

Hypergraph-Based Fast Distributed AC Power Flow Optimization

Xinliang Dai, Yingzhao Lian, Yuning Jiang, Colin N. Jones and Veit Hagenmeyer

Abstract—This paper presents a novel distributed approach for solving AC power flow (PF) problems. The optimization problem is reformulated into a distributed form using a communication structure corresponding to a hypergraph, by which complex relationships between subgrids can be expressed as hyperedges. Then, a hypergraph-based distributed sequential quadratic programming (HDSQP) approach is proposed to handle the reformulated problems, and the hypergraph-based distributed quadratic optimization algorithm (HDQ) is used as the inner algorithm to solve the corresponding QP subproblems, which are respectively condensed using Schur complements with respect to coupling variables defined by hyperedges. Furthermore, we rigorously establish the convergence guarantee of the proposed algorithm with a locally quadratic rate and the one-step convergence of the inner algorithm when using the Levenberg-Marquardt regularization. Our analysis also demonstrates that the computational complexity of the proposed algorithm is much lower than the state-of-art distributed algorithm. We implement the proposed algorithm in an open-source toolbox, `rapidPF`¹, and conduct numerical tests that validate the proof and demonstrate the great potential of the proposed distributed algorithm in terms of communication effort and computational speed.

I. INTRODUCTION

The power flow (PF) problem is a fundamental problem in power system analysis and has many applications, such as planning, expanding, and operating power systems [1]. Traditionally, centralized methods such as Gauss-Seidel [2] or Newton-type methods [3], [4] have been used to solve PF problems. In recent years, several studies were carried out in various aspects, including analysis of power flow equations [5], state estimation [6], [7], distributionally robust optimal control [8], initialization strategies [9], [10], convex relaxation [11], [12], and convex restriction [13]. With the increasing penetration of distributed energy resources and the need for optimization and control of power systems with many controllable devices, distributed approaches have gained significant research attention [14]. For systems like Germany’s power grid, which has four transmission system operators (TSOs) and over 900 distribution system opera-

This work was supported in part by the German Federal Ministry of Education and Research within the project MORENet – Modellierung, Optimierung und Regelung von Netzwerken heterogener Energiesysteme mit volatiler erneuerbarer Energieerzeugung, in part by the BMBF-project ENSURE II with grant number 03SFK1F0-2 and in part by the Swiss National Science Foundation (SNSF) under the NCCR Automation project, grant agreement 51NF40.180545.

XD and VH are with Karlsruhe Institute of Technology, Karlsruhe, Germany. Email: {xinliang.dai, veit.hagenmeyer}@kit.edu

YZ, YJ and CJ are with École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. Email: {yingzhao.lian, yuning.jiang, colin.jones}@epfl.ch

¹Open-source toolkit: <https://github.com/xinliang-dai/rapidPF>

tors (DSOs), sharing detailed grid models is not preferred. Therefore, a centralized approach is not preferred by systems operators or is even prohibited by the respective regulation.

The present paper focuses on AC models to obtain more realistic results. The main challenge is that the AC PF feasibility is NP-hard [15], [16], and is a challenge even for a centralized approach. [17] proposed to solve AC PF problems by breaking the original problem into several smaller power flow subproblems, keeping coupling variables fixed, and then iterating over them. In the follow-up work [18], the convergence was analyzed under some additional assumptions. However, the actual convergence behaviors and scalability are limited in practice. Other well-known distributed algorithms for AC power flow analysis—Optimality Condition Decomposition (OCD) proposed by [19], Auxiliary Problem Principle (APP) by [20], and Alternating Direction Method of Multipliers (ADMM) by [21]—have no convergence guarantees in general, and their convergence behaviors are case-by-case in practice.

Recently, [22] proposed a second-order distributed algorithm, i.e., Augmented Lagrangian based Alternating Direction Inexact Newton method (ALADIN). In contrast to these existing distributed approaches, ALADIN can provide a local convergence guarantee with a quadratic convergence rate for generic distributed nonconvex optimization problems if suitable Hessian approximations are used. Based on the ALADIN algorithm, considerable works have been carried out for power system analysis [23]–[25]. [26] provides open-source MATLAB code for rapid prototyping for distributed power flow (`rapidPF`). Extensive research [27] has improved computing time significantly for solving large-scale AC PF problems by using Gauss-Newton approximation and further exploiting the problem formulation. However, ALADIN is limited by the required computation and communication effort per iteration.

The aforementioned studies either intertwine problem formulation and problem solution or use the standard affinely coupled distributed form. In contrast, we introduce a hypergraph [28] based AC PF framework in the present paper. As a generalization of graphs, hypergraph allows more than two nodes to be connected in the same hyperedge, therefore depicting more complex relationships, e.g., multiple regions connected to a region at the same bus. More specifically, we propose to reformulate the AC PF problem as a zero-residual least-squares problem with a communication structure corresponding to a hypergraph and then solve it by an hypergraph-based distributed sequential quadratic programming (HDSQP) approach. The convergence of the proposed HDSQP is guaranteed, and the convergence rate is quadratic

when approaching the minimizer if the Levenberg-Marquardt method is used. Moreover, the condensed QP subproblems of HDSQP are solved by a variant of the dual decomposition algorithm, i.e., the hypergraph-based distributed quadratic optimization algorithm (HDQ) proposed by [29]. Most notably, the communication matrix of the dual decomposition is the Bollas' Laplacian for hypergraphs [30], and it is also a projection matrix for the HDQ algorithm. As a result, HDQ can converge to the global minimizer in one iteration if the corresponding QP subproblem is convex. This hints at the fact that the proposed distributed approach could converge rapidly with less communication effort.

The aim of the present paper is to exploit the hypergraph-based distributed approach for solving generic AC PF problems. The main contributions are listed in the following:

- (i) We propose a new distributed form of the AC PF problem that uses a communication structure corresponding to a hypergraph to generalize complex relationships between subgrids. Moreover, we propose a hypergraph-based distributed sequential quadratic programming (HDSQP) approach for solving the problem and use the hypergraph-based distributed quadratic optimization algorithm (HDQ) [29] as the inner algorithm for the corresponding condensed QP subproblem at each iteration.
- (ii) We rigorously establish the convergence guarantee of the proposed algorithm HDSQP with a locally quadratic rate and the one-step convergence of the inner algorithm HDQ when using the Levenberg-Marquardt regularization. Our analysis also demonstrates that the computational complexity of the proposed algorithm is much lower than the state-of-art distributed algorithm. Numerical tests are added to the `rapidPF` open-source toolbox, and we show that the proposed algorithm surpasses the state-of-art ALADIN algorithm with respect to computing time and communication effort.

This paper introduces the distributed formulation of AC PF problem in Section II. Then, we present the proposed hypergraph-based distributed optimization algorithm and the convergence analysis in Section III. Finally, we present numerical simulations in Section IV and conclude the paper in Section V. Additionally, an anonymous chat is listed in Appendix.

II. SYSTEM MODEL AND PROBLEM FORMULATION

This paper considers a power system defined by a tuple $\mathcal{S} = (\mathcal{R}, \mathcal{N}, \mathcal{L})$ with the set \mathcal{R} of all regions, \mathcal{N} the set of all buses and \mathcal{L} the set of all branches. We define by n^{reg} , n^{bus} , and n^{line} the number of regions, buses, and branches, respectively. In the present paper, we use complex voltage in polar coordinates:

$$V_i = v_i e^{j\theta_i}, \quad i \in \mathcal{N}.$$

where v_i and θ_i denote the voltage magnitude and angle. Thereby, for each bus $i \in \mathcal{N}$, its steady state $(\theta_i, v_i, p_i, q_i)$ includes v_i and θ_i the voltage magnitude and angle, p_i and q_i the active and reactive power. Throughout this paper, we stack all steady states at region $\ell \in \mathcal{R}$ by $\chi_\ell =$

$\{(\theta_i, v_i, p_i, q_i)\}_{i \in \mathcal{N}_\ell}$ with \mathcal{N}_ℓ bus set at region ℓ . When a vector ξ consists of n^{reg} subvectors, we write $\xi = (\xi_1, \dots, \xi_{n^{\text{reg}}})$. Moreover, x_ℓ denotes the coupled variables, and y_ℓ denotes hidden variables that are totally local to region $\ell \in \mathcal{R}$, i.e., $\chi_\ell := (x_\ell, y_\ell)$. Thus, there is a matrix A_ℓ such that $x_\ell = A_\ell \chi_\ell$.

A. Hypergraph-based modeling

As discussed in [26], we share the components between neighboring regions to ensure physical consistency. Let us take the 6-bus system with two regions, shown in Fig. 1 as an example. The coupled system, shown in Fig. 1(a), has been partitioned into two local regions. To solve the AC PF problem in the region R_1 , besides its buses $\{1, 2, 3\}$ called the *core buses*, the complex voltage of bus $\{4\}$ from neighboring region R_2 is required. Hence, for the sub-problem of the region R_1 , we create an auxiliary bus $\{4\}$ called the *copy bus*, along with its own *core bus*, to formulate a self-contained AC PF problem. The resulting affine consensus constraint can be written as

$$\theta_3^{\text{core}} = \theta_3^{\text{copy}}, \quad \theta_4^{\text{core}} = \theta_4^{\text{copy}}, \quad (1a)$$

$$v_3^{\text{core}} = v_3^{\text{copy}}, \quad v_4^{\text{core}} = v_4^{\text{copy}}. \quad (1b)$$

The multi-region power system \mathcal{S} can be transformed into a

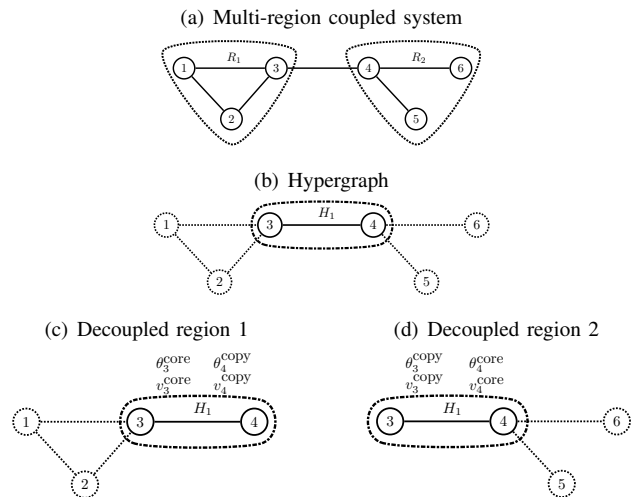


Fig. 1. Decomposition by sharing components for a two-region system

hypergraph $\mathcal{G} = (\mathcal{N}, \mathcal{H})$, where $\mathcal{H} = (H_1, \dots, H_n)$ denotes the set of all hyperedges that can include any number of buses, c.f. [31]. The example mentioned above has only one hyperedge, as shown in Fig. 1(b). The buses $\{1, 2, 5, 6\}$ are called isolated or hidden buses since they are not incident to any hyperedge. Thereby, we have coupling variables of region R_1 and R_2 , i.e., $x_1 = (\theta_3^{\text{core}}, \theta_4^{\text{copy}}, v_3^{\text{core}}, v_4^{\text{copy}})$ and $x_2 = (\theta_3^{\text{copy}}, \theta_4^{\text{core}}, v_3^{\text{copy}}, v_4^{\text{core}})$. Moreover, we denote by $z = (\theta_3, \theta_4, v_3, v_4)$ to represent the common values for coupling variables of all regions. As a result, the consensus constraints (1) can be written as

$$x_\ell = E_\ell z, \quad \ell \in \mathcal{R}, \quad (2)$$

where E_ℓ denote the incident matrix of a specific region $\ell \in \mathcal{R}$.

B. Hypergraph-based AC Power Flow Problem

In polar coordinates, the resulting conventional AC power flow (PF) problem can be written as a set of power flow equations

$$p_i = p_i^g - p_d^l = v_i \sum_{k \in \mathcal{N}} v_k (\mathcal{G}_{ik} \cos \theta_{ik} + \mathcal{B}_{ik} \sin \theta_{ik}), \quad (3a)$$

$$q_i = q_i^g - q_d^l = v_i \sum_{k \in \mathcal{N}} v_k (\mathcal{G}_{ik} \sin \theta_{ik} - \mathcal{B}_{ij} \cos \theta_{ik}) \quad (3b)$$

for all buses $i \in \mathcal{N}$, where p_i^g, q_i^g (reps. p_i^d, q_i^d) denote the real and reactive power injections from generator(s) (resp. loads) at bus i , θ_{ik} denotes the angle difference between bus i and bus k , $\mathcal{G}_{ij}, \mathcal{B}_{ij}$ are the real and imaginary parts of the bus admittance matrix entries $Y_{ik} = \mathcal{G}_{ij} + j\mathcal{B}_{ij}$.

Remark 1 *Since multiple stable AC power flow solutions may exist [32], [33], especially in the presence of power flow reversal on distribution systems [34], [35], the present paper focuses on local solutions.*

Following [27], these power flow equations can be written as a residual function

$$r(\chi) = 0 \quad (4)$$

with state variables $\chi = (\theta, v, p, g)$, so that the AC PF problems can be formulated as a standard zero-residual least-squares problem

$$\min_{\chi} f(\chi) = \frac{1}{2} \|r(\chi)\|_2^2. \quad (5)$$

In other words, f is the sum of squared residuals of power flow equations for all buses $i \in \mathcal{N}$. Hence, both the state variables χ and the objective f are separable, i.e.,

$$f(\chi) = \sum_{\ell \in \mathcal{R}} f_{\ell}(\chi_{\ell}) = \sum_{\ell \in \mathcal{R}} \frac{1}{2} \|r_{\ell}(\chi_{\ell})\|_2^2. \quad (6)$$

As a result, the coupled problem (5) can be reformulated with a communication structure corresponding to a hypergraph

$$\min_{\chi} f(\chi) = \sum_{\ell \in \mathcal{R}} f_{\ell}(\chi_{\ell}) \quad (7a)$$

$$\text{subject to } x_{\ell} = E_{\ell} z \quad | \quad \lambda_{\ell}, \ell \in \mathcal{R}. \quad (7b)$$

Recall that χ consists of two components, i.e.,

$$\chi_{\ell} = (x_{\ell}, y_{\ell}), \quad \forall \ell \in \mathcal{R}, \quad (8)$$

where x_{ℓ} represents coupling variables and y_{ℓ} represents hidden variables that are entirely local.

III. DISTRIBUTED OPTIMIZATION ALGORITHM

This section introduces the sequential quadratic programming (SQP) framework and the condensed reformulation of QP subproblem at each iteration. Then, based on the preliminaries introduced in Section II, we propose a hypergraph-based distributed approach to solve the AC PF problem (7). Convergence analysis is carried out at the end of this section.

A. Preliminaries

The sequential quadratic programming (SQP) framework is one of the most effective methods for Nonlinear Programming (NLP), in which a sequence of QP subproblems are iteratively constructed and solved. This paper focuses on the full-step variant and its corresponding local convergence.

Because in practice, a conventional flat start can always provide a good initial guess for AC PF problems. Regarding the globalization routine or line search for SQP, more details refer to [36, Chapter 18].

At the k -th iteration, the QP subproblem is written as

$$\min_{\chi^{k+1}, z} m^k(\chi^{k+1}) = \sum_{\ell \in \mathcal{R}} m_{\ell}^k(\chi_{\ell}^{k+1}) \quad (9a)$$

$$\text{subject to } A_{\ell} \chi_{\ell}^{k+1} = E_{\ell} z \quad | \quad \lambda_{\ell}, \forall \ell \in \mathcal{R}. \quad (9b)$$

with quadratic models

$$m_{\ell}^k(\chi_{\ell}) = \frac{1}{2} (\chi_{\ell})^{\top} \nabla^2 f_{\ell}^k \chi_{\ell} + (\nabla f_{\ell}^k - \nabla^2 f_{\ell}^k \chi_{\ell}^k)^{\top} \chi_{\ell}, \quad (10)$$

and for notation simplification, $f_{\ell}^k = f_{\ell}(\chi_{\ell}^k)$ used for all $\ell \in \mathcal{R}$. The derivatives of the objectives f_{ℓ} at iterate χ_{ℓ}^k can be expressed as

$$\nabla f_{\ell}^k = (J_{\ell}^k)^{\top} r_{\ell}^k, \quad (11a)$$

$$\nabla^2 f_{\ell}^k = (J_{\ell}^k)^{\top} J_{\ell}^k + Q_{\ell}^k \quad (11b)$$

with

$$J_{\ell}^k = J_{\ell}(\chi_{\ell}^k) = [\nabla r_{\ell,1}, \nabla r_{\ell,2}, \dots, \nabla r_{\ell, n_{\text{pf}}}]^{\top}, \quad (12a)$$

$$Q_{\ell}^k = Q_{\ell}(\chi_{\ell}^k) = \sum_{m=1}^{n_{\text{pf}}} r_{\ell,m}(\chi_{\ell}^k) \nabla^2 r_{\ell,m}(\chi_{\ell}^k). \quad (12b)$$

Here, $r_{\ell,m}$ represents the residual of the m -th power flow equation at region $\ell \in \mathcal{R}$. In practice, the first term $(J_{\ell}^k)^{\top} J_{\ell}^k$ dominates the second term Q_{ℓ}^k , because the residuals $r_{\ell,m}$ are close to zero near the solution [36, Chapter 10]. In the present paper, Levenberg-Marquardt regularization is used, i.e.,

$$B_{\ell}^k = (J_{\ell}^k)^{\top} J_{\ell}^k + \varepsilon I, \quad \forall \ell \in \mathcal{R},$$

to approximate Hessians such that (9) is strongly convex. Here, one empirical choice of ε in practice, is $\varepsilon = 10^{-10}$. As discussed in [36, Chapter 10], when ε is sufficiently small, the Levenberg-Marquardt method shares the same performance with the classical Gauss-Newton method under mild assumptions.

In order to write the QP subproblems (9) in a condensed form, we set

$$F_{\ell}(x_{\ell}) = \min_{y_{\ell}} f_{\ell}(x_{\ell}, y_{\ell}), \quad (13)$$

where x_{ℓ} denotes the coupling variables and y_{ℓ} denotes the variables that are entirely local. Accordingly, the Jacobian matrix J_{ℓ}^k can be partitioned into two blocks w.r.t. x_{ℓ} and y_{ℓ} as

$$J_{\ell}^k = [J_{\ell}^x, J_{\ell}^y], \quad \forall \ell \in \mathcal{R}. \quad (14)$$

Consequently, the gradient g_{ℓ}^k and the approximated Hessians B_{ℓ}^k can be written as

$$g_{\ell}^k = \left[(J_{\ell}^x)^{\top} r_{\ell}^x, (J_{\ell}^y)^{\top} r_{\ell}^y \right] \text{ and } B_{\ell}^k = \begin{bmatrix} B_{\ell}^{xx} & B_{\ell}^{xy} \\ B_{\ell}^{yx} & B_{\ell}^{yy} \end{bmatrix} \quad (15)$$

with the assistance of the Levenberg-Marquardt method

$$B_{\ell}^{xx} = (J_{\ell}^x)^{\top} J_{\ell}^x + \varepsilon I, \quad (16a)$$

$$B_{\ell}^{yy} = (J_{\ell}^y)^{\top} J_{\ell}^y + \varepsilon I, \quad (16b)$$

$$B_{\ell}^{xy} = (B_{\ell}^{yx})^{\top} = (J_{\ell}^x)^{\top} J_{\ell}^y \quad (16c)$$

for all $\ell \in \mathcal{R}$. By using the Schur complement, the first and the second derivatives of the function, F_ℓ can be given by

$$\bar{g}_\ell^k = g_\ell^x - B_\ell^{xy} [B_\ell^{yy}]^{-1} g_\ell^y, \quad (17a)$$

$$\bar{B}_\ell^k = B_\ell^{xx} - B_\ell^{xy} [B_\ell^{yy}]^{-1} B_\ell^{yx}. \quad (17b)$$

Thereby, the QP subproblems (9) can be reformulated into a condensed form

$$\min_{x,z} \bar{m}^k(x) = \sum_{\ell \in \mathcal{R}} \bar{m}_\ell^k(x_\ell) \quad (18a)$$

$$\text{subject to } x_\ell = E_\ell z \quad | \lambda_\ell, \forall \ell \in \mathcal{R} \quad (18b)$$

with reduced quadratic models

$$\bar{m}_\ell^k(x_\ell) = \frac{1}{2} x_\ell^\top \bar{B}_\ell^k x_\ell + (\bar{g}_\ell^k - \bar{B}_\ell^k x_\ell^k)^\top x_\ell. \quad (19)$$

Here, $x_\ell^k = A_\ell \chi^k$ for all $\ell \in \mathcal{R}$. The Lagrangian function of the condensed QP subproblem (18) can thus be written as

$$\mathcal{L}(x, z, \lambda) = \sum_{\ell \in \mathcal{R}} \{ \bar{m}_\ell^k(x_\ell) + \lambda_\ell^\top x_\ell \} - \lambda^\top E z \quad (20)$$

with $E = [E_1^\top, \dots, E_\ell^\top]^\top$. The Karush–Kuhn–Tucker (KKT) conditions of (18) are as follows,

$$\nabla_x \mathcal{L} = 0 = \bar{B}^k (x - x^k) + \bar{g}^k + \lambda, \quad (21a)$$

$$\nabla_z \mathcal{L} = 0 = E^\top \lambda, \quad (21b)$$

$$\nabla_\lambda \mathcal{L} = 0 = x - E z, \quad (21c)$$

where $\bar{B}^k = \text{diag}\{\bar{B}_\ell^k\}_{\ell \in \mathcal{R}}$ stacks all \bar{B}_ℓ^k into a block diagonal matrix.

B. Hypergraph-Based SQP

Based on Section III-A, we propose an hypergraph-based distributed sequential quadratic programming (HDSQP) approach to solve (7). More specifically, the hypergraph-based distributed quadratic optimization algorithm (HDQ), proposed as a variant of dual decomposition by [29], is implemented to solve its condensed QP subproblems (18). Remarkably, HDQ can converge with convexity assumption in one iteration to save total computing time and communication effort.

Algorithm 1 outlines the proposed HDSQP. Line 1 evaluates derivatives of the full-dimensional subproblem (9) and the corresponding condensed subproblems (18) with assistance of the Levenberg-Marquardt method (22) and the Schur complement (17). The resulting condensed subproblem (18) is a strongly convex QP with a communication structure corresponding to a hypergraph. Thereby, HDQ is added as inner algorithm to solve the condensed QP subproblem (18) (Line 2-4) due to fast convergence rate. The HDQ algorithm consists of three steps. In Line 2, temporary local coupling variables \bar{x}_ℓ for all region $\ell \in \mathcal{R}$ are obtained with respect to the KKT condition (21a) under initial condition $\lambda_\ell = 0$. Then, weighted averaging is conducted to compute a temporary state \bar{z} in Line 3, where the weights are determined by the condensed Hessian approximation $\bar{B}^k = \text{diag}\{\bar{B}_\ell^k\}_{\ell \in \mathcal{R}}$. In the fourth step, the dual variable λ is updated based on the deviation of temporary weighted primal residual $\bar{B}^k (\bar{x} - E \bar{z})$ in (25a).

Due to the positive definiteness of approximated Hessians based on the Levenberg-Marquardt method, the HDQ can

Algorithm 1 Hypergraph-based Distributed Sequential Quadratic Programming (HDSQP)

Initialization: χ^0 as a flat start

repeat

1 Evaluate derivatives at iterate χ^k

$$B_\ell^k = (J_\ell^k)^\top J_\ell^k + \varepsilon I \quad \text{and} \quad g_\ell^k = (J_\ell^k)^\top r_\ell^k, \quad (22)$$

and the corresponding condensed derivatives \bar{B}_ℓ^k and \bar{g}_ℓ^k by Schur complement (17) for all $\ell \in \mathcal{R}$

2 Compute temporary local coupling variables

$$\bar{x}_\ell = (\bar{B}_\ell^k)^{-1} (\bar{B}_\ell^k x_\ell^k - \bar{g}_\ell^k). \quad (23)$$

with $x_\ell^k = A_\ell \chi_\ell^k$, which essentially solves the decoupled subproblems for all $\ell \in \mathcal{R}$.

3 Compute weighted average

$$\bar{z} = \left(\sum_{\ell \in \mathcal{R}} E_\ell^\top \bar{B}_\ell^k E_\ell \right)^{-1} \sum_{\ell \in \mathcal{R}} E_\ell^\top \bar{B}_\ell^k \bar{x}_\ell. \quad (24)$$

4 Update primal and dual variables for all $\ell \in \mathcal{R}$ by

$$\lambda_\ell^{k+1} = \bar{B}_\ell^k (\bar{x}_\ell - E_\ell \bar{z}), \quad (25a)$$

$$\chi_\ell^{k+1} = (B_\ell^k)^{-1} (B_\ell^k \chi_\ell^k - g_\ell^k - A_\ell^\top \lambda_\ell^{k+1}). \quad (25b)$$

until Primal variables χ converge;

converge to the global minimizer of the condensed QP subproblem (18) in one iteration for saving computational and communication efforts. In the end, based on dual variable λ provided by the inner algorithm HDQ, the new full-dimensional iterate χ^{k+1} is updated by (25b). Note that all the steps in Algorithm 1 can be executed in parallel, except for the weighted averaging (Line 3).

C. Local Convergence Analysis

Without loss of generality, we assume that the flat start can provide a good initial guess (Remark 3) such that the present paper focuses on the local convergence of Algorithm 1. Here, local means that the initial iterate is located in a small neighborhood of a local minimizer, within which the solution has physical meaning.

In the following, we first analyze the convergence of the inner algorithm HDQ for solving condensed QP subproblems (18), and prove that it can converge to a global minimizer in one step. Then, regardless of the inexactness caused by condensing subproblems, we prove that Algorithm 1 can converge with a locally quadratic convergence rate when the Levenberg-Marquardt method is used to approximate Hessians.

Proposition 1 *Let the Levenberg-Marquardt method be used to evaluate B_ℓ^k such that the condensed QP subproblem (18) is strongly convex, then*

$$(x_\ell^{k+1} := A_\ell \chi_\ell^{k+1}, z^{k+1}, z^{k+1} := \bar{z}, \lambda_\ell^{k+1})$$

given by Algorithm 1 solves (18) at iteration k .

The detailed proof appears in Appendix A. As we know

that the AC power flow equation (3) is sufficiently smooth, the objective f is twice-Lipschitz continuously differentiable, i.e., there exists a constant $L > 0$

$$\frac{\|\nabla f(\chi) - \nabla f(\chi^*)\|}{\|\chi - \chi^*\|} = \|\nabla^2 f(\tilde{\chi})\| \leq L \quad (26)$$

with $\tilde{\chi} = \chi - t(\chi - \chi^*)$ for some $t \in (0, 1)$. Moreover, since the optimal solution is feasible to (3), we have zero-residual $r^* = 0$ at the optimizer and

$$Q^* = Q(\chi^*) = 0. \quad (27)$$

Locally, we can thus, have $\|Q(\chi)\| = \mathcal{O}(\|\chi - \chi^*\|)$. Before we establish the local convergence result of Algorithm 1, we introduce the definition of regular KKT point for (7).

Definition 1 (Regular KKT point of (7)) A KKT point of (7) is called regular if second order sufficient condition (SOSC) and linear independence constraint qualification (LICQ) hold at the KKT point [36].

Remark 2 Due to (27), Hessian is equivalent to Gauss-Newton approximation at a local minimizer χ^* . Moreover, the Jacobian matrix of the power flow equations (3) is always full-row rank in practice. Thus, SOSC is satisfied for the problem (7).

Additionally, the coupling introduced in Problem (7) is based on the hypergraph, we have LICQ hold for the coupled affine equality constraints (7b), i.e., E is full row rank. As a result, a KKT point for the problem (7) is regular.

Theorem 1 Let the minimizer (χ^*, λ^*) satisfy SOSC such that (χ^*, λ^*) is a regular KKT point, let the parameter ε be sufficiently small. Then, for solving the problem (7), the iterates χ of Algorithm 1 converges locally with a quadratic convergence rate.

If the exact Hessian is used in the QP subproblems (9), the corresponding p^N can be viewed as a standard Newton step of the original least-squares problem (5). Similar to [36], we have

$$\begin{aligned} \|p^N\| &\leq \|\chi^k - \chi^*\| + \|\chi^k + p^N - \chi^*\| \\ &\leq \|\chi^k - \chi^*\| + \hat{L} \|\chi^k - \chi^*\|^2, \end{aligned} \quad (28)$$

where $\hat{L} = L \|(\nabla^2 f^*)^{-1}\|$ and L is the Lipschitz constant for $\nabla^2 f$ for χ near χ^* . Since ε is sufficiently small, the Levenberg-Marquardt method shares the same properties with the Gauss-Newton method. Hence, the analysis in the following is based on the Gauss-Newton method. The detailed proof is given in Appendix B.

IV. NUMERICAL CASE STUDY

In this section, we illustrate the performance of the proposed distributed approach to solve AC PF problems and compare it with the state-of-art ALADIN algorithm.

A. Implementation

The framework presented in this paper is implemented in MATLAB-R2021a, and both the hypergraph-based problem and the proposed HDSQP algorithm are provided in the rapidPF toolbox². As shown in Fig. 2, the toolbox allows users to combine multiple MATPOWER casefiles [37] into a single merged casefile, formulate AC power flow problems as distributed optimization problems and then solve the problems by distributed approaches. Compared with previous work [26], [27], the toolbox can reformulate the problems with a communication structure corresponding to a hypergraph (7) and solve them by the proposed approach HDSQP. Additionally, the IPOPT solver [38] is used for calculating reference solutions, and the CasADi toolbox [39] is used to compute exact Hessians for analysis.

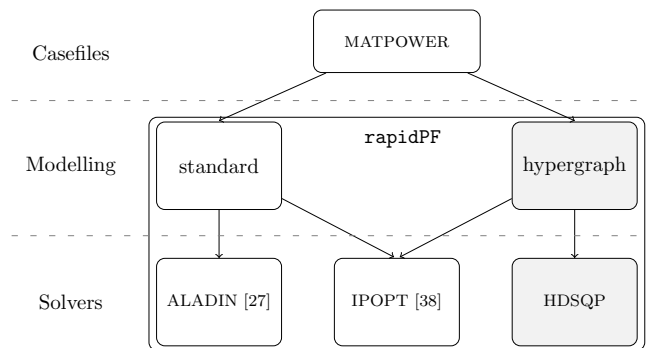


Fig. 2. The open-source toolbox rapidPF

The case studies are carried out on a standard desktop computer with Intel[®] i5-6600K CPU @ 3.50GHz and 16.0 GB installed RAM. Following [27], three benchmarks are created by using the rapidPF toolbox based on IEEE standard test systems. The ALADIN algorithm in the same toolbox is used for comparison. For a fair comparison, both ALADIN and HDSQP algorithms are initialized with a flat start. The computational time is estimated under the assumption that all subproblems are solved in parallel, and the time spent on exchanging sensitivities information is not taken into consideration.

Remark 3 (Initialization) AC PF problems are usually initialized with a flat start, where all voltage angles are set to zero, and all voltage magnitudes are set to 1.0 p.u. [40]. The initialization strategy has been demonstrated numerically that it can provide a good initial guess for the distributed approach in practice [26], [27].

B. Case Studies

Three test cases are studied in the present paper, as shown in Table I. Note that $n^{\text{reg}} = |\mathcal{R}|$, n^{state} and n^{cpl} represent the number of regions, i.e., the cardinality of \mathcal{R} , the dimension of the state variables χ and the coupling variables

²Toolbox Documentation: <https://xinliang-dai.github.io/rapidPF/>

TABLE I
NUMERICAL COMPARISON: HDSQP VS ALADIN

Case	n^{bus}	n^{reg}	n^{state}	n^{cpl}	Algorithm	Iterations	Time [10^{-2} s]	$\ \chi - \chi^*\ $	$ f(\chi) $	Primal Residual
1	53	3	232	40	ALADIN	4	2.33	6.13×10^{-09}	5.05×10^{-12}	3.61×10^{-9}
					HDSQP	6	1.34	4.08×10^{-08}	1.52×10^{-20}	4.03×10^{-8}
2	418	2	1684	24	ALADIN	4	7.17	1.82×10^{-09}	6.37×10^{-09}	7.32×10^{-9}
					HDSQP	10	5.68	1.39×10^{-06}	3.04×10^{-21}	1.37×10^{-9}
3	1180	10	4764	88	ALADIN	5	7.37	4.36×10^{-11}	2.59×10^{-11}	2.56×10^{-9}
					HDSQP	6	3.17	2.17×10^{-08}	1.21×10^{-20}	2.18×10^{-8}

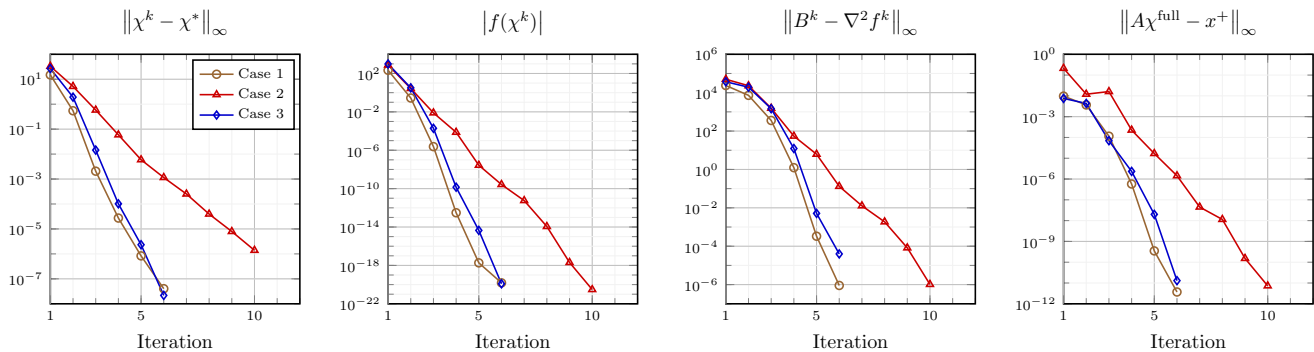


Fig. 3. Convergence behavior of HDSQP

x , respectively. To illustrate the convergence performance of the HDSQP algorithm, we introduce four quantities, i.e.,

- 1) deviation of state iterates to the minimizer $\|\chi^k - \chi^*\|$,
- 2) power flow residual $|f(\chi^k)|$,
- 3) error of Levenberg-Marquardt Approximation $\|B^k - \nabla^2 f^k\|$,
- 4) inexactness caused by condensing QP subproblems, i.e., deviation between the solution x^+ to condensed QP subproblems (18) with the solution χ^{full} to the full-dimensional QP subproblems (9)

$$\|A\chi^k - x^k\| \quad \text{with } A = \text{diag}\{A_\ell\}_{\ell \in \mathcal{R}}.$$

For a fair comparison, all the problems are initialized with a flat start. Table I shows that the proposed HDSQP can converge rapidly to a very highly accurate solution regarding the deviation of state variables, power flow residuals, and primal residual $x - Ez$. Furthermore, the convergence behavior of the proposed algorithm for all three test cases is presented in Fig. 3. As the primal iterates χ^k approach, the minimizer χ^* , Levenberg-Marquardt Approximation, and the solution to the QP subproblems by using Schur decomposition become more accurate. Case 1 and Case 3 share almost the same performance and converge in 6 iterations, while Case 2 takes more iterations. The relative lower accuracy of the Schur complement possibly slows down the overall convergence rate of the proposed HDSQP approach.

C. HDSQP vs. ALADIN

The ALADIN algorithm used for comparison is a Gauss-Newton-based variant tailored to deal with AC power flow problem in [27]. As discussed in [27], this ALADIN variant has been illustrated that it outperforms the other existing state-of-art distributed approaches. Therefore, to illustrate the effectiveness of Algorithm 1, we compare it with this Gauss-Newton ALADIN variant. First of all, let us have a close

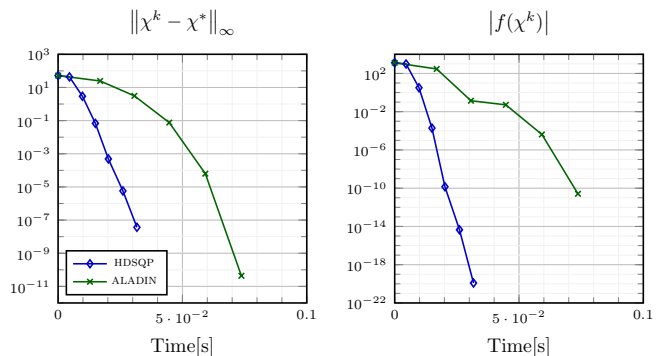


Fig. 4. Comparison of different algorithms for Case 3

look at the computational complexity per iteration of both algorithms. The Gauss-Newton ALADIN proposed in [27]

requires

$$\underbrace{\mathcal{O}\left(\sum_{\ell} (n_{\ell}^{\text{state}})^3\right)}_{\text{parallelizable}} + \underbrace{\mathcal{O}\left((n^{\text{state}})^3\right)}_{\text{consensus}}$$

float operations per iteration while HDSQP needs

$$\underbrace{\mathcal{O}\left(\sum_{\ell} (n_{\ell}^{\text{state}})^3\right)}_{\text{parallelizable}} + \underbrace{\mathcal{O}\left((n^{\text{cpl}})^3\right)}_{\text{consensus}}.$$

Here, one can see that the parallelizable computational complexity are same for both approaches, while the proposed Algorithm 1 is much cheaper in the consensus part. This is because $n^{\text{state}} \gg n^{\text{cpl}}$ always holds for electric power systems in practice, as shown in Table I. Fig. 4 displays the convergence behaviors of both algorithms for Case 3. Although ALADIN can converge with one iteration faster, the proposed HDSQP has a much shorter computing time per iteration, benefitting from the fast convergence of the inner algorithm and thus surpasses ALADIN in terms of total computing time.

Table I presents simulation results for comparison between the ALADIN and the HDSQP algorithms. For all three test cases, computational speedup of HDSQP can be 30% to 50% compared with ALADIN as presented in Table I. Furthermore, HDSQP requires only condensed Hessian in the centralized step, while ALADIN requires both the first and the second order derivatives from all subproblems to solve a full-dimensional coupled QP problem in the coordinator. This could further slow the total running time during parallel computing. Consequently, the proposed HDSQP outperforms ALADIN in solving AC PF problems in aspects of computing time and communication effort.

V. CONCLUSION AND OUTLOOK

The present paper proposes a distributed approach, hypergraph-based distributed sequential quadratic programming (HDSQP), for solving power flow (PF) problems. By introducing the hypergraph theory, the QP subproblems can be solved efficiently by the inner algorithm, i.e., the hypergraph-based distributed quadratic optimization algorithm (HDQ) [29]. A mathematical proof is provided that the inner algorithm HDQ can converge in one iteration, and the local convergence rate of the proposed HDSQP can achieve quadratic by implementing the Levenberg-Marquardt method to approximate Hessians. Simulation results and analysis of the computational complexity demonstrate that the proposed algorithm outperforms the state-of-the-art distributed algorithm in terms of computing time for small- and medium-sized power grids at the cost of slightly increased iterations. Moreover, the numerical tests are added to the open-source toolbox `rapidPF`.

One drawback of the proposed approaches is associated with the inner algorithm HDQ. The inner algorithm employs a weighted averaging technique that utilizes the Hessian matrix, resulting in solutions converging to an equilibrium point near the exact optimizer. This hinders the scalability and numerical robustness of the proposed HDSQP. To address

these limitations, future work could focus on tuning the Levenberg-Marquardt method or alternating between HD-SQP and ALADIN to enhance the scalability and numerical robustness.

REFERENCES

- [1] J. J. Grainger, *Power system analysis*. McGraw-Hill, 1999.
- [2] A. F. Glimm and G. W. Stagg, "Automatic calculation of load flows," *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 76, no. 3, pp. 817–825, 1957.
- [3] W. F. Tinney and C. E. Hart, "Power flow solution by newton's method," *IEEE Trans. Power Appar. Syst.*, vol. PAS-86, no. 11, pp. 1449–1460, 1967.
- [4] B. Stott and O. Alsac, "Fast decoupled load flow," *IEEE Trans. Power Appar. Syst.*, vol. PAS-93, no. 3, pp. 859–869, 1974.
- [5] K. Dvijotham, M. Chertkov, and S. Low, "A differential analysis of the power flow equations," in *2015 54th IEEE Conf. Decis. Control (CDC)*, pp. 23–30, 2015.
- [6] Y. Guo, X. Zhou, C. Zhao, L. Chen, G. Hug, and T. H. Summers, "An online joint optimization–estimation architecture for distribution networks," *IEEE Transactions on Control Systems Technology*, 2023.
- [7] Y. Guo, X. Zhou, C. Zhao, L. Chen, and T. H. Summers, "Optimal power flow with state estimation in the loop for distribution networks," *IEEE Systems Journal*, 2023.
- [8] J. Guo, G. Hug, and O. K. Tonguz, "A case for nonconvex distributed optimization in large-scale power systems," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3842–3851, 2016.
- [9] J. Thorp and S. Naqavi, "Load flow fractals," in *28th IEEE Conf. Decis. Control (CDC)*, pp. 1822–1827 vol.2, 1989.
- [10] J. Thorp, S. Naqavi, and H.-D. Chiang, "More load flow fractals," in *29th IEEE Conf. Decis. Control (CDC)*, pp. 3028–3030 vol.6, 1990.
- [11] R. Madani, J. Lavaei, and R. Baldick, "Convexification of power flow problem over arbitrary networks," in *2015 54th IEEE Conf. Decis. Control (CDC)*, pp. 1–8, 2015.
- [12] D. K. Molzahn, I. A. 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, 2019.
- [13] D. Lee, H. D. Nguyen, K. Dvijotham, and K. Turitsyn, "Convex restriction of power flow feasibility sets," *IEEE Trans. Control. Netw. Syst.*, vol. 6, no. 3, pp. 1235–1245, 2019.
- [14] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.
- [15] K. Lehmann, A. Grastien, and P. Van Hentenryck, "AC-feasibility on tree networks is NP-Hard," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 798–801, 2015.
- [16] D. Bienstock and A. Verma, "Strong NP-Hardness of AC power flows feasibility," *Oper. Res. Lett.*, vol. 47, no. 6, pp. 494–501, 2019.
- [17] H. Sun and B. Zhang, "Distributed power flow calculation for whole networks including transmission and distribution," in *2008 IEEE PES T&D Conference & Exposition*, pp. 1–6, IEEE, 2008.
- [18] H. Sun, Q. Guo, B. Zhang, Y. Guo, Z. Li, and J. Wang, "Master–slave-splitting based distributed global power flow method for integrated transmission and distribution analysis," *IEEE Trans. Smart Grid*, vol. 6, no. 3, pp. 1484–1492, 2014.
- [19] G. Hug-Glanzmann and G. Andersson, "Decentralized optimal power flow control for overlapping areas in power systems," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 327–336, 2009.
- [20] R. Baldick, B. H. Kim, C. Chase, and Y. Luo, "A fast distributed implementation of optimal power flow," *IEEE Trans. Power Syst.*, vol. 14, no. 3, pp. 858–864, 1999.
- [21] T. Erseghe, "Distributed optimal power flow using ADMM," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2370–2380, 2014.
- [22] B. Houska, J. Frasch, and M. Diehl, "An augmented Lagrangian based algorithm for distributed nonconvex optimization," *SIAM J. Optim.*, vol. 26, no. 2, pp. 1101–1127, 2016.
- [23] A. Engelmann, Y. Jiang, T. Mühlpfordt, B. Houska, and T. Faulwasser, "Toward distributed OPF using ALADIN," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 584–594, 2018.
- [24] J. Zhai, X. Dai, Y. Jiang, Y. Xue, V. Hagenmeyer, C. Jones, and X.-P. Zhang, "Distributed optimal power flow for VSC-MTDC meshed AC/DC grids using ALADIN," *IEEE Trans. Power Syst.*, pp. 1–1, 2022.

- [25] R. Bauer, X. Dai, and V. Hagenmeyer, "A shapley value-based distributed ac opf approach for redispatch congestion cost allocation," in *14th ACM e-Energy*, (Orlando, FL, USA), Association for Computing Machinery, 2023.
- [26] T. Mühlpfordt, X. Dai, A. Engelmann, and V. Hagenmeyer, "Distributed power flow and distributed optimization—formulation, solution, and open source implementation," *Sustain. Energy, Grids Netw.*, vol. 26, p. 100471, 2021.
- [27] X. Dai, Y. Cai, Y. Jiang, and V. Hagenmeyer, "Rapid scalable distributed power flow with open-source implementation," in *9th IFAC NECSYS*, 2022.
- [28] C. Berge, *Graphs and hypergraphs*. North-Holland Pub. Co., 1973.
- [29] I. Papastaiakoudis, M. Li, and I. Lestas, "Hypergraph based distributed quadratic optimization," in *25th International Symposium on Mathematical Theory of Networks and Systems*, 2022.
- [30] M. Bolla, "Spectra, euclidean representations and clusterings of hypergraphs," *Discrete Math.*, vol. 117, no. 1-3, pp. 19–39, 1993.
- [31] V. I. Voloshin, *Introduction to graph and hypergraph theory*. Nova Science Publishers, 2009.
- [32] A. J. Korsak, "On the question of uniqueness of stable load-flow solutions," *IEEE Trans. Power Appar. Syst.*, no. 3, pp. 1093–1100, 1972.
- [33] D. Mehta, D. K. Molzahn, and K. Turitsyn, "Recent advances in computational methods for the power flow equations," in *2016 American Control Conference (ACC)*, pp. 1753–1765, IEEE, 2016.
- [34] H. D. Nguyen and K. S. Turitsyn, "Appearance of multiple stable load flow solutions under power flow reversal conditions," in *2014 IEEE PES General Meeting— Conference & Exposition*, pp. 1–5, IEEE, 2014.
- [35] H. D. Nguyen and K. Turitsyn, "Voltage multistability and pulse emergency control for distribution system with power flow reversal," *IEEE Trans. Smart Grid.*, vol. 6, no. 6, pp. 2985–2996, 2015.
- [36] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.
- [37] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2010.
- [38] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.
- [39] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.
- [40] S. Frank and S. Rebennack, "An introduction to optimal power flow: Theory, formulation, and examples," *IIE transactions*, vol. 48, no. 12, pp. 1172–1197, 2016.

APPENDIX

A. Proof of Proposition 1

Based on [29], we can prove that the new iterate $(x^{k+1}, z^{k+1}, \lambda^{k+1})$ satisfies the KKT condition (21). By substituting (24) into (25a), we have

$$\lambda^{k+1} = \overline{B}^k M^k \bar{x} \quad (29)$$

with $M^k = I - E \left(E^\top \overline{B}^k E \right)^{-1} E^\top \overline{B}^k$. Consequently, we have

$$E^\top \lambda^{k+1} = E^\top \overline{B}^k M^k \bar{x} = 0. \quad (30)$$

This satisfies the dual feasibility (21b). By substituting (23) into (29), we have

$$\lambda^{k+1} = \overline{M}^k \bar{b}^k, \quad (31)$$

with $\overline{M}^k = \overline{B}^k M^k \left(\overline{B}^k \right)^{-1}$ and $\bar{b}^k = \overline{B}^k x^k - \bar{g}^k$. Accordingly, we can rewrite (25b) in a condensed form

$$\begin{aligned} x^{k+1} &= \left(\overline{B}^k \right)^{-1} \left(\overline{B}^k x^k - \bar{g}^k - \lambda^{k+1} \right) \\ &= \left(\overline{B}^k \right)^{-1} \left(\overline{B}^k x^k - \bar{g}^k - \overline{M}^k \bar{b}^k \right) \\ &= \left(\overline{B}^k \right)^{-1} \left(I - \overline{M}^k \right) \bar{b}^k \\ &= \left(I - M^k \right) \left(\overline{B}^k \right)^{-1} \bar{b}^k \\ &= E \left(E^\top \overline{B}^k E \right)^{-1} E^\top \bar{b}^k, \end{aligned} \quad (32)$$

and its common value

$$\begin{aligned} z^{k+1} &= \left(E^\top \overline{B}^k E \right)^{-1} E^\top \overline{B}^k E \left(E^\top \overline{B}^k E \right)^{-1} E^\top \bar{b}^k \\ &= \left(E^\top \overline{B}^k E \right)^{-1} E^\top \bar{b}^k = \bar{z}. \end{aligned} \quad (34)$$

Thereby, primal feasibility (21c) is satisfied by

$$x_\ell^{k+1} = E_\ell z^{k+1}, \ell \in \mathcal{R}. \quad (35)$$

Moreover, the condition (21a) is trivially satisfied due to (32). We have thus established the Theorem 1 by combining (30) (32) (35).

B. Proof of Theorem 1

The deviation between the Newton step p^N and the Gauss-Newton step p^{GN} can be written as

$$\begin{aligned} p^N - p^{GN} &= \left(B^k \right)^{-1} \left(B^k p^N + \nabla f^k \right) \\ &= \left(B^k \right)^{-1} \left(B^k - \nabla^2 f^k \right) p^N \\ &= - \left(B^k \right)^{-1} Q^k p^N \end{aligned} \quad (36)$$

As a result, we obtain the following inequality

$$\begin{aligned} \|\chi^k + p^{GN} - \chi^*\| &\leq \|\chi^k + p^N - \chi^*\| + \|p^N - p^{GN}\| \\ &\leq \omega_1 \|\chi^k - \chi^*\| + \omega_2 \|\chi^k - \chi^*\|^2, \end{aligned}$$

where

$$\omega_1 = s^k \cdot q^k \text{ and } \omega_2 = \hat{L}(s^k \cdot q^k + 1) \quad (37)$$

with bounded $s^k = \left\| \left(B^k \right)^{-1} \right\|$ and $q^k = \|Q^k\| = \mathcal{O}(\|\chi^k - \chi^*\|)$. The locally quadratic convergence rate of Algorithm 1 can be, thus, established [36].

C. Anonyms

Abbr.	Description
ADMM	Alternating Direction Method of Multipliers
ALADIN	Augmented Lagrangian based Alternating Direction Inexact Newton method
APP	Auxiliary Problem Principle
HDQ	hypergraph-based distributed quadratic optimization algorithm
HDSQP	hypergraph-based distributed sequential quadratic programming
KKT	Karush–Kuhn–Tucker
LICQ	linear independence constraint qualification
NLP	Nonlinear Programming
OCD	Optimality Condition Decomposition
PF	power flow
DSOs	distribution system operators
TSOs	transmission system operators