP-hard. Therefore, only exponentially complex global methods can guarantee the optimality of problems (12). However, for efficient implementation of the algorithm, polynomial-time algorithms are desirable, even with a loss in the optimality. Therefore, we capitalize on sequential convex approximations to design a good heuristics, which is efficient compared to global methods. Our approach is inspired from the approximations used in [5], in the context of centralized OPF.

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

In the case of (12d), we note that for any $r \in \{1, \ldots, |\mathcal{N}_r|\}$, the values of $(x_k^{(n)}_r)$ and $(x_k^{(n)}_r)$ represent a donut, see Fig. 1a.

In other words, the 2-dimensional set
\[
X_r^k = \{(x_k^{(n)}_r), (x_k^{(n)}_r) \in \mathbb{R}^2 | (x_k^{(n)}_r)^2 < (x_k^{(n)}_r)^2 \leq (x_k^{(n)}_r)^2 \}
\]

is a donut, which is clearly nonconvex.

We approximate the nonconvex set $X_r^k$ by considering a convex subset of $X_r^k$ instead, which we denote by $\hat{X}_r^k$, see Fig. 1b. To do this, we simply consider the hyperplane tangent to the inner circle of the donut at the point $C$ in Fig. 1b.

Specifically, given a point $((x_k^{(n)}_r), (x_k^{(n)}_r)) \in X_r^k$, $\hat{X}_r^k$ is the intersection of $X_r^k$ and the halfspace
\[
\{(x_k^{(n)}_r), (x_k^{(n)}_r) | a_r^k(x_k^{(n)}_r) + b_r^k(x_k^{(n)}_r) \geq c_r \}
\]

where
\[
a_r^k = \frac{\|x_k^{(n)}_r\|^2}{1 + \|x_k^{(n)}_r\|^2}, \quad b_r^k = \frac{\|x_k^{(n)}_r\|^2}{1 + \|x_k^{(n)}_r\|^2}, \quad c_r = \frac{\|x_k^{(n)}_r\|^2}{1 + \|x_k^{(n)}_r\|^2}
\]

if $(x_k^{(n)}_r)_r = 0$ and
\[
a_r^k = 0, \quad b_r^k = \frac{\|x_k^{(n)}_r\|^2}{1 + \|x_k^{(n)}_r\|^2}, \quad c_r = \frac{\|x_k^{(n)}_r\|^2}{1 + \|x_k^{(n)}_r\|^2}
\]

if $(x_k^{(n)}_r)_r = 0$. In the case of nonlinear nonconvex constraints (12d) and (12e), we capitalized on the well known Taylor’s approximation. Specifically, given a point $\tilde{z}_k$, we denote by $\hat{X}_r^k$ the first order Taylor’s approximation of $X_r^k$ at $\tilde{z}_k$. Similarly, we denote by $\hat{X}_r^k$ the first order Taylor’s approximation of $\mu_r^k$ at $\tilde{z}_k$. The approximation is refined in an iterative manner until a stopping criterion is satisfied.

It is worth noting that to construct the functions $\hat{\mu}_r^k$ and $\hat{X}_r^k$, one only needs the values of $\hat{x}_k$ where $\hat{x}$ is the component of $\tilde{z}_k$ corresponding to $(x_k^{(n)}_r, x_k^{(n)}_r)$.

By using the constraint approximations discussed above, 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 $v$ and $y_k$ from ADMM-DOPF $n$th iteration. Set $(v^{(n)}, v^{(n)}) = v$. For all $r \in \{1, \ldots, |\mathcal{N}_r|\}$, set $((x_k^{(n)}_r), (x_k^{(n)}_r)) = ((E_k v^{(n)}), (E_k v^{(n)}))$ and construct $\hat{X}_r^k$. Let $m = 1$ and initialize $\tilde{z}_k$.

2) Solve the approximated subproblem:
\[
\min f^k_\theta(p_k^0) + y_k^T(x_k - E_k v) + \frac{\rho}{2} \|x_k - E_k v\|^2_2 \tag{16a}
\]
subject to
\[
(z_k = (p_k^0, q_k^0, p_k, q_k, i_k^m, i_k^m, x_k^{(n)}_r), \hat{p}_k, \hat{v}_k, \hat{q}_k, \hat{x}_k^{(n)}_r, \hat{x}_k^{(n)}_r, \hat{y}_k^{(n)}_r, \hat{y}_k^{(n)}_r) = 0 \tag{16b}
\]
\[
\alpha_k(z_k) = 0 \quad \beta_k(z_k) = 0 \quad \gamma_k(z_k) = 0 \tag{16c}
\]
\[
\hat{\lambda}_r^k(z_k) = 0 \quad \hat{\mu}_r^k(z_k) = 0 \quad \hat{\gamma}_r(z_k) = 0 \tag{16d}
\]
\[
((x_k^{(n)}_r), (x_k^{(n)}_r)) \in \hat{X}_r^k, \quad r = 1, \ldots, |\mathcal{N}_r| \tag{16e}
\]

where the variables are $p_k^0, q_k^0, p_k, q_k, i_k^m, i_k^m, x_k^{(n)}_r, x_k^{(n)}_r, x_k^{(n)}_r$.
The unconstrained convex quadratic optimization problem (13), i.e., \( \sum_{k \in N} E_k^T Y_k = 0 \), (21) follows from that \( E_k = \text{diag}(E_k, E_k) \), and (22) follows from that \( \sum_{k \in N} E_k^T E_k = \text{diag}(1, \ldots, |N|/N) \). 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 an averaging can be accomplished by using fully distributed algorithms such as gossiping \([33]\). Therefore, step 3 of 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 increment 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**

Because the original problem \((2)\) or equivalently problem \((10)\) is nonconvex and NP-hard, optimality and convergence guarantees of non-global methods are usually difficult to achieve if not impossible \([37, \S \ 9]\). 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, i.e., step 2 of the ADMM-DOPF algorithm. Then we capitalized on one of the scenario, 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 the step 2, the main ingredient of 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 build the essential ingredient of the analysis. In particular, we consider the convex objective function \(f(p, x)\) in the place of \((12a)\). Moreover, instead of the nonconvex constraints \((12d)\) and \((12e)\) \([cf (6d), (6e)]\), we consider the constraint

\[
p = g(x)
\]

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

\[
(p, x) \in Z
\]

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

\[
\begin{align*}
\text{minimize} & \quad f(p, x) \\
\text{subject to} & \quad p = g(x) \\
& \quad (p, x) \in Z,
\end{align*}
\]

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 \([see (16d), (16e)]\) and the nonconvex constraint \((12h)\) by using a convex constraint \([see (16n)]\). Particularized to the smaller dimensional problem \((23)\), the approximations pointed above equivalent to replace \(g\) by its first order Taylor’s approximation \(\hat{g}\) and to approximate \(Z\) by some convex set \(\hat{Z}\), where \(\hat{Z} \subseteq Z\). The result is the...
approximated subproblem given by
\[
\begin{align*}
\text{minimize} & \quad f(p, x) = g(\hat{x}) + g'((x - \hat{x})) \\
\text{subject to} & \quad p = \hat{g}(x) = g(\hat{x}) + g'((x - \hat{x})) \\
& \quad (p, x) \in \hat{Z},
\end{align*}
\]
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 representational smaller dimensional problem (26). Recall that the key idea of Algorithm 2 is to iteratively refine the first order Taylor’s approximations \( \hat{g}_k(z_k) \) and \( \hat{\mu}_k(z_k) \) [see step 2.3 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 sequential refinement of \( \hat{g} \), where the shaded area represents the set \( Z \), rectangular box represents the convex set \( Z \), the solid curve represents function \( g \), dotted curves represent sequential approximations \( \hat{g} \), and thick solid curves represent the contours of \( f \). Note that there are several interesting scenarios, which deserve attention to build intuitively the behavior of Algorithm 2, see Fig. 2a, 2d. Fig. 2a shows the first scenario, where \( \text{an improper approximation of set} Z \) makes the approximated problem (26) infeasible. In contrast, Fig. 2b depicts a scenario, where \( \text{an improper choice of the approximation point} \hat{x} \) makes the approximated problem (26) infeasible. Fig. 2c shows a sequence of approximations, which eventually converges to a desired point \( A \) that would have obtained without having the first order Taylor’s approximations on \( g \). Finally, Fig. 2d shows a scenario, where a sequence of approximations switch between two points \( A \) and \( B \), i.e., there is no convergence. Any other scenario can be constructed by combining cases from Fig. 2a, 2b, and 2d.

Analogously, the discussion above suggests that the approximation points \( (\hat{z}_k)_{k \in \mathcal{N}} \) [cf. \( \hat{x} \)] used when constructing \( \hat{g}_k(z_k) \) and \( \hat{\mu}_k(z_k) \) [cf. \( \hat{Z} \)], can heavily influence the performance of Algorithm 2. Therefore, especially if the scenario 1 and 2 depicted in Fig. 2a and Fig. 2b 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 \( z_k \) and \( \lambda_k \) that can make Algorithm 2 often converge to a point as depicted in Fig. 2c and barely encounters the scenarios depicted in Fig. 2a, Fig. 2b, and Fig. 2d; see Section V-A for details.

B. Optimality properties of Algorithm 2 solution

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

**Assumption 1**: For any \( k \in \mathcal{N} \), there exists \( (z_k^*, u_k^*) \), to which Algorithm 2 can converge. Specifically, there exists \( (z_k^*, u_k^*) \), where \( \lim_{m \to \infty} (z_k^{(m)}, u_k^{(m)}) = (z_k^*, u_k^*) \) for all \( k \in \mathcal{N} \). In addition, for all \( k \in \mathcal{N} \), the components \((x_k^*, x_k^*)\) of \( z_k^* \), strictly satisfy the constraint (16).

Under Assumption 1, the following assertion can be made:

**Proposition 1**: Suppose Assumption 1 holds. Then the output \((z_k^*, u_k^*)\) of Algorithm 2 satisfy Karush-Kuhn-Tucker (KKT) conditions for problem (12).

**Proof**: See Appendix B-A.

Combined with our empirical observations that Algorithm 2 almost always converges to a point as depicted in Fig. 2c (i.e., Assumption 1 holds usually), Proposition 1 claims that that point is locally optimal or globally optimal.

C. Optimality properties of ADMM-DOPF solution

As we already pointed out, there is no guarantee that the eventual output \((z, v, u, y)\) 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 \((z_k^*, u_k^*)\) 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 properties of \((z, u)\) of ADMM-DOPF output, as we will see later. However, the properties of the remaining output \((v, y)\) is still to be investigated. In this section, combined with the results of Proposition 1, we analyze the optimality properties of 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
\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, q \\
& \quad h_i(x) = 0, \quad i = 1, \ldots, p \\
& \quad r_j(x) = 0, \quad i = 1, \ldots, s,
\end{align*}
\]
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 \( r_j : \mathbb{R}^n \to \mathbb{R} \), \( i = 1, \ldots, s \) are the equality constraint functions, and \( x \in \mathbb{R}^n \) is the optimization variable. Moreover, let \( \lambda_i \) denote the dual variable associated with constraint \( f_i(x) \leq 0 \), \( \nu_i \) and \( \omega_i \) denote the dual variable associated with constraint \( h_i(x) = 0 \) and \( r_j(x) = 0 \), respectively. Then an arbitrary...
point \((x^*, \lambda_1^*, \ldots, \lambda_q^*, \nu_1^*, \ldots, \nu_p^*, \omega_1^*, \ldots, \omega_p^*)\) is called \((\delta, \epsilon)\)-KKT optimal, if
\[
\begin{align*}
  f_i(x^*) &\leq 0, \quad i = 1, \ldots, q \\
  h_i(x^*) & = 0, \quad i = 1, \ldots, q \\
  (1/s) \sum_{i=1}^s ||r_i(x^*)||_2^2 & = \delta \\
  \lambda_i^* &\geq 0, \quad i = 1, \ldots, q \\
  \lambda_i^* f_i(x^*) & = 0, \quad i = 1, \ldots, q
\end{align*}
\]

Note that (28)-(33) are closely related to the well known KKT optimality criteria, see [39] §5.5.3. It suggest that smaller \(\delta\) and \(\epsilon\), better the point \((x^*, \lambda_1^*, \ldots, \lambda_q^*, \nu_1^*, \ldots, \nu_p^*, \omega_1^*, \ldots, \omega_p^*)\) 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 \(z = (z_k)_{k\in\mathbb{N}}\) to denote the vector of all the local primal variables in (7), \(v = (v^w, v^m)\) to denote the vector of all net variables, \(u = (u_k)_{k\in\mathbb{N}}\) to denote the dual variables associated with constraints (10b)-(10e), and finally \(y\) to denote the dual variables associated with constraint (10f).

Let us assume that at the termination of ADMM-DOPF, the output corresponding to \(v\) and \(u\) is \(v^*\) and \(u^*\), respectively. The output of ADMM-DOPF corresponding to \(z\) and \(u\) are simply the output of Algorithm 2 given by \(z^* = (z_k^*)_{k\in\mathbb{N}}\) and \(u^* = (u_k^*)_{k\in\mathbb{N}}\). However, unlike in convex problems, in the case of problem (10), one cannot take as granted that the consistency constraint (10f) is satisfied (cf. [35] §3.2.1). In particular, \(||x_k^n - E_kv^*||_2^2 \to 0\) does not necessarily hold when \(n \to \infty\), where \(k \in \mathbb{N}\) and \(n\) is the ADMM-DOPF iteration index. However, 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 §V. For latter use, let us quantify this error floor from \(\delta_k\), i.e.,
\[
\delta_k = x_k^* - E_kv^*, \quad k \in \mathbb{N}
\]

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

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

**Proof:** See appendix [B-B]

We note that deriving an analytical expression of \(\delta\) by using problem (10) data is very difficult. However, we can numerically compute \(\delta = a^{-1} \delta\), as well as \(\epsilon = b^{-1} \rho \delta\) given in Proposition 2. Extensive numerical experiences show that we usually have very small values for \(\delta\). For example, for all considered simulations with \(\rho = 10^6\) [see §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 [21], centralized OPF solver provided by Matpower [36], and the SDP relaxation from [3]. To evaluate the convergence properties of the algorithm we run it on four test example, all of which violate the sufficient conditions for zero duality provided in [3]. As a result, methods based on SDP relaxation can only provide a lower bound on these examples. The test examples are based on the 3 bus example from [20], the 9 bus example by Chow [36], and the 14 and 30 bus IEEE benchmark examples archived at [40]. The OPF problems yielding from the above setup all have zero duality gap. However, the power demands, the reactive generation limits and the apparent power flow line limit, of those examples can be altered to violate the sufficient condition, [20], [21]. In particular, this can be done by setting \(s_{\text{max}}^r = s_{\text{max}}^q = 50\) in the case of the 3 bus example, \(q_{\text{max}} = (10)_{k\in\mathbb{N}}\) and \(p^q = 1.1p^q\) in the case of the 9 bus example, \(q_{\text{max}} = (0)_{k\in\mathbb{N}}\) and \(p^q = 0.1p^q\) in the case of the 14 bus example, and \(p^q = 0.5p^q\) and \(q^q = 0.1q^q\) in the case of the 30 bus example. Moreover, to evaluate the scalability properties of the algorithm we run it on two larger examples, the 118 and 300 IEEE benchmark examples [40].

The simulations where executed in a sequential computational environment, using matlab version 8.1.0.604 (R2013a) [41]. 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 \(\epsilon = 10^{-10}\) and \(\text{max}_\text{iter} = 10\). For the ADMM method we use \(v^w = (1, \cdots, 1)\) \(v^m = (0, \cdots, 0)\) and \(y = (0, \cdots, 0)\) as an initial point.

A. Properties of Algorithm 2

In order to retain a coherent presentation we start by addressing the claims made in Sections IV-A and IV-B, where 4 scenarios, or possible outcomes, of Algorithm 2 [Fig. 2] were identified. In particular, scenarios 1 and 2 [Fig. 2a and 2b] correspond to the case where the approximated problem (26) is infeasible. In scenario 3, the sequence of approximations eventually converges to a KKT point [Fig. 2c, Proposition 2]. In contrast to scenario 4, where the sequence of approximations does not converge [Fig. 2d]. Scenario 3 can be observed from the numerical evaluations when Algorithm 2 stops after the decrement between two successive iterations is below \(\epsilon\), i.e., the stopping criteria given in equation (17) is used. On the other hand, scenario 4 can be observed when the algorithm reaches maximum number of iterations, \(\text{max}_\text{iter}\), i.e., the stopping criteria given in equation (18) is used.

During the numerical evaluations of the six test examples, which will be discussed more thoroughly in the sequence, Algorithm 2 was executed 75070000 times. The number of occurrences of each of the scenarios can be seen table in [1] where the first column indicates the scenario, the second column indicates the number of occurrences, and the third column indicates the percentage of each column. These results strongly suggest that scenario 3 is by far the most dominant.

Over the course of Algorithm 2, the power injections are subject to approximations. Therefore, to gain insight into
the convergence properties Algorithm 2 in the context of feasibility, we define the following metric associated to the degree of feasibility (DF):

$$DF_k(n) = \min_{p^+ + jq^\prime \in S_k} |p_k^{(n)} + jq_k^{(n)} - (p + jq)|$$  \hspace{1cm} (37)

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

$$S_k = \left\{ z \in \mathbb{C} \mid p_k^{\text{min}} - p_k^{\text{max}} \leq \text{Re}(z) \leq p_k^{\text{min}} - p_k^{\text{max}}, \quad q_k^{\text{min}} - q_k^{\text{max}} \leq \text{Im}(z) \leq q_k^{\text{min}} - q_k^{\text{max}} \right\}$$  \hspace{1cm} (38)

In order to provide a statistical description of $DF_k(n)$ 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 $5.7 \times 10^{-12}$.

As a consequence of this promising behavior of Algorithm 2, we will proceed under Assumption 7.

### B. Connection to Proposition 2

In this section we relate the numerical evaluations to Proposition 2. In particular, we inspect the behavior of $\delta$, and $\epsilon$ with respect to $\rho$, which are defined in Section IV-C.

Fig. 3 depicts $\delta$ at every 500th ADMM iterations, for $\rho = 10^6, \cdots, 10^{13}$. In the 30 bus example the results are almost identical for $\rho = 10^6, 10^7, \cdots, 10^{13}$ 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, i.e., $\delta < \gamma$ 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 show a promising behavior, as $\delta$ has a 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 $\rho$. Thereby, indicating 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 slower 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. 4 depicts $\epsilon$ at every 500th ADMM iterations, for different $\rho$. In contrast to $\delta$, decreasing trend in $\epsilon$ is not necessary to obtain a feasible solution to the problem. However, under Assumption 7, as $\delta$ and $\epsilon$ go to zero, the algorithm converges to 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 $\epsilon$ 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 below $10^{-2}$.

### C. Scalability properties

To study the scalability properties of the proposed algorithm, we test it on the 3, 9, 14, and 30 bus examples mentioned above and two larger examples, the IEEE 118 and 300 bus examples, for a well chosen $\rho$. In particular, we choose $\rho = 10^6$ in the case of the 3, 9, 14, and 30 bus networks, and $\rho = 10^7$ in the case the IEEE 118 and 300 bus examples. We are mainly interested in the parallel running times, denoted by $T_p$. Since the simulations are executed in a sequential computational environment, we obtain $T_p$ by the relationship $T_p = T_n/|N|$, where $T_n$ is the sequential CPU time.

Fig. 5 shows $T_p$ [Fig. 5a], $\delta$ [Fig. 5b], $|f^\star|/f^\star$ [Fig. 5c], after 500, 1000, and 1500 ADMM iterations. $|f^\star|/f^\star$ is the relative objective value related to $f^\star$, where $f^\star$ denotes the best known objective value, the global optimum given by the branch and bound method in the case of the 3, 9, 118, and
300 bus examples, and solution obtained by Matpower in the case of 14 and 30 bus networks. The results show that $T_p$ for achieving the same number of ADMM iterations is similar for the 9, 14, 30, 118, and 300 bus networks. Therefore, suggesting that $T_p$ does not depend on the number of buses. However, the 3 bus example shows worse time performance, which is not surprising since it is the only example that has flow line limits, and therefore contains greater number of variables per subproblem. Fig. 7b and 7c show that for all examples, except the 3 bus example, only a minor improvement is achieved in $\delta$ and $|f - f^*|/f^*$ after the first 500 ADMM iterations. Thereby, indicating that the ADMM iterations converge within 500 iterations to good values of $\delta$ and $|f - f^*|/f^*$ when $\rho$ is chosen appropriately. Furthermore, the achieved $\delta$ and $|f - f^*|/f^*$ are all of similar order in the case of the 30, 118, and 300 bus examples. This suggests that $\delta$ and $|f - f^*|/f^*$ do not depend on the number of buses for larger networks, indicating superior scalability properties.

VI. CONCLUSIONS
We proposed a distributed algorithm for the optimal power flow (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 a local optimum, 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 A
ON THE USE OF QUADRATIC PROGRAMMING QP SOLVERS
Note that, not all 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 $\lambda^k_r$, $r \in \{1, \ldots, |N|\}$, see Fig. 1b. Next, we approximate the nonlinear boundary of $\lambda^k_r$ by affine functions as depicted in Fig. 1c. We denote by $\gamma^k_r$ the approximated polyhedral set. We can apply similar ideas to approximate the feasible regions specified by (16g) [cf. (16a)-(16b)], where we use $\gamma^k_r$ 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 $\tilde{Y}^k_r$ and affine functions of the form $\tilde{\gamma}_k$ 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 $\tilde{Y}^k_r$ and affine function $\tilde{\gamma}_k$. Let $\tilde{m} = 1$.
2. Solve the QP:
   \[
   \begin{align*}
   \min & \quad f^k(p^r_k) + y^T(x_k - \bar{E}_k v) + \frac{\rho}{2}||x_k - \bar{E}_k v||^2_2 \\
   \text{s.t.} & \quad z_k = (p^u_k, q^u_k, q_k, \tilde{x}^u_k, \tilde{x}^m_k, x_k^m, \tilde{p}_k, \tilde{p}_k, p_k) \quad (39a)
   \end{align*}
   \]
   \[
   \begin{align*}
   \alpha_k(z_k) &= 0 & (39c) \\
   \hat{\lambda}^s_2(z_k) &= 0 & (39d) \\
   \hat{\mu}^s_2(z_k) &= 0 & (39e) \\
   \beta_k(z_k) &\leq 0 & (39f) \\
   \tilde{\gamma}_k(z_k) &\leq 0 & (39g)
   \end{align*}
   \]
   where the variables are $\tilde{x}^u_k, q^u_k, q_k, \tilde{x}^m_k, x_k^m, \tilde{p}_k, \tilde{p}_k, p_k$, and $z_k$. The solution corresponding to the variable $z_k, ((x^u_k)_r, (x^m_k)_r)$ are denoted by $z_k^{(m)}, x_r^{(m)}$ respectively and and all the dual optimal variables are denoted by $u_k^{(m)}$.
3. Stopping criterion: If $\gamma_k(z_k^{(m)}) \leq 0$ and $x_k^{(m)} \in \tilde{X}^k_r$ for all $r \in \{1, \ldots, |\mathcal{N}_k|\}$, stop and return $(z_k^{(m)}, u_k^{(m)})$. Otherwise, increase the precession of set $\tilde{Y}^k_r$ and function $\tilde{\gamma}_k$ by adding a hyperplane and an affine function, respectively, set $\tilde{m} := \tilde{m} + 1$ and go to step 2.

The set $\tilde{Y}^k_r$ is initialized in the first step by approximating the exterior boundary of the donut $\chi^k_r$ [Fig. 14] by an equilateral octagon as shown in Fig. 15 and $\tilde{\gamma}_k$ is initialized correspondingly. The second step simply involves solving a QP programming problem. The algorithm terminates in the third step if $\gamma_k(x^m_k) \leq 0$ and $x^m_k \in \tilde{X}^k_r$ for all $r \in \{1, \ldots, |\mathcal{N}_k|\}$. However, if $x^m_k \in \tilde{X}^k \setminus \chi^k_r$ we increase the precession of $\tilde{Y}^k_r$ by adding a hyperplane on the exterior boundary of the donut $\chi^k_r$, so that $x^m_k \notin \tilde{Y}^k_r$. In particular, we set $\tilde{Y}^k_r = \tilde{Y}^k_r \cap \mathcal{W}$ where $\mathcal{W}$ is the halfspace

\[
\mathcal{W} = \left\{ (x^u_k)_r, (x^m_k)_r \in \mathbb{R}^2 | \alpha_r(x^u_k)_r + \beta_r(x^m_k)_r \leq \gamma_r \right\},
\]

where

\[
\alpha_r = \text{sign}(\text{Re}(x^*_r)) \sqrt{\frac{(x^m_k)_r^2}{1 + (\text{Im}(x^*_r)/\text{Re}(x^*_r))^2}},
\]
\[
\beta_r = \alpha_r \left( \frac{(x^*_r)}{\text{Re}(x^*_r)} \right), \quad r = (x^m_k)_r,
\]

if $\text{Re}(x^*_r) \neq 0$ and $\alpha_r = 0, \quad \beta_r = \text{sign}(\text{Re}(x^*_r)), \quad r = (x^m_k)_r$.

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

**APPENDIX B**

**Proofs**

A. Proof of Proposition 1

**Proof:** Obviously, problem (16) is convex and in any iteration $m$ of Algorithm 2, $(z^m_k, u^m_k)$ [so is $(z_k^0, u_k^0)$] are primal and dual optimal, with zero duality gap. Thus, $(z^*_k, u^*_k)$ satisfies KKT conditions for problem (16) [39 § 5.5.3]. However, in order to show that $(z^*_k, u^*_k)$ satisfies KKT conditions for problem (12), we need to show 1) $z^*_k$ is primal feasible, 2) $u^*_k$ is dual feasible, 3) $z^*_k$ and $u^*_k$ satisfy complementary slackness conditions, and 4) derivative of the Lagrangian of problem (12) vanishes with $z^*_k$ and $u^*_k$ [39 § 5.5.3].

We start by noting that the original functions definitions $\lambda_k(z_k)$ and $\mu_k(z_k)$ [see (12c) and (12d)] are characterized by using the basic form

\[
h(p, x_1, x_2, y_1, y_2) = p - x_1 y_1 - x_2 y_2, \quad (41)
\]

where $p \in \mathbb{R}$ represents power, $x_1, x_2 \in \mathbb{R}$ represent current, $y_1, y_2 \in \mathbb{R}$ represent voltages, and we have denoted $(p, x_1, x_2, y_1, y_2)$ compatibly by $z$. Let $\hat{h}$ denote the first order Taylor’s approximation of $h$ at $\hat{z}$. That is, $\hat{h}$ characterizes the basic form of the first order Taylor’s approximation of function definitions $\hat{\lambda}^s_2(z_k)$ and $\hat{\mu}^s_2(z_k)$, see (16f) and (16g). Therefore, without loss of generality, we make our assertions based on $h$ and $\hat{h}$ together with the assumption

\[
\lim_{m \to \infty} z^m = z^*, \quad \text{where } z^* \text{ plays the role of } z_k^* \text{ and } z^m \text{ plays the role of } z_k^m.
\]

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

**Lemma 1:** Given the function $h$ definition of the form (41), and $\lim_{m \to \infty} z^m = z^*$, we have 1) $h(z^*) = \hat{h}(z^*)$ and 2) $\lim_{m \to \infty} \nabla h(x^m) = \nabla z h(z^*)$.

**Proof:** see Appendix B-C

From Lemma 1 above, we conclude that

\[
\nabla h(z^*) = \nabla z h(z^*) \quad \text{and} \quad \nabla h(z^*) = \nabla z h(z^*) \quad (42)
\]

By relating the result (42) to our original problems (12) and (16), we can deduce that

\[
\lambda_k(z^*_k) = \hat{\lambda}^s_2(z^*_k), \quad \mu_k(z^*_k) = \hat{\mu}^s_2(z^*_k), \quad (43)
\]

and

\[
\nabla_x \lambda_k(z^*_k) = \nabla_x \hat{\lambda}^s_2(z^*_k), \quad \nabla_x \mu_k(z^*_k) = \nabla_x \hat{\mu}^s_2(z^*_k), \quad (44)
\]

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

Now we can easily conclude that $z^*_k$ is primal feasible for problem (12). This follows from (43), the fact that constraints (12f), (12c), and (12g) are identical to (16b), (16c), (16d), and (16g), respectively, and that $\chi^k_r \subseteq \chi^k_r$. Dual feasibility of $u^*_k$ associated with constraints (16f) and (16g) affirms the dual feasibility of $u^*_k$ associated with identical constraints (12f) and (12g) in the case of constraint (16d), refer to (15) that $\tilde{\chi}^k_r$ is characterized by

\[
(x^u_k)_r, (x^m_k)_r \in \mathbb{R}^2 \text{ such that } c_r \leq \alpha_r(x^u_k)_r + b_r(x^m_k)_r \text{ and } (x^u_k)_r^2 + (x^m_k)_r^2 \leq (x^m_k)_r^2 \quad (45)
\]

Thus, dual feasibility of $u^*_k$ components associated with first (respectively second) constraint above ensures the dual feasibility of same $u^*_k$ components associated with $(x^u_k)_r^2 \leq (x^m_k)_r^2 + (x^m_k)_r^2 \leq (x^m_k)_r^2 \quad (12d)$. Thus, we conclude $u^*_k$ is dual feasible for problem (12).

From (43), the fact that constraints (12c), (12b), and (12g) are identical to (16b), (16c), (16d), and (16g), respectively, and that the components $(x^u_k, x^m_k)$ of $z^*_k$ strictly satisfy the constraint (16f) [see Assumption I], it follows that
\( \mathbf{z}_k^* \) and \( \mathbf{u}_k^* \) satisfy complementary slackness conditions for problem (12). In addition, \textit{Assumption 1} together with complementary slackness condition ensure that the components of \( \mathbf{u}_k^* \) associated with constraints (16f) are identically zero.

Finally, recall that \( \{\mathbf{z}_k^*, \mathbf{u}_k^*\} \) are optimal primal and dual variables for problem (16). Therefore, the derivative of the Lagrangian \( \mathcal{L}_k(\mathbf{z}_k, \mathbf{u}_k) \) associated with problem (16) vanishes at \( (\mathbf{z}_k^*, \mathbf{u}_k^*) \), i.e., \( \nabla_{\mathbf{z}_k} \mathcal{L}_k(\mathbf{z}_k^*, \mathbf{u}_k^*) = 0 \). This result combined with (44), the fact that constraints (12b), (12c), (12d), and (12g) are identical to (16b), (16c), (16d), and (16g), respectively, and the fact that the components of \( \mathbf{u}_k^* \) associated with constraints (16f) are identically zero, affirms that derivative of the Lagrangian \( \mathcal{L}_k(\mathbf{z}_k, \mathbf{u}_k) \) associated with problem (12) vanishes at \( \mathbf{z}_k^* \) and \( \mathbf{u}_k^* \), i.e.,
\[
\nabla_{\mathbf{z}_k} \mathcal{L}_k(\mathbf{z}_k^*, \mathbf{u}_k^*) = 0 ,
\]
which concludes the proof.

\[\square\]

\section*{B. Proof of Proposition 2}

\textbf{Proof:} Given \textit{Assumption 1} holds, \textit{Proposition 7} asserts that all constraints, but (10f) of problem (10) are primal feasible. Combined with (12a), it trivially follows that \( \delta = a^{-1} \hat{\delta} \), where \( a = \text{len}(\delta_k)_{k \in \mathcal{N}} \) and \( \hat{\delta} = \sum_{k \in \mathcal{N}} |\delta_k|/2 \) [cf (30)]. To show that \( \epsilon = (1 - b^{-1}) \rho^2 \hat{\delta} \) [cf (33)], let us consider the Lagrangian \( \mathcal{L}(\mathbf{z}, \mathbf{v}, \mathbf{u}, \mathbf{y}) \) associated with problem (12). Note that \( \mathcal{L}(\mathbf{z}, \mathbf{v}, \mathbf{u}, \mathbf{y}) \) is the Lagrangian \( \mathcal{L}_k(\mathbf{z}_k, \mathbf{u}_k|y_k) \) of problem (12) as
\[
\mathcal{L}(\mathbf{z}, \mathbf{v}, \mathbf{u}, \mathbf{y}) = \sum_{k \in \mathcal{N}} \left( \mathcal{L}_k(\mathbf{z}_k, \mathbf{u}_k|y_k) - (\rho/2) \|\mathbf{x}_k - \mathbf{E}_k \mathbf{v}\|^2/2 \right).
\]
Note the notation used when passing the parameters to \( \mathcal{L}_k \), where we have highlighted the dependence of \( \mathcal{L}_k \) on \( y_k \) [cf (12a)]. Let us now inspect the derivative of the Lagrangian \( \mathcal{L}(\mathbf{z}, \mathbf{v}, \mathbf{u}, \mathbf{y}) \), evaluated at \( (\mathbf{z}^*, \mathbf{v}^*, \mathbf{u}^*, \mathbf{y}^*) \). In particular, we have (47) [see top of this page], where \( \mathbf{z}_k \) is given in (48) [compare with (33)]. Here the first equality follows from standard derivation combined with \textit{Proposition 7} the second equality follows from (46) and by invoking the optimality conditions for problem (12), i.e.,
\[
\sum_{k \in \mathcal{N}} \mathbf{E}_k \mathbf{y}^* = 0.
\]
From (47) (48), we conclude that \( \epsilon = (1 - b^{-1}) \rho^2 \hat{\delta} \) [cf (33)], where \( b = \text{len}(\mathbf{z}^*, \mathbf{v}^*) \). Finally, conditions (28), (29), (51), and (52), associated with problem (10) follow from straightforward arguments, which concludes the proof.

\[\square\]

\section*{C. Proof of Lemma 7}

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 \( z \).

From the definition of the Taylor series expansion at \( z^m \) we have
\[
h(z) - \hat{h}(z^m)(z) = (1/2)(z^m - z)^T \mathbf{H}(z^m - z).
\]
Moreover, differentiation of (49) yields
\[
\nabla h(z) - \nabla \hat{h}(z^m)(z) = \mathbf{H}(z^m - z).
\]
To show the case 1 of the proposition, we have the following relations:
\[
h(z^*) - \hat{h}(z^m)(z^*) = (1/2)(z^m - z^*)^T \mathbf{H}(z^m - z^*),
\]
\[
(1/2) \lambda_{\min}(\mathbf{H}) \|z^m - z^*\|^2 \leq h(z^*) - \hat{h}(z^m)(z^*) \leq (1/2) \lambda_{\max}(\mathbf{H}) \|z^m - z^*\|^2.
\]
\[ \nabla z_N L(z^*, v^*, u^*, y^*) = \begin{bmatrix} \nabla z_1 L_1(z_1^*, u_1^*) - \rho z_1 \\ \vdots \\ \nabla z_N L_N(z_N^*, u_N^*, y_N^*) - \rho z_N \end{bmatrix} = \begin{bmatrix} -\rho z_1 \\ \vdots \\ -\rho z_N \end{bmatrix}, \tag{47} \]

\[ z_k = \left( 0, 0, 0, 0, 0, (x_{kN}^{v^*} - E_k v_k^{v^*}), (x_{kN}^{v^*} - E_k v_k^{v^*}) \right) \cdot \begin{bmatrix} 0, 0, 0, 0 \end{bmatrix}. \tag{48} \]

\( \rho = 10^7, \ldots, 10^{12} \), the algorithm converges to worse objective values as \( \rho \) increases. These results suggest a trade off between a good objective values, for the smaller \( \rho \)'s, and a consensus point with higher accuracy, for the larger \( \rho \)'s.

Table I assembles the results of the benchmarks together with the proposed algorithm with \( \rho = 10^6 \) for the four test examples. The first column indicates the test example, and the second, third, and forth columns indicate the the optimal values of the SDP relaxation, the branch and bound algorithm, and Matpower, respectively. Columns five, six, and seven indicate the objective value, \( \delta \), and \( \epsilon \), respectively, at the final ADMM iteration considered. From the table we observe that the 3 and 9 bus examples converge to the global optimal value given by the branch and bound algorithm. In the case of the 14 and 30 bus networks, the algorithm returns the same value as Matpower, which is very close to the theoretical lower bound given by the SDP relaxation.

### REFERENCES

1. J. Carpentier. “Contribution à l’étude du dispatching économique,” Bulletin Societe Francaise Electriciens, vol. 3, no. 8, pp. 431–447, 1962.

2. S. Frank, I. Steponavice, and S. Rebennack, “Optimal power flow: a bibliographic survey I,” Energy Systems, vol. 3, no. 3, pp. 221–258, Sept. 2012.

3. J. Lavaei and S. H. Low, “Zero duality gap in optimal power flow problem,” Power Systems, IEEE Transactions on, vol. 27, no. 1, pp. 92–107, 2012.

4. R. O’Neil, J. Jardim, and O. Alsac, “DC power flow revisited,” Power Systems, IEEE Transactions on, vol. 24, no. 3, Aug. 2009.

5. R. C. Burchett, H. H. Happ, and K. A. Wingan, “Large scale optimal power flow,” Power Apparatus and Systems, IEEE Transactions on, vol. 101, no. 10, pp. 3722 – 3732, Oct. 1982.

6. R. C. Burchett, H. H. Happ, and K. A. Wingan, “Large scale optimal power flow,” Power Apparatus and Systems, IEEE Transactions on, vol. 101, no. 10, pp. 3722 – 3732, Oct. 1982.

7. W. D. Rosehart, C. A. Cahilazare, and A. Vannelli, “Sequential methods in solving economic power problems,” in IEEE Canadian Conference on Electrical and Computer Engineering, 1997, vol. 1, pp. 281–284.

8. D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, “Optimal power flow by Newton approach,” Power Apparatus and Systems, IEEE Transactions on, vol. PAS-103, no. 10, pp. 2864–2880, 1984.

9. G. A. Maria and J. A. Findlay, “A Newton optimal power flow program for Ontario Hydro EMS,” Power Systems, IEEE Transactions on, vol. 2, no. 3, pp. 576–582, 1987.

10. S. Bruno, S. Lamonaca, G. Rotondo, U. Stecchi, and M. La Scala, “Unbalanced three-phase optimal power flow for smart grids,” Industrial Electronics, IEEE Transactions on, vol. 58, no. 10, pp. 4504–4513, 2011.

11. R. Jabr, “A primal-dual interior-point method to solve the optimal power flow dispatching problem,” Optimization and Engineering, vol. 4, no. 4, pp. 309–336, 2003.

12. H. Wang, C. E. Murillo-Sánchez, R. D. Zimmerman, and R. J. Thomas, “On computational issues of market-based optimal power flow,” Power Systems, IEEE Transactions on, vol. 22, no. 3, pp. 1185–1193, Aug. 2007.

13. D. C. Walters and G. B. Sheble, “Genetic algorithm solution of economic dispatch with valve point loading,” Power Systems, IEEE Transactions on, vol. 8, no. 3, pp. 1325–1332, 1993.

14. S. Frank, I. Steponavice, and S. Rebennack, “Optimal power flow: a bibliographic survey II,” Energy Systems, vol. 3, no. 3, pp. 259–289, Sept. 2012.

15. X. Bai, H. Wei, K. Fujisawa, and Y. Wang, “Semidefinite programming for optimal power flow problems,” International Journal of Electrical Power & Energy Systems, vol. 30, no. 6-7, pp. 383–392, July 2008.

16. S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, New York, NY, USA, 2004.

17. S. Bose, D. Gayme, S. H. Low, and K. M. Chandy, “Optimal power flow over tree networks,” in Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on, Monticello, IL, 2011.

18. B. Zhang and D. Tse, “Geometry of injection regions of power networks,” Power Systems, IEEE Transactions on, vol. 28, no. 2, pp. 788–797, 2013.

19. J. Lavaei, D. Tse, and B. Zhang, “Geometry of Power Flows and Optimization in Distribution Networks,” ArXiv e-prints, Apr. 2012. [Online]. Available: http://adsabs.harvard.edu/abs/2012arXiv1204.4419L.

20. 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 Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on, 2011, pp. 1492–1499.

21. A. Gopalakrishnan, A. U. Raghunathan, D. Nkowsko, and L. T. Biegler, “Global optimization of optimal power flow using a branch and bound algorithm,” in Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on, 2012, pp. 609–616.

22. E. L. Quinn, “Privacy and the New Energy Infrastructure,” Tech. Rep., Center for Energy and Environmental Security (CEES), University of Colorado, Feb 2009. [Online]. Available at SSRN: http://ssrn.com/abstract=1370731 or http://dx.doi.org/10.2139/ssrn.1370731.

23. A. Cavoukian, J. Polonetsky, and C. Wolf, “Smartprivacy for the smart grid: embedding privacy into the design of electricity conservation,” Identity in the Information Society, vol. 3, no. 2, pp. 275–294, 2010.

24. S. Salinas, M. Li, and P. Li, “Privacy-preserving energy theft detection in smart grids: A P2P computing approach,” Selected Areas in Communications, IEEE Journal on, vol. 31, no. 9, pp. 257–267, 2013.

25. B. H. Kim and R. Baldick, “Coarse-grained distributed optimal power flow,” Power Systems, IEEE Transactions on, vol. 12, no. 2, pp. 932–939, 1997.

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

27. B. H. Kim and R. Baldick, “A comparison of distributed optimal power flow algorithms,” Power Systems, IEEE Transactions on, vol. 15, no. 2, pp. 599–604, 2000.
