A Two-Stage Decomposition Approach for AC Optimal Power Flow
Shenyinying Tu, Andreas Wächter, Ermin Wei

Abstract—The alternating current optimal power flow (AC-OPF) problem is critical to power system operations and planning, but it is generally hard to solve due to its nonconvex and large-scale nature. This paper proposes a scalable decomposition approach to solve AC-OPF problems. The power network is decomposed into a master network and a number of subnetworks, where each network has its own AC-OPF subproblem. We formulate the problem as a two-stage optimization problem and require only a small amount of communication between the master network and subnetworks. The key contribution is a novel smoothing technique that renders the response of a subnetwork differentiable with respect to the input from the master problem, utilizing properties of the barrier problem formulation that naturally arises when the subproblem is solved by a primal-dual interior point algorithm. As a consequence, existing efficient nonlinear programming solver implementations can be used to solve both the master problem and the subproblems. The advantage of this algorithm is that speedup can be obtained by processing the subnetworks in parallel. The method is readily extended to instances with stochastic loads in the subnetworks. Numerical results show favorable performance and illustrate the scalability of the algorithm.

Index Terms—Optimal power flow, decomposition, primal-dual interior point method, two-stage optimization, smoothing technique, stochastic optimization

I. INTRODUCTION

The study of optimal power flow (OPF) [1]–[4] is fundamental in power systems, because it is an essential building block to investigate questions in operation and planning, such as state estimation [5]–[7], unit commitment [8], [9], stability and reliability assessment [10], [11], etc. It seeks to optimize some cost function, such as generation cost or transmission loss, while also satisfying the physical constraints of the power network.

One of the difficulties of solving alternating current optimal power flow (AC-OPF) problems arises from the non-convex constraint set. The direct current optimal power flow (DC-OPF) model has been widely used in practice by linearizing the AC-OPF model, which assumes the optimal solution is close to the normal operating point. However, the penetration of renewable energy introduces high fluctuations in the energy generation, which can be far away from the regular operation. Hence the DC-OPF formulation with renewable generation may lead to infeasible or suboptimal solutions. On the other hand, the growing network size makes AC-OPF problems computationally expensive to solve. To efficiently implement OPF in large-scale systems, it is beneficial to decompose the overall problem into smaller pieces, each of which can be solved independently.

Various decomposition approaches for power networks have been proposed. By applying a decomposition technique, a large network is divided into smaller subnetworks that can be solved efficiently and in parallel [12]. Decentralized OPF approaches are proposed in [13]–[17], where augmented Lagrangian methods are employed, including the auxiliary problem principle, the alternating direction multiplier method (ADMM) and the predictor-corrector proximal multiplier method. Each region solves its own OPF subproblem independently and communicates with its neighbors defined by respective partitions. However, augmented Lagrangian algorithms may fail to converge due to the lack of convexity.

Primal-dual interior point methods (PDIPM) for AC-OPF problems are studied in [18], [19]. However, these methods require the computation of a Newton step from a large linear system, which might be prohibitively expensive in large-scale OPF problems. One recent work [20] proposes a parallel PDIPM for decomposed power networks based on matrix splitting [21], in which an outer PDIPM loop and an inner matrix-splitting loop are involved. Though the paper shows that the problem can converge with a few PDIPM iterations, it still takes hundreds of matrix-splitting iterations within each PDIPM iteration.

More recent works on distributed algorithms [22]–[24] focus on relaxed models of OPF problems, where the nonconvex constraint set is relaxed by a convex outer approximation. Although these algorithms provide improvements in the computation time, feasibility is not always guaranteed.

In this paper, we propose a novel way of decomposing power networks, and solve it using two-stage optimization techniques. This approach is inspired by [25], where only provided the local convergence guarantee, we extend the approach into power flow setting. In particular, the network is partitioned into a master network and a set of subnetworks, each having its own AC-OPF problem. At each iteration of the master problem, subproblems are solved in parallel using an PDIPM, and the master network then makes decisions based on the output of subnetworks. Communication between the master network and subnetworks is only required for a small amount of variables.

In general, the optimal response of a subproblem is non-differentiable with respect to the communication variables. Our key contribution is a smoothing technique that results
in a differentiable optimal value function of a subproblem as a function of the input from master problem. This permits the use of efficient general-purpose gradient based nonlinear programming (NLP) solvers for the solution of the master problem. This is made possible by utilizing properties of the barrier problem formulation that naturally arises when the subnetworks are solved with an PDIPM.

One application of this decomposition approach is the co-optimization of a transmission network and distribution networks. In a traditional distribution system, power flow is unidirectional, and distribution systems are typically modeled as a load bus in the transmission system. However, with high penetration of renewable energy resources in distribution networks, it is also reasonable to consider each distribution system as an active power plant. In this case, the distribution system can no longer be decoupled from the transmission network, resulting in a challenging large-scale AC-OPF problem. We can decompose this problem by considering the transmission network as the master problem and distribution networks as subnetworks. The distribution networks are not necessarily tree structured networks. Our framework is applicable to the case with meshed distribution networks.

Moreover, because of the uncertainties of renewable energy in distribution systems, the data in subproblems is often stochastic and therefore requires heavy computation to solve. To account for this, the proposed framework can naturally simulate and approximate the expected cost by making many copies of a subnetwork, each one with different realizations of the random data. All of the subnetwork problems can be solved in parallel, which makes this formulation possible to solve.

This paper is organized as follows: Section II reviews the branch flow model formulation of AC-OPF problems. Section III proposes the decomposition approach of the AC-OPF problem and reformulates the problem as a two-stage optimization problem. Implementation details and numerical results are given in Section IV and Section V concludes the paper.

II. BRANCH FLOW MODEL

Consider a directed power network $G := (N, E)$, where $N$ denotes the set of buses, and $E$ denotes the set of branches. For each branch $(i, j)$ in $E$, let $y_{ij}$ denote its admittance, $z_{ij} := 1/y_{ij}$ be the corresponding impedance. Let $I_{ij}$ be the complex current, and $S_{ij} := P_{ij} + jQ_{ij}$ is the net power flow from bus $i$ to bus $j$. For each bus $j$ in $N$, let $V_j$ be the complex voltage, and $s_j$ be the power injection. Then the branch flow model is defined by [27]:

$$I_{ij} = y_{ij}(V_i - V_j), \quad \forall i \rightarrow j \in E$$

$$S_{ij} = V_jI_{ij}^H, \quad \forall i \rightarrow j \in E$$

$$s_j = \sum_{k \rightarrow j} S_{kj} - \sum_{i \rightarrow j} (S_{ij} - z_{ij}|I_{ij}|^2), \quad \forall j \in N$$

$$|V_j|^2 \leq \overline{V}_j, \quad \forall j \in N$$

$$s_j \leq |s_j| \leq \overline{s}_j, \quad \forall j \in N$$

where the complex conjugate of $I_{ij}$ is denoted by $I_{ij}^H$. At node $j$, $V_j$, $\overline{V}_j$ give the lower and upper bounds on voltage magnitude, and $\underline{s}_j$, $\overline{s}_j$ give the lower and upper bounds on power injection. Typical costs of the OPF problem include the line loss along the branches and/or power generation cost, which are usually quadratic functions of the variables. Let $C(V, S, I)$ be the cost function of the problem. Then we can formulate the AC-OPF problem as

$$\min_{V, S, I} C(V, S, I, s)$$

$$s.t. [ ]$$

The rest of the paper will focus on solving this problem. Note that our decomposition approach is not restricted to the branch flow model, and it can be applied to other power flow model representations.

III. TWO-STAGE OPTIMIZATION FRAMEWORK

A. Decomposition Scheme

This paper proposes a decomposition approach of solving the AC-OPF problem. In our decomposition scheme, a power network is decomposed into a master network and a number of independent subnetworks $\{\partial : \partial \in \mathcal{D}\}$, as sketched in Fig 1. Let $\mathcal{G} := (\tilde{N}, \tilde{E})$ be the master network, and let $G_\partial := (N_\partial, E_\partial)$ represent the subnetwork, for $\partial \in \mathcal{D}$. Motivated by the topology of transmission and distribution networks, we suppose that the network can be decomposed such that each subnetwork $G_\partial$ overlaps with the master network by exactly one bus $\partial$, denoted by $n_\partial \in N_\partial$ or $n_\partial \in \tilde{N}$ as shown in Fig 2. To avoid confusion in notation, we mark all master problem variables with a tilde (“̂”).

After decomposition, for each connecting bus, the problem has decision variables for both the master network $(\tilde{V}_{n_\partial}, \tilde{s}_{n_\partial})$ and the corresponding subnetwork $(V_{n_\partial}, s_{n_\partial})$. For the master network, let $\tilde{V}_{n_\partial}$ denote the complex voltage at $n_\partial$, and $\tilde{s}_{n_\partial}$ be the net power flow from the master network into bus $n_\partial$:

$$\tilde{s}_{n_\partial} = \sum_{n_\partial \rightarrow k \in E} \tilde{S}_{nk} - \left( \sum_{i \rightarrow n_\partial \in E} (\tilde{S}_{in_\partial} - \tilde{Z}_{in_\partial} |\tilde{I}_{in_\partial}|^2) \right).$$

For subdivnet $\partial$, let $V_{n_\partial}$ denote the complex voltage at $n_\partial$, and $s_{n_\partial}$ be the net power flow from subnetwork $\partial$ into the...
posed into a master level problem and a set of subproblems: (1), we impose boundary constraints that couple the decision variables at the connecting bus \( n_0 \):

\[
\begin{align*}
\bar{V}_{n_0} &= V_{n_0}, \quad (5a) \\
\bar{s}_{n_0} + s_{0_b} &= 0. \quad (5b)
\end{align*}
\]

Therefore, the entire AC-OPF problem can be decomposed into a master level problem and a set of subproblems:

\[
\begin{align*}
\min & \quad C(V, S, I, s) + \sum_{\delta \in \mathbb{D}} C_\delta^*(\bar{V}_{n_0}, \bar{s}_{n_0}) \\
\text{s.t.} & \quad I_{ij} = y_{ij}(\bar{V}_i - \bar{V}_j), \quad \forall i \rightarrow j \in \bar{E} \\
& \quad \bar{s}_j = \sum_{j-k} \bar{s}_{jk} - \sum_{i-j} (\bar{S}_{ij} - z_{ij}) |I_{ij}|^2, \quad \forall j \in \bar{N} \cap \{n_0\} \in \mathbb{D} \\
& \quad \bar{s}_{n_0} = \sum_{n_0 \rightarrow i,j \in E} \bar{s}_{n_0} - \sum_{i-j} (\bar{S}_{in_0} - z_{in_0}) |I_{in_0}|^2 \\
& \quad |\bar{y}_j| \leq |\bar{V}_j| \leq \bar{v}_j, \quad \forall j \in \bar{N} \\
& \quad \bar{s}_j \leq \bar{s}_j \leq \bar{s}_j, \quad \forall j \in \bar{N} \cap \{n_0\} \in \mathbb{D}
\end{align*}
\]

where \( C_\delta^*(\bar{V}_{n_0}, \bar{s}_{n_0}) \) is the optimal objective value of the following subproblem \( \delta \) given fixed values for \( \bar{V}_{n_0}, \bar{s}_{n_0} \):

\[
\begin{align*}
\min & \quad C_\delta(V, S, I, s) \quad (6a) \\
\text{s.t.} & \quad V_{n_0} = \bar{V}_{n_0}, \quad (6b) \\
& \quad s_{0_b} = -\bar{s}_{n_0}, \quad (6c) \\
& \quad I_{ij} = y_{ij}(V_i - V_j), \quad \forall i \rightarrow j \in E_0 \quad (6d) \\
& \quad S_{ij} = V_i I_j^H, \quad \forall i \rightarrow j \in E_0 \quad (6e) \\
& \quad s_j = \sum_{j-k} S_{jk} - \sum_{i-j} (S_{ij} - z_{ij}) |I_{ij}|^2, \quad \forall j \in N_0 \cap \{0_b\} \quad (6f) \\
& \quad s_{0_b} = \sum_{0_b \rightarrow i,j \in E_0} S_{0_b} - \sum_{i-j} (S_{i0_b} - z_{i0_b}) |I_{i0_b}|^2 \quad (6g) \\
& \quad r_j \leq |V_j| \leq v_j, \quad \forall j \in N_0 \cap \{0_b\} \quad (6h) \\
& \quad s_j \leq s_j \leq s_j, \quad \forall j \in N_0 \cap \{0_b\} \quad (6i)
\end{align*}
\]

For simplification, let \( x := (\bar{V}_i, \bar{s}_i : i \in \bar{N}; \bar{s}_{ij}, \bar{I}_{ij} : (i, j) \in \bar{E}) \) consists of all the variables in the master network. Let \( x_0 := (\bar{V}_{n_0}, \bar{s}_{n_0}) \) be the subset of master problem variables \( x \) that couple subnetwork \( \delta \) with the master network, and \( y_0 := (V_i, s_i : i \in N_0; I_{ij}, I_{ij} : (i, j) \in E_0) \) be the local variables in subnetwork \( \delta \). Then the above AC-OPF problem (7) can be expressed compactly as a two-stage nonlinear programming problem:

\[
\begin{align*}
\min_x & \quad C(x) + \sum_{\delta \in \mathbb{D}} C_\delta^*(x_0) \quad (7a) \\
\text{s.t.} & \quad g(x) = 0, \quad (7b) \\
& \quad h(x) \leq 0, \quad (7c)
\end{align*}
\]

where

\[
C_\delta^*(x_0) = \min_{y_0} C_\delta(y_0; x_0) \quad (8a)
\]

\[
s.t. \quad g_0(y_0; x_0) = 0, \quad (8b) \\
& \quad h_0(y_0; x_0) \leq 0. \quad (8c)
\]

Here, the constraint functions \( g, h, g_0 \) and \( h_0 \) are smooth.

Even though in the above formulation, only voltage magnitudes and power injections of the connecting bus are passed to a subnetwork, choices of communication variables can be made differently, for example, voltage phase angle and current injection. Also, the two-stage decomposition has applicability to other variations of the AC-OPF problem, including subnetworks with stochastic wind turbines and three phase phase representation.

In the proposed method, the master problem (7) is optimized with a nonlinear programming (NLP) solver. Whenever the NLP solver requires the value or derivatives of the objective function (7a), the quantities \( \{x_0\}_{\delta \in \mathbb{D}} \) corresponding to the current iterate are passed to the subproblems (8). Their optimal solutions are computed, and the optimal objective values \( C_\delta^*(x_0) \), together with the first and second order derivatives \( \frac{\partial C_\delta^*}{\partial x} \) and \( \frac{\partial^2 C_\delta^*}{\partial x^2} \), are passed back to the NLP solver, which continues to solve until a minimizer is found. The information exchange is illustrated in Fig 3. Since only few variables are communicated within the network, this approach can be efficiently implemented in a distributed setting. Note that this algorithm is ill-defined if any of the subproblems becomes infeasible given \( x_0 \) at one of the master problem iterates. For now, we assume subproblems are always feasible for any \( x_0 \). We will discuss in section III-D how this assumption can be lifted.

### B. Subproblems

1) **Smoothing of Subproblems:** The description above ignores the crucial fact that, in general, the optimal value functions \( C_\delta^* \) may not be differentiable in \( x_0 \). More specifically, whenever the set of inequality constraints (8c) that are tight at the optimal solution changes with \( x_0 \), the function \( C_\delta^* \) typically is non-differentiable and experiences abrupt changes in first derivatives around those positions. This may result in convergence failures of the master problem NLP solver.
As a remedy, we replace (8) by its barrier problem formulation:

\[ C^b_0(x_0, \bar{\mu}) = \min_{y_0} C_0(y_0; x_0) - \bar{\mu} \sum_i \ln(s_i) \quad \text{(9a)} \]

\[ s.t. \quad h_0(y_0; x_0) + s = 0. \quad \text{(9b)} \]

Here, the inequality constraint (8c) has been replaced by an equality constraint that introduces slack variables \( s \). To simplify notation, we drop the equality constraints (8c) for the remainder of this section). The objective function (9a) now includes a logarithmic barrier term with weight \( \bar{\mu} > 0 \) which keeps the slack variables strictly positive. Interior point methods are based on this formulation and obtain a primal-dual solution for (9). These methods work with the computed master problem, parameter \( \bar{\mu} \) which can be optimized with standard NLP solvers. Section III-B3 describes how the derivatives of \( C^b_0(x_0, \bar{\mu}) \) depend smoothly on \( x_0 \).

Algorithm 1: Generic PDIPM Framework

Input: \( x_0 \), initial iterate \((y_0^{(0)}, s_0^{(0)}, \lambda_0^{(0)})\), initial barrier parameter \( \mu^{(0)} \).

1: Set \( k \leftarrow 0 \).

2: repeat

3: \( \text{while } \| F(y_0^{(k)}, s_0^{(k)}, \lambda_0^{(k)}; x_0, \mu^{(k)}) \| > \epsilon_F^{(k)} \) do

4: \( \quad \text{Compute Newton step } (\Delta y_0^{(k)}, \Delta s_0^{(k)}, \Delta \lambda_0^{(k)}). \)

5: \( \quad \text{Perform line search to compute step size } \alpha^{(k)}. \)

6: \( \quad \text{Update iterate } (y_0^{(k+1)}, s_0^{(k+1)}, \lambda_0^{(k+1)}) = (y_0^{(k)}, s_0^{(k)}, \lambda_0^{(k)}) + \alpha^{(k)} (\Delta y_0^{(k)}, \Delta s_0^{(k)}, \Delta \lambda_0^{(k)}). \)

7: \( \quad \text{Increase iteration counter } k \leftarrow k + 1. \)

8: \( \text{end while} \)

9: \( \quad \text{Decrease } \mu^{(k)} = \min(0.1 \mu^{(k)}, (\mu^{(k)})^{1.5}). \)

10: until \( \mu^{(k)} \leq \epsilon_\mu. \)

11: return \( (y_0^{(k)}, s_0^{(k)}, \lambda_0^{(k)}). \)

Step 4 requires the computation of the Newton step, which is computed as the solution of the linear system

\[ J(y_0^{(k)}, s_0^{(k)}, \lambda_0^{(k)}; x_0, \mu^{(k)}) \left[ \begin{array}{c} \Delta y_0^{(k)} \\ \Delta s_0^{(k)} \\ \Delta \lambda_0^{(k)} \end{array} \right] = -F(y_0^{(k)}, s_0^{(k)}, \lambda_0^{(k)}; x_0, \mu^{(k)}) \]

where

\[ J = \begin{bmatrix} V_{y_0}^T C_0 - \sum_i V_{y_0}^T h_0 \lambda_i & 0 & -V_{y_0} h^T \\ V_{y_0} h & I & 0 \\ 0 & \text{diag}(\lambda_0) & \text{diag}(s_0) \end{bmatrix} \]

is the Jacobian of \( F \). Here we drop the function arguments for brevity.

Algorithm 1 shows the steps of a generic line search PDIPM. Practical methods are more involved [28], [29], but this description highlights the features that are relevant in our context. In the while loop, \( \epsilon_F^{(k)} \) is the tolerance to which the barrier problem for the current value of \( \mu^{(k)} \) is solved, and \( \epsilon_\mu \) is the overall convergence tolerance. In a regular setting, \( \epsilon_\mu \) is set to a tight tolerance \( \epsilon \) (e.g., 10\(^{-5}\)) and \( \epsilon_F^{(k)} = 0.1 \mu^{(k)} \) [28].

As we discussed in Section III-B1, the smoothed master problem (10) requires the solution of the barrier problem (9) for a fixed given value \( \bar{\mu} > 0 \) of the barrier parameter. This can be computed with a simple modification of Algorithm 1. Instead of decreasing the barrier parameter \( \mu^{(k)} \) all the way to zero, the algorithm eventually fixes it to \( \bar{\mu} \) and from then on sets \( \epsilon_F^{(k)} \) to the overall tight convergence tolerance \( \epsilon \). The solution returned is then an optimal primal-dual solution for (9).

Here, \( \lambda \) is the dual variable corresponding to the constraint (9b), \( s \circ \lambda \) is the component-wise product of two vectors \( s \) and \( \lambda \), and the notation \( 1 \) defines a column vector with entries 1. We can use any general purpose PDIPM solver, which on a high level finds a root of \( F(\cdot; x_0, \mu) \) for a fixed value of \( \mu \) by applying Newton’s method to (11), and decreases \( \mu \) to zero in order to converge to a local optimum of the original problem (8).
We emphasize that this approach requires only a small modification of the termination criteria of a PDIPM. This has the significant practical advantage that powerful and efficient implementations of PDIPM such as Ipopt [28] and Knitro [29] could easily be adapted and utilized for the solution of the AC-OPF problem [9]. The next section shows that also the computation of derivatives of $C_o^*(x_d, \bar{\mu})$ can exploit existing features of a PDIPM implementation.

3) Derivative Computations: Recall that $J$ is the Jacobian of primal-dual optimality conditions of problem [9] for any given $x_o$ and $\bar{\mu}$. We assume that some standard second-order optimality conditions, which typically hold for non-degenerate AC-OPF problems [30], are satisfied for [9] so that $J$ is nonsingular. By the implicit function theorem, there exists a set of unique differentiable functions $(y_0^*(x_o), s_o^*(x_o), \lambda_1^+(x_o))$ in the neighborhood of $x_o$, where $y_0^*(x_o), s_o^*(x_o), \lambda_1^+(x_o)$ satisfy the primal-dual optimality condition [11]. Moreover, we have

$$\frac{\partial y_0^*}{\partial x_o} = -J^{-1} \frac{\partial F}{\partial x_o}. \tag{14}$$

Since $C_o^*(x_o, \bar{\mu}) = C_o(y_0^*(x_o); x_o, \bar{\mu})$, we obtain

$$\frac{\partial C_o^*}{\partial x_o} = \frac{\partial C_o^T}{\partial y_0^*}, \tag{15}$$

where $\frac{\partial C_o^T}{\partial y_0^*}$ is the derivative of the subproblem cost function with respect to the local variable $y_0^*$.

We can also derive the Hessian of $C_o^*(x_o; \bar{\mu})$:

$$\frac{\partial^2 C_o^*}{\partial x_o^2} = \frac{\partial^2 y_0^*}{\partial x_o^2} \frac{\partial C_o}{\partial y_0^*} + \frac{\partial y_0^*}{\partial x_o} \frac{\partial^2 C_o}{\partial y_0^*},$$

$$= (-J^{-1} \frac{\partial F}{\partial x_o} - J^{-1} \frac{\partial F}{\partial x_o} \frac{\partial F}{\partial y_0^*}) \frac{\partial C_o}{\partial y_0^*} + \frac{\partial y_0^*}{\partial x_o} \frac{\partial^2 C_0}{\partial y_0^*}, \tag{16}$$

The third step follows from the second step due to the identity $\frac{\partial y_0^*}{\partial x_o} = -J^{-1} \frac{\partial F}{\partial x_o}$. The term $J^{-1} \frac{\partial F}{\partial y_0^*}$ is eliminated because $\frac{\partial F}{\partial y_0^*} = 0$, since the variables in $x_o$, $V_{s_o}$ and $S_{s_o}$ appear only linearly in $y_0^*$.

An important observation is that the matrix $J$ in [15] and [16] is the same as the one used to compute Newton steps in the PDIPM. As a consequence, one can re-use the efficient implementation for solving [12] that is already available in the PDIPM code, similar to the approach described in [31].

C. Two-Stage Algorithm

The overall solution procedure is described in Algorithm 2. It consists of solving a sequence of master problems [10] where the smoothing parameter $\bar{\mu}$ is driven to zero. An NLP method solves each of the master problems. Whenever the NLP solver requires the value or derivatives of $C_o^*(x_o, \bar{\mu})$, the values of $x_o$ corresponding to the current iterate are sent to the subproblems. Each subproblem is then solved by a modified PDIPM (see section III-B2), and the derivatives are computed as described in section III-B3. These quantities are then sent back to the master problem NLP solver which continues its execution.

The optimal solutions of two consecutive master problems in Algorithm 2 can be expected to be close to each other, particularly when $\bar{\mu}$ is small. Therefore, to aid convergence, the solution of a master problem is provided to the NLP solver as starting point for the solution of the next master problem after $\bar{\mu}$ has been decreased. To best exploit this information, we prefer an active-set type solver, such as a sequential quadratic programming (SQP) method [32], over an interior point method, since the latter is known to have inferior warm-starting capabilities [33]. Particularly in the final stages of Algorithm 2 when $\bar{\mu}$ changes only by a very small amount, we ideally want to encounter only very few iterations of the master problem solver. One of the conclusions of our numerical experiments is that this is indeed possible.

D. Infeasible Subproblems

One difficulty of our two-stage decomposition approach is that subproblem [9] may become infeasible for a given $x_o$ during some master problem iteration. As a remedy, using the notation from [9], we let $y_0 = (V_{s_o}, s_{s_o})$ be a subvector of $y_0$ and introduce slack variables for the constraints [65]–[65] that force $y_0$ to take the values $x_o$ prescribed by the master problem:

$$C_o^*(x_o) = \min_{y_0, t} C_0(y_0; x_o) + \eta e^T(r + t) \tag{17a}$$

s.t. $f_0 - x_o = r - t,$ $g_0(y_0; x_o) = 0,$ $h_0(y_0; x_o) \leq 0,$ $r, t \geq 0.$ \tag{17b}\tag{17c}\tag{17d}\tag{17e}$$

Here, $e$ is the vector of all ones with appropriate dimension, and $\eta > 0$ is a fixed parameter. If the subproblem is feasible when $y_0$ is not restricted, this problem is always feasible. Clearly, at a (locally) optimal solution of [17], the new term...
in the objective measures the $\ell_1$-norm of the violation of the coupling constraints \((6b) - (6c)\). This is the standard $\ell_1$-norm penalty function formulation of a nonlinear optimization problem \([34]\). Penalty functions have been used in the past in two-stage decomposition approaches \([35] - [37]\). We choose the $\ell_1$-penalty function because it is "exact" in the sense that the optimal solution of \((17)\) satisfies the original constraints \((6b) - (6c)\) if the undecomposed problem is feasible and the penalty parameter $\eta$ is sufficiently large but finite \([34]\). The smoothing technique described in section \(\text{III-B}\) is then applied to \((17)\) instead of \((9)\).

For the purpose of this paper we assume that a sufficiently large value for $\eta$ has been chosen. A more comprehensive approach would include mechanisms that update the penalty parameter if it is too small \([38]\).

E. Convergence Guarantees

The overall Algorithm \(2\) consists of three nested loops:

1) For given values of $x_0$ and $\tilde{\mu}$, convergence results for an appropriately chosen PDIPM method guarantee convergence of the algorithm solving the barrier problem \((9)\) to a local optimum, under standard assumptions that are typically satisfied by AC-OPF problems. Recall that \((9)\) must be feasible due to the $\ell_1$-norm penalty formulation described in section \(\text{III-D}\) if we assume that the undecomposed problems is feasible.

If the PDIPM always converges to a unique global minimum of \((9)\), then $C_\lambda^*(x_0, \tilde{\mu})$ is uniquely defined and, as discussed in section \(\text{III-B}\), differentiable. This assumption is reasonable since it has been observed that, in practice, a PDIPM applied to an AC-OPF problem usually converges to the global minimizer \([39]\).

2) For a fixed value of $\tilde{\mu}$, the modified master problem \((10)\) is a nonlinear optimization problem with differentiable problem functions, and an appropriately chosen NLP solver, such as an SQP method, will converge to a local optimum (if the original two-stage problem is feasible).

3) Finally, to understand the convergence of the overall Algorithm \(2\), we cite a result from \([35]\) that discusses the existence of local solutions of the two-stage problem:

**Theorem 3.1:** Let $(x^\ast, (x_0^\ast)_{\varnothing \in \mathcal{D}}, (y_0^\ast)_{\varnothing \in \mathcal{D}})$ be a minimizer of the undecomposed problem satisfying the nondegeneracy conditions in \([35]\). Then there exists a locally unique trajectory $(x^\ast(\tilde{\mu}), x_0^\ast(\tilde{\mu}))$ of minimizers to \((10)\), such that $\lim_{\tilde{\mu} \to 0} x^\ast(\tilde{\mu}) = x^\ast$, $\lim_{\tilde{\mu} \to 0} x_0^\ast(\tilde{\mu}) = x_0^\ast$.

Therefore, Algorithm \(2\) converges to a local minimizer $(x^\ast, (x_0^\ast)_{\varnothing \in \mathcal{D}})$ if the master problem solver eventually returns optimal solutions corresponding to its local trajectory. Again, since local solvers typically find the global solutions of AC-OPF problems, this is a reasonable assumption.

IV. NUMERICAL EXAMPLE

A. Implementation Details

The SQP algorithm in the optimization package KNITRO \([40]\) is employed in MATLAB to solve the master problem. When $\tilde{\mu}$ decreases, we initialize the master problem with the optimal solution from last $\tilde{\mu}$. To solve the subproblems, we implemented a basic version of the PDIPM Algorithm \(1\) given in Section \(\text{III-B}\). Warm-start initialization is also used for the solution of subproblems \((9)\), where at each NLP iteration, we initialize the PDIPM with the primal-dual optimal solutions from the last NLP iteration. Experiments show that given warm-start initialization, the PDIPM method always converges within 10 iterations. The penalty parameter was chosen as $\eta = 9$.

B. Numerical Results

This section describes numerical experiments obtained with a combined transmission and distribution network model. Since the focus of this paper is the decomposition algorithm, the distribution model is simplified as a single-phase model. Our approach can be extended to three-phase distribution system in a straightforward manner, by changing the communication variable $x_0$ and the subproblem formulation.

The test instance consists of an IEEE-24-RTS system \([41]\) as master network and two modified IEEE 33-bus systems as subnetwork. Two 33-bus systems were taken from \([42]\), as shown in Figure 4, and the data are scaled to match the loads in the master system. To model active distribution systems, two generators are added into each of the distribution systems. For the distribution system 1, the generators are placed at buses 17, 32 relative to the positions in system 1. For the distribution system 2, the generators are placed at buses 6, 23 relative to the positions in system 2. The two distribution networks are connected at buses 21 and 23 of the master network. The test data of the combined system are provided in \([43]\). The solutions of our decomposition algorithm on this test data recover the solutions computed by MATPOWER \([44]\) for the undecomposed system.

To account for uncertainties of renewable energy in distribution system, we allow $\pm 10\%$ fluctuations in subnetwork loads. For each subnetwork $\varnothing \in \mathcal{D} = \{1, 2\}$, we generated $N$ scenarios. More specifically, for scenario $i$, we chose fluctuations $\xi_{\varnothing, i}$ randomly from a uniform distribution. The goal is the minimization of the expected total generation cost, approximated by sample average approximation. In

\(\text{Figure 4: IEEE 33-bus distribution network. The two copies are connected to the master network at buses 21 and 23, respectively.}\)
Figure 5: Computation time as the number of scenarios is increased.

this case, the objective function of the two-stage problem (10) becomes

\[ C(x) + \sum_{\delta \in \mathcal{D}} \frac{1}{N} \sum_{i=1}^{N} C_{\delta_i}^*(x_\delta, \tilde{\mu}_i, \xi_\delta). \]  (18)

Now we have \(|\mathcal{D}| \times N\) many subproblems, each one corresponding to a particular scenario for a given subnetwork.

To explore the scalability of our decomposition approach, we ran our implementation of Algorithm 2 for range of numbers of scenarios \(N\), from 1 to 500, and measured the wall clock time required to converge. For each \(N\), this experiment was repeated 10 times for different realizations of load fluctuations \(\xi\). We plot the averaged computation time in Figure 5. We observe that the computation time is roughly linear in the number of scenarios, which indicates that our approach seems to scale well as the number of subproblems increases.

Moreover, as the number of scenarios grows, the number of master problem iterations remains constant. In our implementation, we set the initial value of \(\tilde{\mu}\) to be \(10^{-2}\), and then sequentially decrease its value for the \(\mu\)-update in Step 3 of Algorithm 2. Consequently, the master problem is solved for the values \(\tilde{\mu}_1 = 10^{-2}\), \(\tilde{\mu}_2 = 10^{-3}\) and \(\tilde{\mu}_3 = 10^{-6}\), where the last value corresponds to the final tolerance. For each \(\tilde{\mu}_i\), we observe that the master problem SQP solver (10) converges locally at a superlinear rate because second-order derivatives are provided [34]. Figure 6 shows the number of SQP iterations for each of the 10 trials with the different numbers of scenarios. In particular, for the final smoothing parameter \(\tilde{\mu}_3\), the master problem converges within few iterations, with an average of 1.84 iteration. This is made possible by the warm start initialization described in Section III-C.

Finally, the decomposition scheme permits the parallel evaluation of the subproblems. To explore the resulting computational speedup, we generated \(N = 1000\) scenarios for each subnetwork. Figure 7 plots the wall clock time against the number of cores we used. We can see that the computation time is significantly improved by solving subproblems in parallel.

V. CONCLUSION

This paper proposes a novel two-stage optimization algorithm that partitions a power network into a master network and a set of subnetworks. Stochastic instances can be handled by replicating subnetworks with different realizations of the uncertain parameters.

By introducing a barrier term as a smoothing technique for the subproblems, the subnetwork response becomes differentiable with respect to the master problem variables. As a consequence, efficient existing nonlinear optimization algorithms with fast local convergence properties can be utilized for the master problem.

The AC-OPF problems of subnetworks can be solved with primal-dual interior point methods which exhibit fast local convergence guarantees as well. Existing algorithm implementations can be used after minor modifications of their termination criteria. First- and second-order derivatives of the subproblem response with respect to the master
problem variables can be derived via the implicit function theorem and computed efficiently using the Jacobian matrix of the primal-dual optimality conditions, which is already constructed within the interior point solver.

Preliminary experiments show that the approach scales linearly in the number of subnetworks, and that the exploitation of warm-start capabilities of an active-set SQP solver significantly accelerates the solution of subsequent master problems in which the value of the smoothing parameter is decreased. The approach is naturally able to exploit parallel computing resources by handling subproblems in parallel, where only a small amount of communication is required between the master problem and subproblems.

REFERENCES

[1] M. Huneault and F. Galiana, “A survey of the optimal power flow literature,” IEEE transactions on Power Systems, vol. 6, no. 2, pp. 762–770, 1991.

[2] S. Frank, I. Steponavice, and S. Rebennack, “Optimal power flow: a bibliographic survey, part i and ii,” Energy Systems, vol. 3, no. 3, pp. 221–289, 2012.

[3] F. Capitanescu, J. M. Ramos, P. Panciatici, D. K. Molzahn, M. A. Marcolini, L. Platbrood, and L. Wehenkel, “State-of-the-art, challenges, and future trends in security constrained optimal power flow,” Electric Power Systems Research, vol. 81, no. 8, pp. 1731–1741, 2011.

[4] D. K. Molzahn, L. T. Hiskens et al., “A survey of relaxations and approximations of the power flow equations,” Foundations and Trends® in Electric Energy Systems, vol. 4, no. 1-2, pp. 1–221, 2013.

[5] F. F. Wu, “Power system state estimation: a survey,” International Journal of Electrical Power & Energy Systems, vol. 12, no. 2, pp. 80–87, 1990.

[6] A. Monticelli, State estimation in electric power systems: a generalized approach. Springer Science & Business Media, 2012.

[7] M. H. Kim and B. Baldick, “A primal-dual interior point algorithm for nonlinear optimization,” SIAM Journal on Optimization, vol. 20, no. 5, pp. 2281–2299, 2010.

[8] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, “An interior algorithm for nonlinear optimization that combines line search and trust region steps,” Mathematical programming, vol. 107, no. 3, pp. 391–408, 2006.

[9] N. P. Padhy, “Unit commitment—a bibliographical survey,” IEEE transactions on power systems, vol. 19, no. 2, pp. 1196–1205, 2004.

[10] G. Hug-Glanzmann and G. Andersson, “Decentralized optimal power flow for smart microgrids,” IEEE Transactions on Smart Grid, vol. 4, no. 3, pp. 1546–1553, 2013.

[11] A. Y. Lam, B. Zhang, and N. T. David, “Distributed algorithms for optimal power flow problem,” in Decision and Control (CDC), 2012 IEEE 51st Annual Conference on. IEEE, 2012, pp. 430–437.

[12] M. Kräling, C. Chiu, J. Lavaei, and S. Boyd, “Dynamic network energy management via proximal message passing,” Foundations and trends in optimization, vol. 1, no. 2, pp. 70–122, 2013.

[13] V. DeMiguel and F. J. Nogales, “On decomposition methods for a class of partially separable nonlinear programs,” Mathematics of Operations Research, vol. 33, no. 1, pp. 119–139, 2008.

[14] H. Jia, W. Qi, Z. Liu, B. Wang, Y. Zeng, and T. Xu, “Hierarchical risk assessment of transmission system considering the influence of active distribution network,” IEEE Transactions on Power Systems, vol. 30, no. 2, pp. 1084–1093, 2015.

[15] S. H. Low, “Convex relaxation of optimal power flow: A tutorial,” in Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid (IREP Symposium). IEEE, 2013, pp. 1–15.

[16] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” Mathematical programming, vol. 106, no. 1, pp. 25–57, 2006.

[17] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, “An interior algorithm for nonlinear optimization that combines line search and trust region steps,” Mathematical programming, vol. 107, no. 3, pp. 391–408, 2006.

[18] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörffer, “Generic existence of unique lagrange multipliers in ac optimal power flow,” IEEE control systems letters, vol. 2, no. 4, pp. 791–796, 2018.

[19] V. M. Zavala, C. D. Laird, and L. T. Biegler, “A fast moving horizon estimation algorithm based on nonlinear programming sensitivity,” Journal of Process Control, vol. 18, no. 9, pp. 876–884, 2008.

[20] P. T. Boggs and J. W. Tolle, “Sequential quadratic programming,” Acta numerica, vol. 4, pp. 1–51, 1995.

[21] F. A. Potra and S. J. Wright, “Interior-point methods,” Journal of Computational and Applied Mathematics, vol. 124, no. 1-2, pp. 281–302, 2000.

[22] J. Nocedal and S. Wright, Numerical optimization. Springer Science & Business Media, 2006.

[23] V. DeMiguel and W. Murray, “A local convergence analysis of bilevel decomposition algorithms,” Optimization and Engineering, vol. 7, no. 2, pp. 99–133, 2006.

[24] R. D. Braun and I. M. Kroo, “Development and application of the collaborative optimization architecture in a multidisciplinary design environment,” Multidisciplinary design optimization: State of the art, vol. 80, p. 99, 1997.

[25] R. H. Byrd, “Optimal power flow algorithms,” in Proceedings of the 20th National Conference on Power Systems, 2005.

[26] R. H. Byrd, J. Nocedal, and R. A. Waltz, “Knitro: An integrated package for nonlinear optimization,” in Large-scale nonlinear optimization. Springer, 2006, pp. 35–59.

[27] P. M. Subcommittee, “lee reliability test system,” IEEE Transactions on power apparatus and systems, no. 6, pp. 2047–2054, 1979.

[28] V. DeMiguel and F. J. Nogales, “Network reconfiguration in distribution systems for loss reduction and load balancing,” IEEE Transactions on Power Delivery, vol. 4, no. 2, pp. 1401–1407, 1989.

[29] S. Tu, “90-bus transmission-distribution model,” 2019, data retrieved from DR POWER Data Repository. [https://griddata.org/dataset/90-bus-transmission-distribution-model]