PLOS ONE

RESEARCH ARTICLE

Distributed optimal power flow

HyungSeon Oh*

Department of Electrical and Computer Engineering, United States Naval Academy, Annapolis, Maryland, United States of America

* hoh@usna.edu

Abstract

Objective

The objectives of this paper are to 1) construct a new network model compatible with distributed computation, 2) construct the full optimal power flow (OPF) in a distributed fashion so that an effective, non-inferior solution can be found, and 3) develop a scalable algorithm that guarantees the convergence to a local minimum.

Existing challenges

Due to the nonconvexity of the problem, the search for a solution to OPF problems is not scalable, which makes the OPF highly limited for the system operation of large-scale real-world power grids—"the curse of dimensionality". The recent attempts at distributed computation aim for a scalable and efficient algorithm by reducing the computational cost per iteration in exchange of increased communication costs.

Motivation

A new network model allows for efficient computation without increasing communication costs. With the network model, recent advancements in distributed computation make it possible to develop an efficient and scalable algorithm suitable for large-scale OPF optimizations.

Methods

We propose a new network model in which all nodes are directly connected to the center node to keep the communication costs manageable. Based on the network model, we suggest a nodal distributed algorithm and direct communication to all nodes through the center node. We demonstrate that the suggested algorithm converges to a local minimum rather than a point, satisfying the first optimality condition.

Results

The proposed algorithm identifies solutions to OPF problems in various IEEE model systems. The solutions are identical to those using a centrally optimized and heuristic approach. The computation time at each node does not depend on the system size, and N_{iter} does not increase significantly with the system size.

Citation: Oh H (2021) Distributed optimal power flow. PLoS ONE 16(6): e0251948. https://doi.org/10.1371/journal.pone.0251948

Editor: Ping He, Jinan University, China, HONG KONG

Received: January 4, 2021

Accepted: May 4, 2021

Published: June 18, 2021

Data Availability Statement: Relevant data are publicly available through the Texas A&M Electric Grid Test Case Repository: https://electricgrids.engr.tamu.edu/electric-grid-test-cases/.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Conclusion

Our proposed network model is a star network for maintaining the shortest node-to-node distances to allow a linear information exchange. The proposed algorithm guarantees the convergence to a local minimum rather than a maximum or a saddle point, and it maintains computational efficiency for a large-scale OPF, scalable algorithm.

Introduction

In modern societies, demand for electricity is expected to be satisfied continuously via controllable generation technologies. An event is a situation in which the demand is not fulfilled. Ten-in-one, a widely used reliability criterion for events, means that an event should occur just once in a 10-year span. To meet this standard, system operators schedule the generation portfolio and the grid systems in advance. For example, a day-ahead unit commitment determines the 24-hourly dispatches, along with unit commitment decisions, to meet varying hourly demands. For each hour, the demand profiles are assumed to be constant, which defines the process's steady-state operation. In the absence of an unexpected disturbance, stochastic hourly demand is the unique source of uncertainty in traditional power system operation. Over the last decade or so, renewable energy resources and smart grid technologies have been integrated into systems to improve energy efficiency and reduce greenhouse gas emissions. This integration has introduced uncertainty into the operation of power systems, presenting a new challenge. If high-precision forecasting could be introduced to estimate future energy resources and control demand, existing operations' tools would remain useful, assuming they could be integrated into the expected effective demand (≡ demand–expected demand reduction–expected renewable energy resources). Unfortunately, even though the precision of forecasting tools has improved, the errors in their long-term forecasts, for a day ahead, for example, are not yet sufficiently small for reliable operation. Frequent decisions are a potential way to accommodate the uncertainty. For example, a day-ahead 24-hour unit commitment (UC) decision is made once in a daily cycle. If the errors in 2-hour ahead forecasts are small enough, then the UC decision with a forecast every 2 hours would still be a reliable tool for the power system's operation. The computational capability to support such decisions plays a key role in this process.

Optimal power flow (OPF) is a backbone in the steady-state operation of power systems. The characteristics of OPF are highly nonlinear and nonconvex. The computational complexity associated with these characteristics makes power flow (PF) analysis a non-deterministic polynomial-time (*NP*)-hard problem [1]. In most operational practices, a linear approximation of OPF, namely, direct current (DC) OPF, is pursued. Although easy to solve, DC OPF does not address voltage problems, losses, and the dispatch of reactive power generation. Due to these issues, DC OPF may not be feasible. To address the problem correctly, it is ideal to aim for a nonlinear and nonconvex OPF. In addition to the nonconvex nature of the full OPF, uncertainties increase the number of variables in traditional, central decision-making processes. Therefore, frequent but short-term decisions concerning large-scale power systems can be challenging. With the recent advancements in hardware in multi-core machines, distributed computation becomes an attractive approach for enhancing computational efficiency. An exemplary area in power system analysis concerns the use of distributed computation for OPF. Motivated readers can find information related to distributed approaches to solving OPF problems [2, 3]. Within distributed computation, the alternating direction method of multipliers

(ADMM) has gained popularity due to its straightforward implementation and its provable convergence (if the original problem is convex). For a linear DC OPF, the ADMM approach can result in successful convergence to a global solution [4–6]. A full OPF problem on a radial network can be relaxed to a convex semidefinite programming (SDP) problem, and the relaxed problem is exact if optimal power injections lie in a region where the voltage upper bounds do not bind [7, 8]. Several studies have used the ADMM approach to solve OPF problems in a radial network [9–11]. OPF problems usually involve the operation of mesh transmission systems, but an SDP solution for a mesh network may not be physically meaningful [2].

A nodal OPF would be the most intuitive approach to extending a central OPF in a distributed fashion. In the nodal OPF [12], the information exchange among the nearest neighbors leads to high communication costs. The maximum node-to-node distance (also termed the path length (PL) [13]) plays a key role in the communication costs. The convergence tends to be very slow due to the contaminated information received from local decisions during communications. To the best of our knowledge, there has been no report of the successful convergence of this approach for any mesh networks.

The PL can be reduced when a clustering approach is undertaken for a mesh network via the partitioning of a system into multiple subsystems [14–19]. Two adjacent subsystems share some nodes and branches between the nodes; thus, the PL is small, keeping the communication costs manageable. The primary and the dual variables at the shared nodes and branches are constrained equally. This approach can be efficient if the shared nodes adequately represent the other nodes in the same subsystem. Several studies have proposed an efficient algorithm for partitioning a system so that the ADMM converges to a solution [16, 17, 19, 20]. In contrast to a study by Sun, Phan, and Ghosh [12], the flow constraints can be integrated for the lines of intra-subsystems. In two studies, Erseghe [14, 15] integrated the flow limits of the lines of inter-subsystems by redefining the subsystems to overlap the lines. Guo, Hug, and Tonguz [16] report that the inclusion of the limit still yields a solution. However, the approach has several shortcomings that contribute to computational inefficiency: 1) the low quality of the solution, 2) the need for a warm starting point for convergence, and 3) the communication costs. In addition, the convergence behavior is not reported, so it is not possible to discuss the computational efficiency. The solutions presented in several studies [14–17] are low-voltage, inferior solutions due to increased losses. Engelmann et al. [18] added a significantly large term regarding reactive power injections that affects the optimality conditions and, as a result, the distributed problem is different from the original one. The necessity of a solution for the nonconvex PF as a starting point increases the computational costs. Even though the PL decreases in comparison to the nodal OPF in the study by Sun, Phan, and Ghosh [12], the communication costs increase the computational costs due to the tradeoff discussed by Guo, Hug, and Tonguz [16]. These shortcomings make the benefit of distributed computation questionable. The distributed SDP approach by Madani, Kalbat, and Lavaei [19] yields the global (and therefore identical) solution to the central SDP approximation to the OPF problem because the problem is convex, but the solution may not be physically feasible. In addition to the computational inefficiency, there is no approach that can theoretically yield an optimal balance between the computational cost for one subsystem and the communication costs of the subsystems. Another study by Guo, Hug, and Tonguz [20] proposes a heuristic approach for selecting subsystems, but it does not yield a unique choice because its initialization is based on the local solution for the nonconvex OPF. In several studies [14, 17, 18, 20], a positive correlation was observed between the system size and the number of nodes in the largest subsystems (see Fig 1). The PL depends linearly on the number of subsystems n, which depends inversely on the size of the largest subsystem (N_{sub}), i.e., $n \geq Nb/N_{sub}$, where the equality holds when the sizes of the subsystems are uniform. Fig 1 indicates a positive correlation between N_{sub} and Nb.

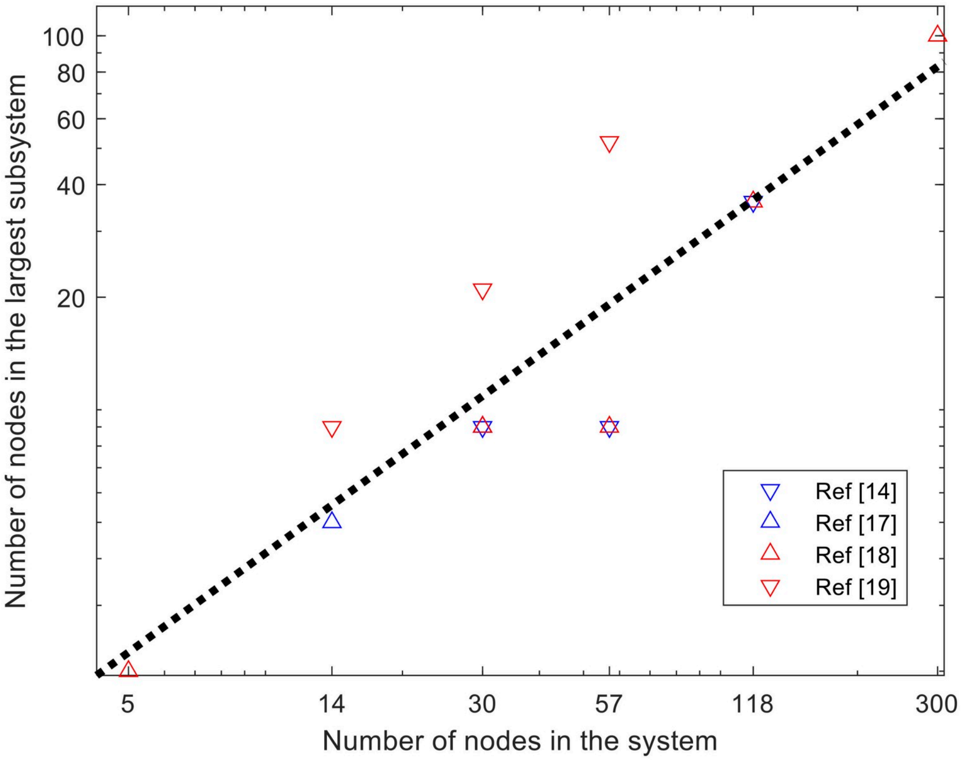

Fig 1. The largest number of nodes in the partitioned subsystems of various power system test cases reported in the literature [14, 17–19]**.**

https://doi.org/10.1371/journal.pone.0251948.g001

The dotted line indicates that the best linear fit for the relationship is $N_{sub} \propto Nb^{0.88}$. If the optimization problem is solved by SDP, the computation time is found in $\vartheta\left(N_{sub}^{3}\right)$ [21] that yields a computation time that is proportional to $\vartheta(Nb^{2.64})$. In addition, the approaches require significant communication costs as well. Therefore, the overall computation cost is much higher than that of the central optimization. In addition to the computational efficiency, it is not guaranteed that the voltages at the boundary of each subsystem represent the voltages of other nodes inside the same subsystem correctly. If they do not, convergences may not be observed because the information exchange is limited to the boundary buses. The slow convergences and/or non-monotonous convergences reported in previous studies [14–17] indicate the insufficient representativeness of the boundary buses. A relatively fast convergence is reported by Engelmann et al. [18] in exchange of per-step communication costs by sharing the sensitivities in addition to the local primal variables. The increase in the communication cost is found in $\vartheta\left(\sum_{i} n_i^2\right)$ where n_i is the dimension of the local gradient at the i^{th} group. Although the progress at each iteration is faster than those of other ADMM approaches, the communication cost itself is much higher than the total computation costs of the nonconvex heuristic solvers or of the SDP solvers. As a result, although these studies are worth exploring, we conclude that the aggregation approach is not practical in terms of computational efficiency and the inferior quality of the solutions for a scalable algorithm due to the tradeoff issue, nonconvexity, and the modeling problem. A new network model for OPF is necessary for the distributed computation.

The contributions of this paper are 1) a new network model that yields direct communication among nodes regardless of the system size, 2) a distributed, fast, and efficient algorithm to solve highly nonlinear and nonconvex OPF problems, and 3) a scalable algorithm that guarantees convergence to a local minimizer. The paper is organized as follows: the theory section proposes a new network model designed for distributed computation and presents an algorithm to solve a full OPF; the next section describes the details of the implementation of the proposed algorithm; the results and discussion present the results for the OPF and comparisons with those from other studies; the following section provides conclusions and research directions for further improving the computational efficiency; and the appendices sketch the proofs of the ranks of the matrices associated with OPF problems and of the convergence.

Theory

Proposed network model and algorithm

We propose a new network model for a nodal OPF to keep communication costs manageable regardless of the system size. For this purpose, the desired properties of the model are as follows:

1. The model must be compatible with PF studies for which Kirchhoff's laws and voltage magnitudes are well defined.
2. Each node is a short distance away from the rest of the nodes to minimize the communication costs.
3. The voltages at a node and those at the rest of the nodes are linearly related.

In the power flow studies, such as PF, OPF, state estimation, and probabilistic PF, voltages are the variables. The constraints in the studies consist of the power flows and injections, as well as the voltage magnitudes in terms of voltages. In the Cartesian coordinate system, the power flows and injections are quadratic in voltage. For example, the power flow over the line connecting Nodes *i* and *j* at *i* is $\boldsymbol{f}_{i-j}^{i} = \boldsymbol{v}_i \boldsymbol{i}_{i-j}^{*} = v^T \boldsymbol{F}_{i-j}^{i} v$, and the power injection at Node *i* is $\boldsymbol{g}_i = p_i + \mathbf{j} q_i = v^T \boldsymbol{S}_i v$ where the quantities sandwiched between voltages are in *2Nb*-by-*2Nb*; $\boldsymbol{F}_{i-j}^{i} \triangleq \boldsymbol{J} e_i e_{i-j}^T \boldsymbol{Y}_{br}^{*} \boldsymbol{J}^H$ where $e_{i\text{-}j}$ is a vector with the cardinality of 2*Nl* of which the element corresponding to the flow over *i-j* is 1, and all other elements are zeros, $\boldsymbol{S}_i \triangleq \boldsymbol{J} e_i e_i^T \boldsymbol{Y}_{bus}^{*} \boldsymbol{J}^H$; and the superscript *H* is the conjugate transpose. The matrices $\boldsymbol{F}_{i-j}^{i}$ and $\boldsymbol{S}_i$ have two nonzero rows at *i* and *Nb*+*i* rows due to $\boldsymbol{J}\, e_i$.

Claim 1: The matrices associated with power flows and power injections are all of rank 4.

Claim 2: The matrices associated with the squares of the voltage magnitudes have rank 2.

See S1 Appendix for the proof of the claims.

For a real-valued symmetric rank-4 matrix M_j, a real-valued eigen pair λ and u exist such that $M_j = \sum_{k=1}^{4} \lambda_{jk} u_{jk} u_{jk}^T = \underbrace{\left[\sqrt{\lambda_{j1}} u_{j1} \;\; \sqrt{\lambda_{j2}} u_{j2} \;\; \sqrt{\lambda_{j3}} u_{j3} \;\; \sqrt{\lambda_{j4}} u_{j4}\right] II_j \Phi_j^T}_{\Phi_j}$. Because M_j is a rank 4 matrix, the number of columns in Φ_j and the number of nonzero diagonal elements in II_j are both 4. Here, the terms $\Phi_{\breve{S}_j}$, $\Phi_{\hat{S}_j}$, $\Phi_{\breve{F}_{j-l}^{j}}$, and $\Phi_{\hat{F}_{j-l}^{j}}$: $\Phi_{\breve{S}_j}$ refer to the matrices associated with the real power injection at Node *j* $\left(\breve{S}_j \triangleq \mathrm{Re}\left(\boldsymbol{S}_j\right) = \Phi_{\breve{S}_j} II_j^{\breve{S}_j} \Phi_{\breve{S}_j}^T\right)$; $\Phi_{\hat{S}_j}$ is associated with reactive power injection at Node *j* $\left(\bar{S}_j \triangleq \mathrm{Im}\left(\boldsymbol{S}_j\right) = \Phi_{\bar{S}_j} II_j^{\bar{S}_j} \Phi_{\bar{S}_j}^T\right)$; $\Phi_{\breve{F}_{j-l}^{j}}$ is associated with real power flow over the line *j-l* at the side of Node *j* $\left(\breve{F}_{j-l}^{j} = \Phi_{\breve{F}_{j-l}^{j}} II_{j-l}^{\breve{F}_{j-l}^{j}} \Phi_{\breve{F}_{j-l}^{j}}^T\right)$; and $\Phi_{\hat{F}_{j-l}^{j}}$ is associated with the reactive power flow over the line *j*-*l* at the side of Node *j* $\left(\bar{F}^{j}_{j-l} = \Phi_{\bar{F}^{j}_{j-l}} II^{\bar{F}^{j}_{j-l}}_{j-l} \Phi^{T}_{\bar{F}^{j}_{j-l}}\right)$. Note that the eigenpairs are dependent solely on the system parameters (not on voltages). With the eigenpairs, the quantities of interest in the PF are the real power injection at Node *j* p_j, the reactive power injection at Node *j* q_j, the real power flow over a line *j*-*l* at Node *j* $\breve{f}^{j}_{j-l}$, the reactive power flow over a line *j*-*l* at Node *j* $\bar{f}^{j}_{j-l}$, and the squared voltage's magnitude at Node *j* E_j. There are quadratic relationships among them:

$$p_j\left(= \breve{g}_j - \breve{d}_j\right) = \alpha_j^T II_j^{\breve{S}_j} \alpha_j;\ \ q_j\left(= \bar{g}_j - \bar{d}_j\right) = \beta_j^T II_j^{\bar{S}_j} \beta_j;\ \ \breve{f}^{j}_{j-l} = \gamma^{T}_{j-l,j} II^{\breve{F}^{j}_{j-l}}_{j-l,j} \gamma_{j-l,j};\ \ \bar{f}^{j}_{j-l} = \delta^{T}_{j-l,j} II^{\bar{F}^{j}_{j-l}}_{j-l,j} \delta_{j-l,j};\ \ E_j = \omega_j^T \omega_j (1)$$

Eq (1) expresses Kirchhoff's laws and the voltage magnitudes. The nodal variables α_j, β_j, $\gamma_{j\text{-}l,j}$, $\delta_{j\text{-}l,j}$, and ω_j are defined as follows: the α_j are the voltages projected onto the eigenspaces spanned by the real power injection, and the β_j cover the reactive power injection, the $\gamma_{j\text{-}l,j}$ cover the real power flow over *j*-*l* at Node *j*, the $\delta_{j\text{-}l,j}$ cover the reactive power flow over *j*-*l* at Node *j*, and the ω_j cover the voltage magnitudes at the j^{th} node in the same manner. Although there is one each of α_j, β_j, and ω_j at each node, the nl_j of $\gamma_{j\text{-}l,j}$ and $\delta_{j\text{-}l,j}$ are defined such that nl_j represents the number of branches that are directly connected to Node *j* (i.e., $l = 1, 2, \ldots, nl_j$). It is worth noting that the nodal variables are linearly associated with the voltages (the definitions of the nodal variables in Eq (1)), which indicates that PL = 1. Therefore, the nodal variables satisfy all the desired properties of the new network model.

Next, we let nl_j be the number of lines connected to the j^{th} node. Then, the dimensions of α, β, γ, δ, and ω are 4, 4, $4nl_j$, $4nl_j$, and 2, respectively. Note that their dimensions do not depend on the system size. Let μ_j be a nodal variable that integrates local generation variables as follows:

$$\mu_j = \left[x_j^T\ xQ_j^T\ \mu_{end}\right]^T;\ \ x_j^{8nl_j+10} = \left[\alpha_j^T\ \beta_j^T\ \gamma_{jl}^T\ \delta_{jl}^T\ \omega_j^T\right]^T;\ \ xQ_j^{2nl_j+2ng_j} = \left[\left(\breve{f}^{j}_{j-l}\right)^T\ \left(\bar{f}^{j}_{j-l}\right)^T\ p_j^T\ q_j^T\right]^T$$

where $\mu_{end} = 1$. The generalized PF formulation is listed in previous work [22]. Note that the cardinality of the variable is fixed—independent of the system size. Therefore, the new network model yields: 1) a fixed number of variables, and 2) a PL of 1.

Proposed network model: A star grid with two channels

In the definition of the local variable x_j, it is noticeable that there are three variable types: 1) power injection and flow variables α, β, γ, and δ, 2) the voltage magnitude variable ω, and 3) quadratic variables, ${}_{Q}\,x_j$ comprising power flows $\left(\breve{f}^{j}_{j-k}, \bar{f}^{j}_{j-k}\right)$ and power generations ($\boldsymbol{g}_j$). The first two types of variables are linear, and the third type is quadratic, in terms of the voltages. Once x_j (α, β, γ, and δ) is determined, it is straightforward to find the power generations at Node *j* with Eq (1) (i.e., x_j defines the feasible region of μ_j). The voltage v_p can be reconstructed from α, β, γ, and δ. The communication path used to collect their values is termed the power channel. Here, ω is directly associated with the voltage magnitudes, and the values are reported through another communication path termed the voltage channel to update voltages v_m. Although the voltages v_p and v_m should be identical, conditions are relaxed so that they can be different.

Fig 2 illustrates an example of the traditional network model (left) and the proposed network model (right) for a modified 4-bus system that has branches connecting 1–2, 1–3, 1–4, 2–4, and 3–4. In the proposed model, all the nodes are connected to the center node (black dot at the center) through the power channel (red lines) and the voltage channel (green lines), and

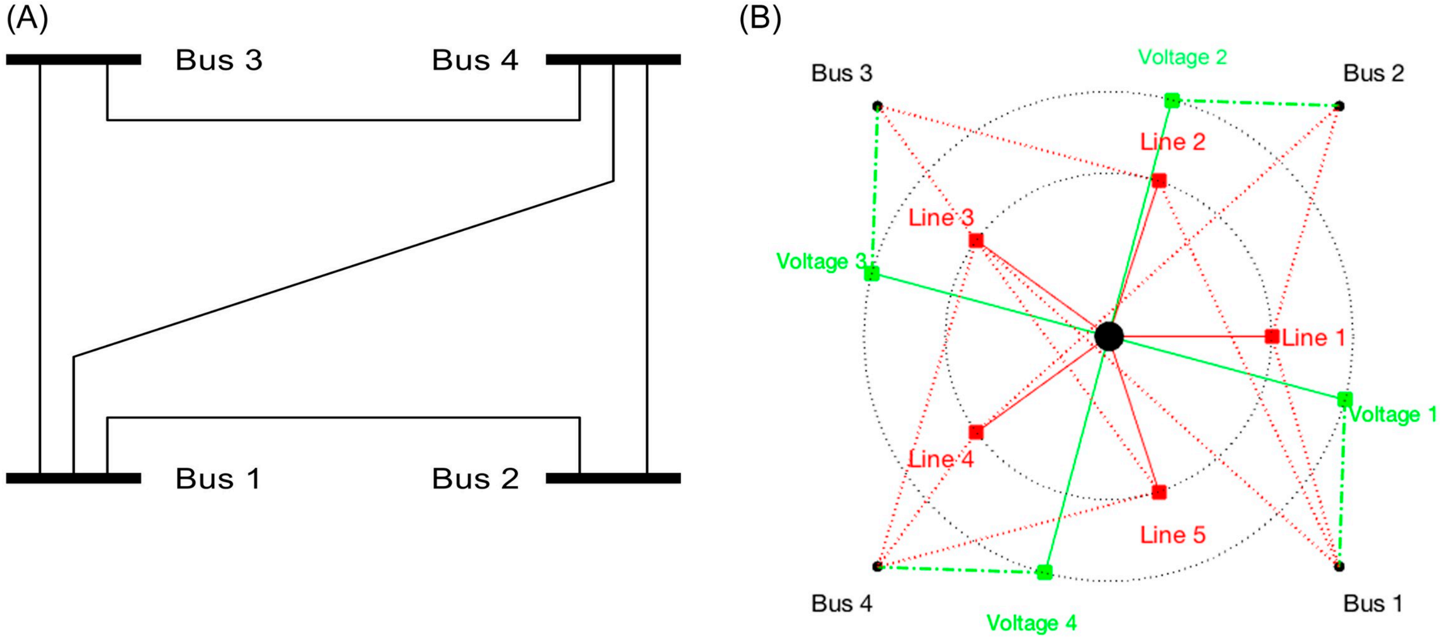

Fig 2. A star and linear network for a modified IEEE-4 bus system. The traditional network model is shown on the left (A) and the proposed network model is presented on the right (B). Red lines indicate the power channel, and green lines are the voltage channel.

https://doi.org/10.1371/journal.pone.0251948.g002

all the nodes are of distance 1 from the center (PL = 1). The network is a complete bipartite graph of order 2 with a maximum-diameter star network. All communications between the variables take place in local nodes (α, β, γ, δ, and ω), and the center node is linear in terms of voltages.

Nodal OPF with nodal variables and its convex relaxation

Even though we describe the OPF problem in this paper, the same framework is applied to any other problem, including PF, OPF, state estimation, and probabilistic PF problems, because they have unified formulas [22]. The central OPF formulation as a nonconvex, quadratically constrained quadratic problem is given in Eq (3), its distributed optimization problem (nodal OPF) at each node *I* is given in Eq (3A), and the nodal OPF is formulated in terms of an ADMM algorithm in Eq (3B).

$$\min_{v,g} \; \breve{g}^T diag(a)\breve{g} + 2b^T\breve{g}, \quad g = [\breve{g}^T \quad \bar{g}^T]^T \tag{2}$$

$$\text{s.t.} \quad \begin{array}{ll}
1. & v^T\breve{S}_j v - L_g^T\breve{g} + \breve{d}_j = 0,\; v^T\bar{S}_j v - L_g^T\bar{g} + \bar{d}_j = 0 \;\; \forall j \in B \\
2. & v^T\breve{F}_{i-j}^i v - \breve{f}_{i-j}^i = 0,\; v^T\bar{F}_{i-j}^i v - \bar{f}_{i-j}^i = 0 \;\; \forall j \in Br\{i\} \\
3. & f^T\left(e_l e_l^T + e_{l+Nl}e_{l+Nl}^T\right)f - cap_l^2 \le 0,\; f = \left[\breve{f}^T \; \bar{f}^T\right]^T \;\; \forall l \in Br \\
4. & g^T\left(e_m e_m^T\right)g - \left(p_m^{\max} + p_m^{\min}\right)e_m^T g + p_m^{\max}\cdot p_m^{\min} \le 0 \;\; \forall m \in G \\
5. & g^T\left(e_{m+Ng}e_{m+Ng}^T\right)g - \left(q_m^{\max} + q_m^{\min}\right)e_{m+Ng}^T g + q_m^{\max}\cdot q_m^{\min} \le 0 \\
6. & v^T\left(e_j e_j^T + e_{j+Nb}e_{j+Nb}^T\right)v - E_j^{\max} \le 0 \;\; \forall j \in B \\
7. & -v^T\left(e_j e_j^T + e_{j+Nb}e_{j+Nb}^T\right)v + E_j^{\min} \le 0 \;\; \forall j \in B
\end{array}$$

$$\min_{\mu} \; L(\mu) = \min_{\mu} \; \sum_i L_i(\mu_i) = \min_{\mu} \sum_i \mu_i^T M_{ob}^i \mu_i \tag{3A}$$

$$\text{where } M_{ob}^i = \begin{bmatrix} 0 & 0^{(8nl_i+10)\times 2} & 0 \\ 0 & diag(a_i \;\; 0) & (0 \;\; b_i)^T \\ 0 & (0 \;\; b_i) & 0 \end{bmatrix} \text{ and } \mu_i = \begin{bmatrix} x_i \\ xQ_i(x_i) \\ \mu_{i,end} \end{bmatrix}$$

$$\text{s.t.} \left\{ \begin{array}{ll} 1.\; \mu_i^T Re(II_i^{S_{inj}})\mu_i + \breve{d}_i = 0; \mu_i^T Im(II_i^{S_{inj}})\mu_i + \bar{d}_i = 0; & \text{real and reactive power balance} \\ 2.\; \mu_i^T Re(II_{i-l,i}^{F_{flow}})\mu_i = 0; \mu_i^T Im(II_{i-l,i}^{F_{flow}})\mu_i = 0; \; j \in Br\{i\}; & \text{definitions for the real and the reactive power flows as shown in 2 (2)} \\ 3.\; \mu_i^T II_{i-l,i}^{F_{flow}^{\max}} \mu_i - cap_l^2 \le 0; & \text{thermal flow limit} \\ 4.\; \mu_i^T Re\left(II_{m,i}^G\right)\mu_i + p_{m,i}^{\max} p_{m,i}^{\min} \le 0; \mu_i^T Im\left(II_{m,i}^G\right)\mu_i + q_{m,i}^{\max} q_{m,i}^{\min} \le 0 & \text{real and reactive power generation limits} \\ 5.\; E_i^{\min} \le \mu_i^T II_i^{E_i} \mu_i \le E_i^{\max}; & \text{limits for the voltage magnitudes} \\ 6.\; A_j^T \mu_i - \Phi_L^T v_L - \Phi_M^T v_M = 0 \leftrightarrow x_i - \Phi_L^T v_L - \Phi_M^T v_M = 0; & \text{nodal variables in terms of the central voltages} \end{array} \right.$$

where A is a block matrix to collect an element

$$\xi_j = A_\xi^T \mu_j; \mu_i^{nx\times 1} = \left[\alpha_i^{4\times 1};\; \beta_i^{4\times 1};\; \gamma_{i-l,i}^{4nl_i\times 1};\; \delta_{i-l,i}^{4nl_i\times 1};\; \omega_i^{2\times 1};\; \breve{f}_{i-l,i}^{nl_i\times 1};\; \bar{f}_{i-l,i}^{nl_i\times 1};\; g_i^{2ng_i\times 1};\; \mu_{i,end}^{1\times 1}\right];$$

$$\left(\alpha_i^T \;\; \beta_i^T\right)^T = A_{\alpha\beta}^T \mu_i; \left(\gamma_{i-l,i}^T \;\; \delta_{i-l,i}^T\right) = A_{\gamma\delta_l}^T \mu_i; \left(\breve{f}_{i-l,i} \; \bar{f}_{i-l,i}\right)^T = A_{fl}^T \mu_i; \left(g_i^{pT} \quad g_i^{qT} \quad \mu_{i,end}\right) = A_{gm}^T \mu_i;$$

$$\omega_i = A_\omega^T \mu_i; A_j = \left[A_{\alpha\beta} \quad A_{\gamma\delta_1} \quad \cdots \quad A_{\gamma\delta_{Br\{j\}}}\right]; II_{i-l,i}^{F_{flow}^{\max}} \triangleq A_{fl} A_{fl}^T; II_i^{E_i} \triangleq A_\omega A_\omega^T;$$

$$\Phi_L = \left[\Phi_{\breve{S}_i} \Phi_{\bar{S}_i} \Phi_{\breve{F}_{i-1}^i} \; \Phi_{\bar{F}_{i-1}^i} \cdots \Phi_{\breve{F}_{i-Br\{i\}}^i} \; \Phi_{\bar{F}_{i-Br\{i\}}^i}\right]; \Phi_M = e_j + e_{j+Nb};$$

$$II_j^{S_{inj}} \triangleq A_{\alpha\beta} blockdiagonal \begin{pmatrix} II_j^{\breve{S}_j} \\ \mathbf{j} II_j^{\bar{S}_j} \end{pmatrix} A_{\alpha\beta}^T - \tfrac{1}{2} A_g \begin{bmatrix} 0 & 0 & 1^{ng_j} \\ 0 & 0 & \mathbf{j}1 \\ 1^T & \mathbf{j}1^T & 0 \end{bmatrix} A_g^T;$$

$$II_{i-l,i}^{F_{flow}} \triangleq A_{\gamma\delta_l} \begin{bmatrix} II_{i-l,i}^{\breve{F}_{i-j}^i} & 0 \\ 0 & \mathbf{j} II_{i-l,i}^{\bar{F}_{i-j}^i} \end{bmatrix} A_{\gamma\delta_l}^T - \tfrac{1}{2} A_{fl} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & \mathbf{j}1 \\ 1^T & \mathbf{j}1^T & 0 \end{bmatrix} A_{fl}^T; \text{ and}$$

$$II_{m,i}^G \triangleq A_{gm} \begin{bmatrix} 1 & 0 & -p_{m,i}^{av} \\ 0 & \mathbf{j} & -\mathbf{j} q_{m,i}^{av} \\ -p_{m,i}^{av} & -\mathbf{j} q_{m,i}^{av} & 0 \end{bmatrix} A_{gm}^T; p_{m,i}^{av} = \left(p_{m,i}^{\max} + p_{m,i}^{\min}\right)/2, \text{ and } q_{m,i}^{av} = \left(q_{m,i}^{\max} + q_{m,i}^{\min}\right)/2.$$

$$\min_{x_i \in X_i} \; \sum_i w_i(x_i, y, z_i) = \min_{x_i \in X_i} \; \sum_i [f_i(x_i) + g_i(x_i, y, z_i)] \tag{3B}$$

where $y = \left(v_L^T \quad v_M^T\right)^T, f_j\left(x_j\right) \triangleq \min_{x_i \in X_i} \sum_i x_i^T M_{ob}^i x_i, g\left(x_j, y, z_j\right) = \frac{\rho_j}{2} \|x_j - \Phi_j^T y\|^2 + z_j^T\left(x_j - \Phi_j^T y\right),$ or equivalently $g\left(x_j, y, z_j\right) = \frac{\rho_j}{2} \|x_j - \Phi_j^T y + z_j/\rho_j\|^2 - \frac{1}{2\rho_j} \|z_j\|^2$; and X_j is the column space of Φ_{j}. at Node j defined by the constraints listed in 1–5 in (3A).

Note that all *A's* are full column rank matrices that collect relevant parts from μ_i, and that Φ_i includes the linear relationships between the local variable x_i and the central variable v and y. Using the definition of the function f_j, the following observations are made: 1) f_j is a nonconvex, but smooth, function that is C^1 on an open set containing X_j (defined by the column space of Φ_j), and ∇f_j is Lipschitz continuous on X_j; 2) X_j is nonempty, closed, and convex; and 3) w_j is coercive.

Even though Eqs (3A) and (3B) have low cardinalities in the decision variables due to the nonconvex nature of the problem, the uniqueness and existence of the solution are not guaranteed. To address the complexity issue, a surrogate function h_j approximating Eq (3B) is defined as follows:

$$H(x, y, z) = \sum_j h_j\left(x_j, y, z_j\right) = \sum_j \left[u_j\left(x_j\right) + g_j\left(x_j, y, z_j\right)\right] \text{ where } u_j\left(x_j\right) = \text{SDP of } f_j\left(x_j\right) \quad (4)$$

Because u_j is the SDP relaxation of f_j, it is convex, Lipchitz continuous, and continuously differentiable on X_j. Note that u_j relaxes the rank constraint only; therefore, the first derivatives of u_j and of f_j regarding x_j are identical (i.e., $\nabla_{x_j} f_j\left(x_j\right) = \nabla_{x_j} u_j\left(x_j\right)$). Because the SDP is convex, the solution at the k^{th} iteration is uniquely determined by $\tilde{x}_j^{k+1} \triangleq \underset{x_j \in X_j}{\arg\min}\, h_j\left(x_j, y^k, z_j^k\right)$ at given y^k and $z_j{}^k$. Note that the relaxed nodal OPF is convex and has a small number of variables; therefore, its computational complexity remains manageable.

Properties of u_j, f_j, g_j, h_j, w_j. and $\tilde{x}_j$

The nodal problem f_j is C^∞ smooth, but nonconvex, and prox-regular at x_j relative to f_j^∇ if $x_j \in \text{dom}(f_j)$, $f_j^\nabla \in \partial f_j\left(x_j\right)$. Further, whenever $\|\bar{x}_j - x_j\| < \varepsilon_j^{prox}$ and $\|\hat{x}_j - x_j\| < \varepsilon_j^{prox}$, there exists ε_j^{prox}, $r_j^f > 0$: $f1) f_j\left(\bar{x}_j\right) \geq f_j\left(\hat{x}_j\right) + \left(f_j^\nabla\right)^T \left(\bar{x}_j - \hat{x}_j\right) - \frac{r_j^{\nabla f}}{2}\|\bar{x}_j - \hat{x}_j\|^2$; and $f2) f_j\left(x_j^{k+1}\right) \leq f_j\left(x_j^k\right) + \nabla_{x_j} f_j\left(x_j^k\right)^T \left(x_j^{k+1} - x_j^k\right) + \frac{d_j^f}{2}\|x_j^{k+1} - x_j^k\|^2$. Statement (*f*2) holds by the Descent Lemma [23] with the properties of g_j and of h_j described below. Its SDP relaxation u_j is strongly convex, Lipschitz continuous, and prox-regular, and its first derivative equals that of f_j (i.e.:$u1) \quad u_j\left(\bar{x}_j\right) \geq u_j\left(x_j\right) + \left[\nabla_{x_j} u_j\left(x_j\right)\right]^T \left(\bar{x}_j - x_j\right) + \frac{c_j^u}{2}\|\bar{x}_j - x_j\|^2$ $u2) \quad u_j\left(\bar{x}_j\right) \leq u_j\left(x_j\right) + L_j^u\|\bar{x}_j - x_j\|$; and $u3) \quad \nabla_{x_j} u_j\left(x_j\right) = \nabla_{x_j} f_j\left(x_j\right)$. Whereas the variables considered in f_j and u_j are all local primal variables only, ADMM constraint g_j depends on the central variable y and the multipliers z. With the properties of u_j, f_j, and g_j, w_j is C^∞ smooth, but nonconvex, and prox-regular, whereas h_j is strongly convex and Lipschitz continuous. Because all X_j are bounded sets, w_j and h_j are coercive and lower-bounded over X_j. Because $\tilde{x}_j^{k+1}$ is the optimizer of convex h_j (i.e., $\tilde{x}_j^{k+1} \triangleq \underset{x_j \in \tilde{X}_j;\ x_j \in X_j}{\arg\min}\ h_j\left(x_j, y^k, z_j^k\right)$), the following holds:

$$x1) \quad \nabla_{x_j} u_j\left(\tilde{x}_j^{k+1}\right) + z_j^k + \rho_j\left(x_j - \Phi_j^T y\right) = 0 \text{ and}$$

$$x2) \quad h_j\left(\tilde{x}_j^{k+1}, y^k, z_j^k\right) - h_j\left(x_j, y^k, z_j^k\right) \leq 0 \ \forall x_j \in X_j.$$

Proposed algorithm with provable convergence

Overview of proposed algorithm. Even though the number of variables in the nodal OPF is much smaller than that in the original problem, the complexity of the problem makes it difficult to solve the nodal problem due to its nonconvex nature. To manage the nonconvex nature of the problem, the nonconvex components are relaxed. Once identified, the relaxed solution is mapped onto the feasible region, to the closest point in the feasible region from the identified solution. There are three cases: 1) the convexified problem is not feasible, 2) the convexified problem is feasible, but the solution to the problem is too far from the feasible region, and 3) the solution to the convexified problem is sufficiently close. Only if the mapped point is close enough, the nodal variables are updated and fed to the center node to update voltages.

Proposed algorithm. We propose the following algorithm to solve w_j in conjunction with h_j in Eq 5:

```
Distributed, Regulated, Optimally-Homogeneous, and Scalable (DROHS) Algorithm
1. Set $k = 0$, initialize all parameters, such as $\Delta^k \in (0,1)$, $y^k$, $\rho_j$, $\left\{z_j^k\right\} \in Null\left(\Phi_j\right)$.
2. If $x_j^k$ satisfies the termination criteria, terminate the algorithm.
3. Solve the local problems, $\tilde{x}_j^{k+1} \triangleq \arg\min_{x_j \in X_j} h_j\left(x_j, y^k, z_j^k\right)$.
4. Depending on the solution $\tilde{x}_j^{k+1}$, determine $\zeta_j^{k+1}$ (Fig 3)
    • Case 1: $\tilde{x}_j^{k+1}$ is in the real space of convex $X_j$, i.e., in the column space of $\Phi_j$, $\zeta_j^{k+1} = \tilde{x}_j^{k+1}$.
    • Otherwise $\left(\tilde{x}_j^{k+1} \notin X_j\right)$: Find a feasible projection of $\tilde{x}_j^{k+1}$ onto the feasible region $X_j$, $\tilde{\zeta}_j^{k+1}$ near $\tilde{x}_j^{k+1}$;
    • Case 2: the projection $\tilde{\zeta}_j^{k+1}$ is close enough to $\tilde{x}_j^{k+1}$ (i.e., $\|\tilde{\zeta}_j^{k+1} - \tilde{x}_j^{k+1}\| \leq \varepsilon_j^k$ where $\varepsilon_j^k \triangleq \tau_j^k \|\tilde{x}_j^{k+1} - x_j^k\|$); accept the solution $\zeta_j^{k+1}$, $\zeta_j^{k+1} = \tilde{\zeta}_j^{k+1}$;
    • Case 3: $\tilde{\zeta}_j^{k+1}$ is not sufficiently close; reject the solution $\zeta_j^{k+1}$, $\zeta_j^{k+1} = x_j^k$;
    • update $\hat{x}_j^{k+1}$, $\hat{x}_j^{k+1} = x_j^k + \Delta^k\left(\zeta_j^{k+1} - x_j^k\right)$;
    • set the error bound asymptotically to vanish as iteration proceeds (i.e., $\lim_{k\to\infty} \tau_j^k = 0$).
5. Compute $y^{k+1} = \arg\min_y h\left(\hat{x}_j^{k+1}, y, z_j^k\right)$, and update $z_j^{k+1} = z_j^k + \rho_j\left(\hat{x}_j^{k+1} - \Phi_j^T y^{k+1}\right)$ accordingly; and $\Delta^{k+1} = \Delta^k - a\left(\Delta^k\right)^2$ where $a \in (0.5, 1)$.
6. Update $x_j^{k+1}$ so that all $x$-variables are consistent with global variable $y$ (i.e., $x_j^{k+1} - \Phi_j^T y^{k+1} = 0$).
7. $k \leftarrow k + 1$, and go to 2.
```

Fig 3 illustrates how the x-optimization process determines the $\hat{x}_j^{k+1}$ described in Rule 4 of the algorithm. If the solution $\tilde{x}_j^{k+1}$ is feasible in X_j, $\hat{x}_j^{k+1}$ is the solution (Case 1). If the solution $\tilde{x}_j^{k+1}$ is not in X_j but the point projected onto the convex region X_j is sufficiently close, $\hat{x}_j^{k+1}$ is the projected point (Case 2). For Case 3, where either no solution is identified or the identified solution is not in X_j, and the distance to the projected point is too far away, the solution is rejected, and $\hat{x}_j^{k+1}$ is the point determined in the previous step. Upon the determination of $\hat{x}_j^{k+1}$, the iteration proceeds to a point between the current and the projected point depending on parameter Δ^k.

After the x-optimization is performed, y-optimization follows. The y-optimization at given $\hat{x}_j^{k+1}$ and z^k is a least square problem–convex problem. It should be noted that the solution to the convex problem is always feasible. Rule 6 is a striking feature of the proposed algorithm that is critically different from the ADMM approaches. Instead of the locally independent update of the primal variables in the ADMM approaches, the nodal variables are updated 1) linearly for the computational efficiency, and 2) with respect to the central variable so that the update is reasonably agreeable among nodal variables. The proof of the convergence of the proposed algorithm is presented in this paper's S2 Appendix.

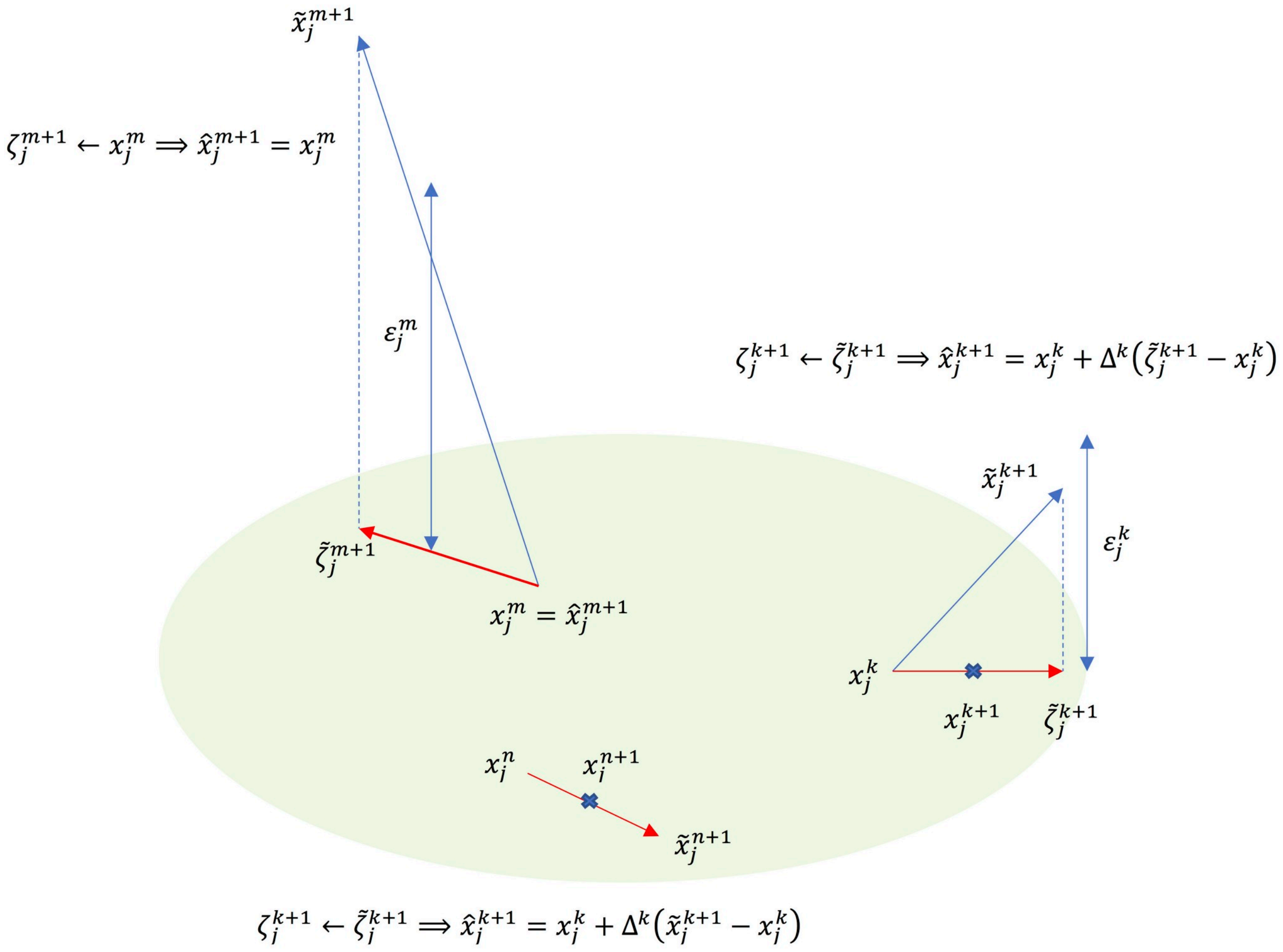
$\tilde{x}_j^{m+1}$
$\zeta_j^{m+1} \leftarrow x_j^m \Longrightarrow \hat{x}_j^{m+1} = x_j^m$
ε_j^m
$\zeta_j^{k+1} \leftarrow \tilde{\zeta}_j^{k+1} \Longrightarrow \hat{x}_j^{k+1} = x_j^k + \Delta^k(\tilde{\zeta}_j^{k+1} - x_j^k)$
$\tilde{x}_j^{k+1}$
ε_j^k
$\tilde{\zeta}_j^{m+1}$
$x_j^m = \hat{x}_j^{m+1}$
x_j^k
x_j^{k+1}
$\tilde{\zeta}_j^{k+1}$
x_j^n
x_j^{n+1}
$\tilde{x}_j^{n+1}$
$\zeta_j^{k+1} \leftarrow \tilde{\zeta}_j^{k+1} \Longrightarrow \hat{x}_j^{k+1} = x_j^k + \Delta^k(\tilde{x}_j^{k+1} - x_j^k)$

Fig 3. ***x*-update from the result of optimization with 1) a feasible solution $\tilde{x}_j^{n+1}$, 2) an infeasible solution, but the distance between the solution and the projection onto the feasible region is close enough $\|\tilde{\zeta}_j^{k+1} - \tilde{x}_j^{k+1}\| \leq \varepsilon_j^k$, and 3) an infeasible solution in that the projection onto the feasible region is too far $\|\tilde{\zeta}_j^{m+1} - \tilde{x}_j^{m+1}\| \leq \varepsilon_j^m$.**

https://doi.org/10.1371/journal.pone.0251948.g003

Implementation details

Implementation of Rule 4

The local SDP relaxation yields a matrix (known as $\tilde{Z}_j^{k+1}$) instead of a vector $\tilde{x}_j^{k+1}$; therefore, a new criterion should be established to compute $\tilde{\zeta}_j^{k+1}$, as follows: Because $\tilde{Z}_j^{k+1}$ is from SDP, all the eigenvalues are non-negative (i.e., $\tilde{Z}_j^{k+1} = \tilde{\zeta}_j^{k+1}\left(\tilde{\zeta}_j^{k+1}\right)^T + \sum_{m=2}^{r} \psi_{j,m}^{k+1}\left(\psi_{j,m}^{k+1}\right)^T$) where $\psi_{j,m}^{k+1}$ is the m^{th} column vector, and r is the rank of $\tilde{Z}_j^{k+1}$:

$$\left\|\tilde{\zeta}_j^{k+1}\left(\tilde{\zeta}_j^{k+1}\right)^T - \tilde{Z}_j^{k+1}\right\| \leq \|\tilde{\zeta}_j^{k+1} + \tilde{x}_j^{k+1}\|\|\tilde{\zeta}_j^{k+1} - \tilde{x}_j^{k+1}\| \tag{5}$$

To construct an equivalent criterion to $\|\tilde{\zeta}_j^{k+1} - \tilde{x}_j^{k+1}\| \leq \varepsilon_j^k$ with $\|\tilde{\zeta}_j^{k+1} + \tilde{x}_j^{k+1}\| \leq 2\|\tilde{\zeta}_j^{k+1}\| + \|\tilde{\zeta}_j^{k+1} - \tilde{x}_j^{k+1}\| \leq 2\|\tilde{\zeta}_j^{k+1}\| + \varepsilon_j^k$, we impose $\|\tilde{\zeta}_j^{k+1}\left(\tilde{\zeta}_j^{k+1}\right)^T - \tilde{Z}_j^k\| \leq \varepsilon_j^k\left(\varepsilon_j^k + 2\|\tilde{\zeta}_j^{k+1}\|\right)$ to determine whether or not the feasible projection is acceptable. The grouping criterion is simplified to $\frac{1}{2\|\tilde{\zeta}_j^{k+1}\|}\left\|\sum_{m=2}^{r}\psi_{j,m}^{k+1}\left(\psi_{j,m}^{k+1}\right)^T\right\| \leq \varepsilon_j^k$ by invoking $\varepsilon_j^k \ll \|\tilde{\zeta}_j^{k+1}\|$. Let $\lambda_{\tilde{Z}_j^k}^m$ be the m^{th} eigenvalue of $\tilde{Z}_j^k$. Then, the criterion becomes $\lambda_{\tilde{Z}_j^k}^2 \leq 2\lambda_{\tilde{Z}_j^k}^1\varepsilon_j^k$.

The definition of $\varepsilon_j^k \triangleq \tau_j^k\|\tilde{x}_j^{k+1} - x_j^k\|$ is also modified as $\varepsilon_j^k \triangleq \tau_j^k\left[\sqrt{\|x_j^k\|^2 + \left\|\tilde{Z}_j^{k+1} - x_j^k\left(x_j^k\right)^T\right\|} - \|x_j^k\|\right]$ using $\|\tilde{Z}_j^{k+1} - x_j^k\left(x_j^k\right)^T\| \leq \left(2\|x_j^k\| + \|\tilde{x}_j^{k+1} - x_j^k\|\right)\|\tilde{x}_j^{k+1} - x_j^k\|$.

Choice of parameters

In contrast to the ADMM approaches, only a few parameters are assigned in the proposed algorithm. Because $\left\{z_j^k\right\} \in Null\left(\Phi_j\right)$, and the specific choice of z_j is irrelevant, all z_j's are random vectors in the null space of known Φ_j. It is important to hold $\sum_{k=1}^{\infty}\|\varepsilon^k\|^2 < \infty$ to guarantee the convergence of the proposed algorithm [24]. Due to the fact that $\sum_{k=1}^{\infty}\|\varepsilon^k\|^2 \leq \|\tilde{x}_j^{k+1} - x_j^k\|_{\max}^2\sum_{k=1}^{\infty}\left(\tau_{\max}^k\right)^2$, the maximum τ over all nodes at the k^{th} iteration is set by $\tau_{\max}^k = \tau_{\max}^0/k$. The parameters used were as follows: $\rho_j = 20$ for v_L and 200 for v_M; $\Delta^0 = 0.3$; and the number of maximum iteration = 100; $\tau_{\max}^0 = 10^{-3}$.

The global variable y is randomly assigned (cold start) to test the algorithm's robustness. For comparison, a flat start and a warm start are also attempted. Here, a flat start refers a point at which real voltages are unity, and imaginary voltages are within [-0.1, 0.1]; and a warm start [20] is a solution to the PF. It is recognized that a faster convergence is obtained when the initial point is close to the feasible region, which is commonly observed in numerical iterative methods.

Termination criterion

The algorithm is terminated when no further progress is made. The progress is measured in terms of the global variable y, and the local variables x and z. After the solution is identified, the objective function W^* is determined according to (3B) where $W(x, y, z) \triangleq \sum_i w_i(x_i, y, z_i)$, i.e., $W^* = W(x^*, y^*, z^*) = \sum_i w_i^*\left(x_i^*, y^*, z_i^*\right)$. The interim value of W at the k^{th} iteration is $W^k = W^k(x^k, y^k, z^k) = \sum_i w_i^k\left(x_i^k, y^k, z_i^k\right)$, and the progress measure η^k at the k^{th} iteration is defined:

$$\text{Progress measure } \eta^k \triangleq \frac{W^k - W^*}{W^*} \tag{6}$$

The criterion used in Rule 2 terminates the algorithm if the progress measure is less than the tolerance (10^{-7}) (i.e., $\eta^k \leq 10^{-7}$).

Results and discussion

Simulation environments

The model systems used in the simulations are available from MATPOWER [25] for 3-, 4-, 9-, 14-, 24-, 30-, 39-, 57-, 85-, 118-, 300-, and 2,000-bus systems. All simulations were performed using a Mac pro with 2 × 2.93 GHz 6-core Intel Xeon processors and 6 GB 1333 MHz DDR3 memory. The local SDP problems were solved using the CVX solver [26]. To compare the results from various systems, the lines in the figures are normalized so that all start at zero. The initial points for all the cases are cold starting points, and all the real and imaginary voltages are set to random numbers (not even a flat start). The qualities of the solutions are all numerically identical to those identified using MATPOWER for the model systems tested (all the solutions v^* are less than 10^{-4} from the MATPOWER solutions ($v_{MATPOWER}$), $\|v^* - v_{MATPOWER}\|/\|v^*\| \leq 10^{-4}$). For comparison, we also attempted a flat start and observed that N_{iter} decreases at least 30%, but the flat start also finds the same solutions.

Subsystems of the test cases

In the ADMM approaches referred to in the literature [14–20], the grid partitioning is performed, but the details of the partitioning are not well listed. In considering the computational coupling, the partitioning is based on spectral clustering [20, 27] while each subsystem contains at least one generator [14, 20]. The inclusion of a generator in a subsystem seems a natural choice to fulfill the power balance equality constraints (1 in (3)) within each substation. However, the inclusion of a generator does not serve the purpose well, because the generator may not be dispatched if its generation costs are excessively high. The spectral clustering is performed using two different algorithms: 1) unnormalized [27] and 2) normalized [28]. The algorithms yield different grid clustering. For example, the unnormalized algorithm results in two clusters for the IEEE 3-bus systems, while the normalized algorithm finds a single cluster for the same system [28]. Fig 4 illustrates the path length, maximum number of nodes in a subsystem, and the number of subsystems.

The lines are best-fit lines, showing the positive correlations of path length (red dotted line path length $\propto Nb^{0.30}$), the maximum number of nodes in a subsystem (green solid line $\propto Nb^{0.43}$), and the number of subsystems (blue broken line $\propto Nb^{0.76}$) based on the unnormalized algorithm (left plot). The results using the algorithm with a different normalization [28] yield path length $\propto Nb^{0.34}$, the maximum number of nodes in a subsystem $\propto Nb^{0.45}$, and the number of subsystems $\propto Nb^{0.66}$ (right plot). Even though the results from the two plots are not exactly the same, the positive correlations with the system size are clear. They affect the computational efficiency of ADMM [14, 20]; path length affects the communication costs and N_{iter} because the decision at each subsystem should be delivered to the rest of the system; the number of subsystems affects the number of computation cores and the communication costs unless PL equals one; and the maximum number of nodes affects the computation costs at each subproblem. Therefore, the decrease in the computation costs at each subsystem is achieved in exchange for the increased costs of communication among subsystems. Because the path length and number of subsystems increase rapidly with system size, the improvement in computational efficiency of the ADMM approach is questionable.

Maximum cardinality of nodal variables in the proposed algorithm

The nodal variables in the x-optimization μ_{j} are $\left[\alpha_i^{4\times1};\ \beta_i^{4\times1};\ \gamma_{i-l,i}^{4nl_i\times1};\ \delta_{i-l,i}^{4nl_i\times1};\ \omega_i^{2\times1};\ \breve{f}_{i-l,i}^{nl_i\times1};\ \bar{f}_{i-l,i}^{nl_i\times1};\ g_i^{2ng_i\times1};\ \mu_{i,end}^{1\times1}\right]$. Because μ_{end} is unity, the cardinality of μ_{j} is $10nl + 2ng + 10$ where nl is the number of lines connected to the j^{th} node, and ng is the number of generators located at the node. In determining the voltages through the power channel, the node with a high value for nl plays a key role. Fig 5 illustrates the maximum cardinality $n_{\mathrm{var}}^{\max}$ of μ_{j} with the systems' sizes. The dashed line indicates $n_{\mathrm{var}}^{\max} \propto Nb^{1/4}$. Even though the positive relationship between $n_{\mathrm{var}}^{\max}$ and Nb is visible among the model systems, the

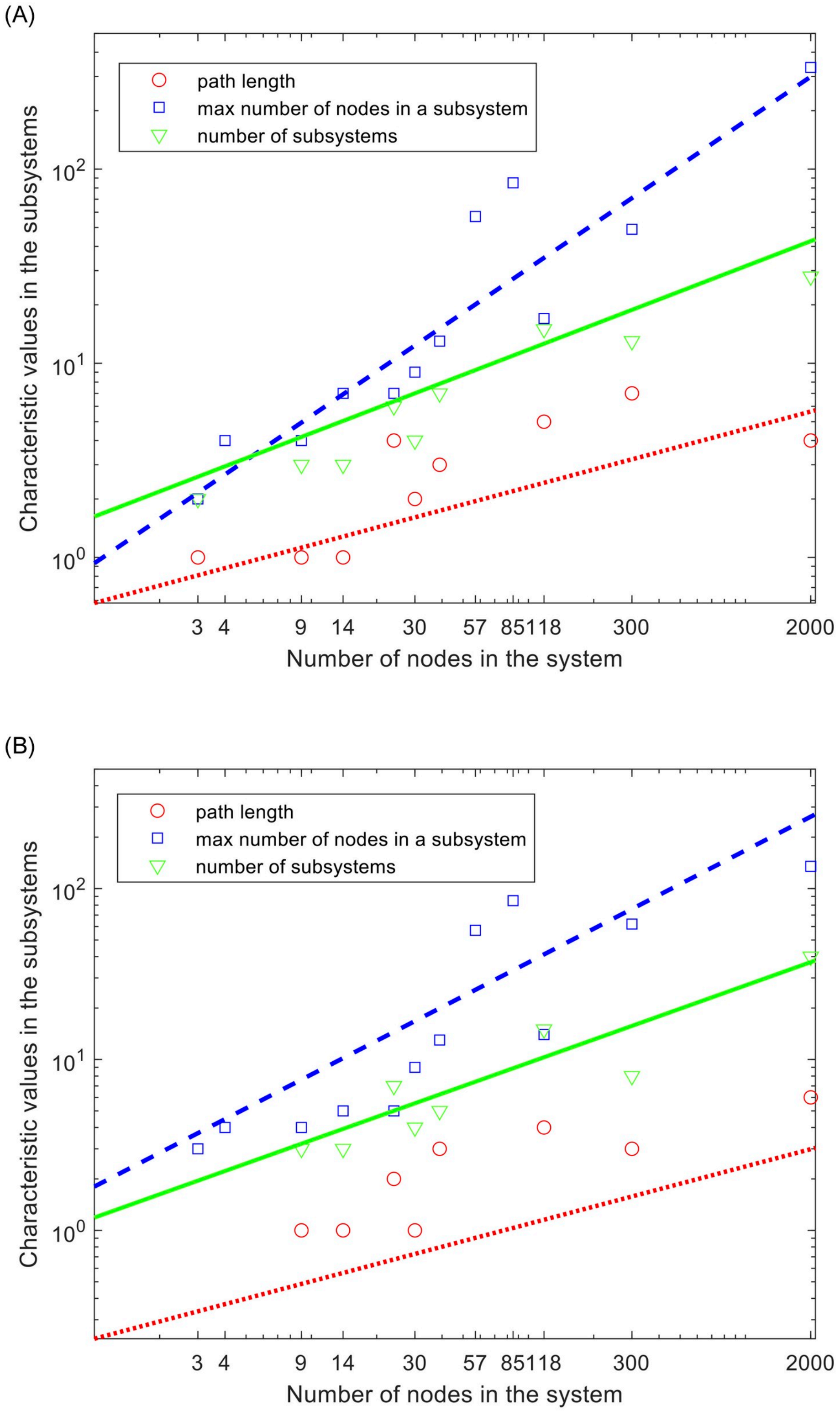

Fig 4. The path length, the maximum number of nodes in a subsystem, and the number of subsystems for the IEEE 3-, 4-, 9-, 14-, 24-, 30-, 39-, 57-, 85-, 118-, 300-, and 2,000-bus systems. The lines indicate the positive correlations on the size of the system. The red-dotted line, the blue broken line, and the green solid line represent the best-fit curves for the path length, the maximum number of nodes in a subsystem, and the number of subsystems, respectively. Two algorithms are applied for clustering nodes: (A) unnormalized algorithm, and (B) normalized based on the algorithm in [28].

https://doi.org/10.1371/journal.pone.0251948.g004

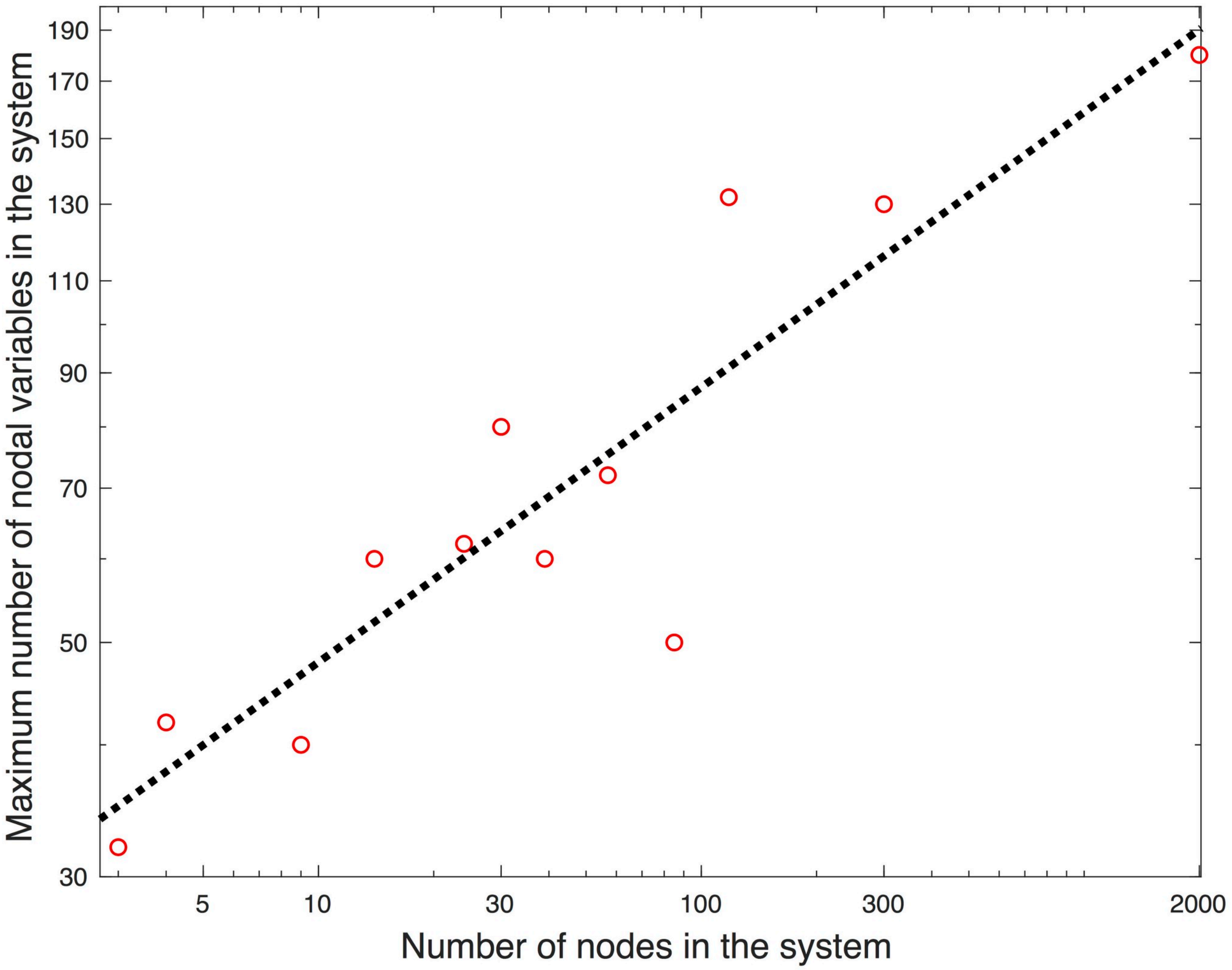

Fig 5. Cardinalities of nodal variables in *x*-optimization in terms of system size.

https://doi.org/10.1371/journal.pone.0251948.g005

relationship is not necessarily positive. The number of variables in the central OPF is typically $2Nb + 2Ng$ where Nb and Ng are the numbers of buses and generators in the system, respectively. For small systems, such as 3-, 4-, 9-, 14-, and 24-bus systems, the maximum cardinalities of the nodal variables are higher than the number of variables in the central OPF; therefore, it would be challenging to keep the computational costs of the distributed OPF for the small systems lower than those of the central OPF. It is noteworthy that the communication costs remain manageable, because each node directly communicates with the central node due to the fact that PL equals unity.

Distributed OPF for 3-, 4-, 24-, 39-, and 85-bus systems

To examine whether the choice of parameters affects the convergences, optimizations were executed with many randomly assigned initial points, and uniform convergences were consistently observed (Fig 6). In general, N_{iter} depends on the initial guess of the voltages. As the distance of the initial point from the solution becomes closer, N_{iter} decreases, which is commonly observed in numerical iterative methods. All the solutions identified are numerically the same as the solutions using MATPOWER.

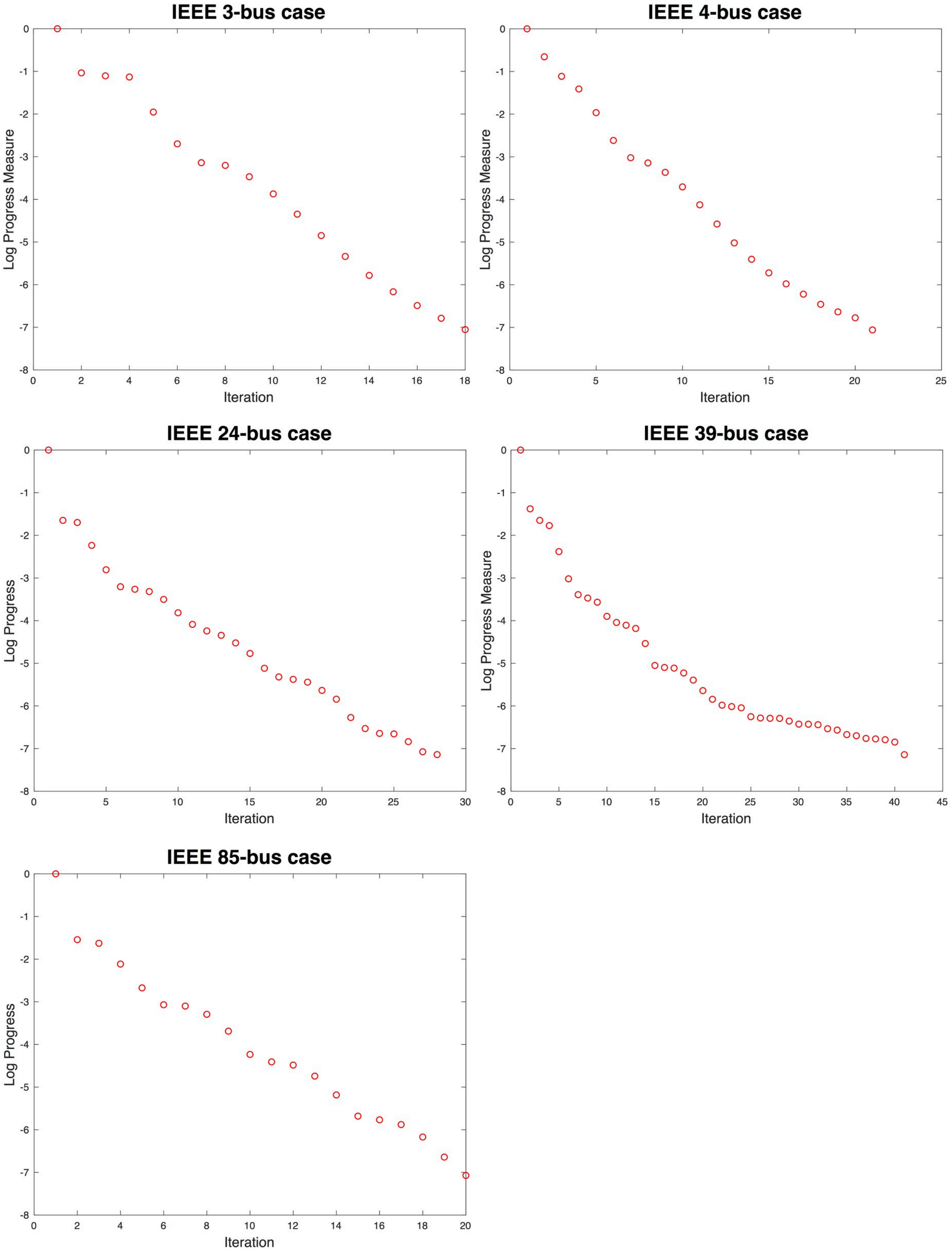

Fig 6. Convergence of proposed algorithm for selected systems (i.e., IEEE 3-, 4-, 9-, 14-, 24-, 39-, and 85-bus systems).

https://doi.org/10.1371/journal.pone.0251948.g006

A worst-case convergence was observed for the IEEE 39-bus system. The nodal OPF for the system involves large negative eigenvalues (i.e., the nodal OPFs are highly nonconvex). According to Rule 4, a relatively large number of solutions are rejected because they are not close to the feasible regions and, therefore, N_{iter} becomes large (approximately 40 iterations). However, the solution identified is a local minimizer, and N_{iter} is still much smaller than the number of iterations for the ADMM approaches. For the visual presentations of the results obtained for various systems, the progress is normalized to the 1st iteration (i.e., the curves begin at 0 for the 1st iteration).

Comparison of convergence to ADMM approaches

The convergence measures were compared to those from the ADMM approaches reported in multiple studies [14, 17–19]. It is worth noting that Erseghe [14] and Zhang et al. [17] do not take the flow limits into consideration; the approach of Engelmann et al. [18] involves a high communication cost; and the approach of Madani, Kalbat, and Lavaei [19] may yield a physically infeasible solution. We attempted the ADMM approaches with the flow constraints and feasibility, but our implemented solver failed to converge with any starting points. Instead, we compared the convergence behaviors of the proposed algorithm to those reported in the previous [14, 17–19]. For the visual presentation, the convergence behaviors are normalized so that all the curves begin at the same point (see Fig 7 for the comparison of the convergence).

In the previous research [14, 17–19], the size of the subsystems increased with that of the system. Therefore, the tradeoff between communication costs amongst the subsystems and the computation costs for each subsystem make it difficult to develop a scalable algorithm. As the system size increases, N_{iter} increases significantly for the results in two of the previous studies [14, 19]. The performance measure in another study [17] fluctuates consistently with various systems, indicating that the convergence may not be guaranteed. Because the final study [18] requires the information exchange regarding sensitivities as well as the primal variables, the computation and communication costs per iteration increase much more rapidly than do the computation costs of a central OPF solver.

Different from the clustering containing multiple nodes used in the ADMM approach, the proposed algorithm for each subsystem contains only 1 node regardless of the system size; each subsystem directly communicates with the rest of the entire system; and the communication involves only the primal variables. With this difference in mind, the fast convergences observed in the proposed algorithm are similar to those in two of the previous studies [17, 18], which are consistently much faster than those in the other two previous studies [14, 19]. Whereas N_{iter} are like those reported by Engelmann et al. [18], the convergences of the proposed algorithm occur more uniformly and consistently.

Many approaches using ADMM do not consider the flow limits [14, 17], and/or feasibility [19] and, therefore, it is not appropriate to compare the quality of the solutions. Instead, we compared our solutions with those using a central OPF solver, MATPOWER. For all the cases described above, they yield numerically identical solutions. It is not clear why the proposed algorithm finds the same solution as the central OPF solver. We tested with the central SDP-relaxation for a small-scale system to estimate the "global" solution. Due to the memory issue, our SDP tests are limited up to 118-bus systems. When the relaxation returns the rank-1 solution, the solution is global and physically feasible. For several cases, the test cases are of the global solutions that are also identified by MATPOWER and by the proposed algorithm. However, there are cases where the SDP relaxation returns a physically infeasible solution. For these cases, both MATPOWER and the proposed algorithm identify numerically the same local minimizers that may not be global solutions. MATPOWER finds a point that meets the

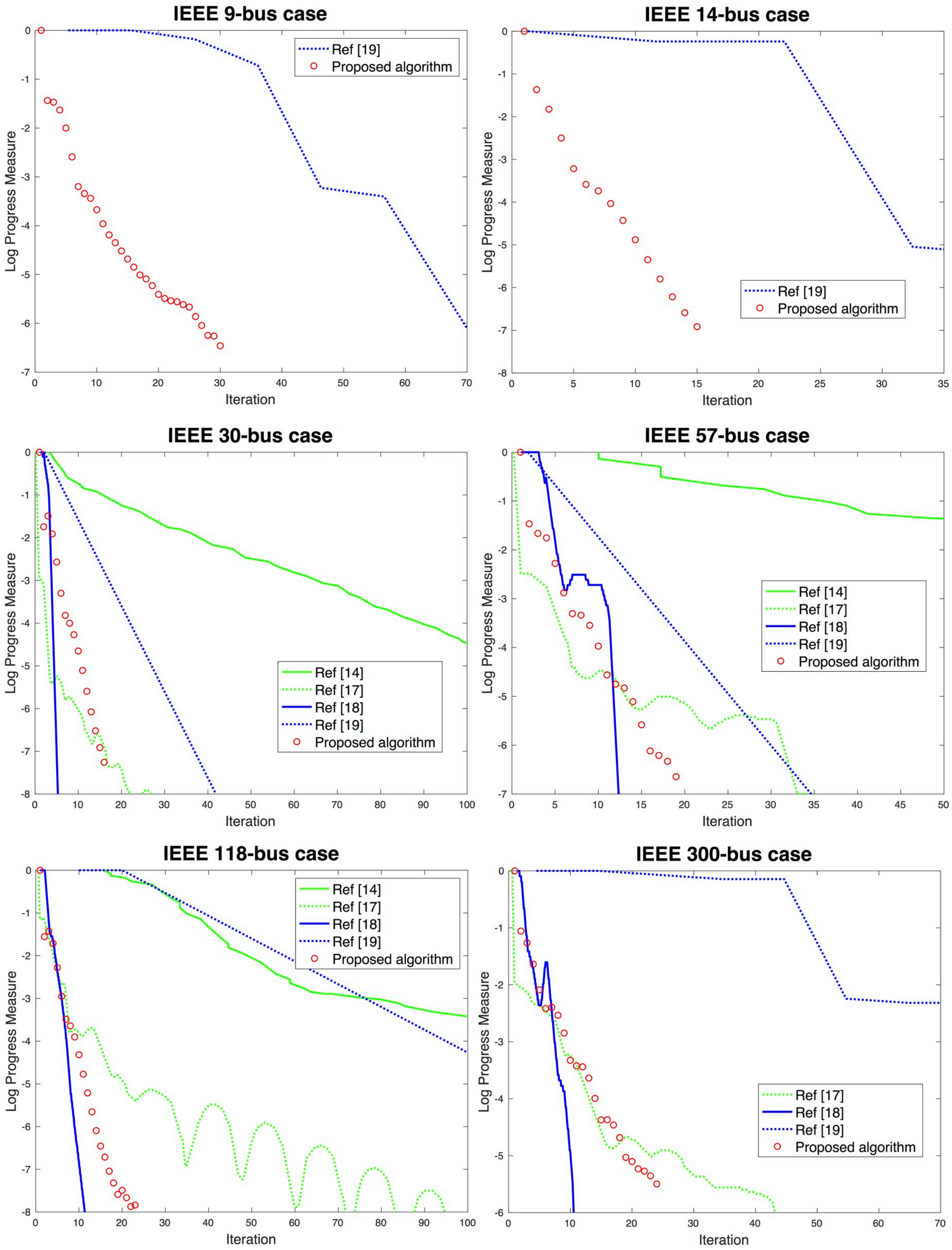

Fig 7. Convergence of proposed algorithm for selected small systems (i.e., IEEE 9-, 14-, 30-, 57-, 118-, and 300-bus systems). The convergences of the ADMM approaches [14, 17–19], are compared.

https://doi.org/10.1371/journal.pone.0251948.g007

first-order necessary conditions for optimality [25]. Although it finds a minimizer in most cases, there is no guarantee that the identified solution is a minimizer—it can be either a maximizer or a saddle point. On the other hand, the proposed algorithm identifies a solution that meets the second-order optimality conditions, which guarantees that the solution is a minimizer [29]. From a practical point of view, there are two advantages of the proposed algorithms over MATPOWER: 1) they are numerically stable and, therefore, robust because all the subproblems are convex—no issues regarding the rank-deficient Hessian matrix, and 2) they use distributed computation and, therefore, manageable computation in each subproblem. However, there are disadvantages of the proposed algorithm: 1) the requirement of multiple cores to perform the distributed optimization, and 2) the number of nodal variables can be larger than that of the central OPF; for example, 3-, 4-, 9-, 14-, and 24-bus (See Fig 5).

Large-scale OPF: 2,000-bus system

We tested the algorithm for a large-scale system, the 2,000-bus system that is a synthetic grid on a footprint of Texas. Fig 8 presents a convergence pattern. The solution identified is numerically identical to the one found using MATPOWER. N_{iter} remains small, as is the case for the small systems (See Figs 6 and 7). Note that the nodal OPF includes only 1 bus, and the communication costs remain small because PL equals unity. If a sufficient number of computation cores are provided, the proposed algorithm is scalable if the computation cost for solving a subproblem does not increase significantly as the system size increases.

Different from other algorithms reported in the literature (i.e., [14–18]), the proposed algorithm converges regardless of the initial points, but the number of iterations decreases by at least a factor of 2 if it uses a flat start. Another shortcoming of the algorithms reported in previous research [14–18], is the quality of their solutions. Whereas the existing OPFs find low-voltage [17] or suboptimal [16] solutions, the proposed algorithm finds the same solution as a central OPF solver. One previous study [16] claimed that there might be an optimal number of partitioned subsystems due to the tradeoff between communications and computational costs used to obtain a local solution. The proposed method keeps the PL = 1, and the central computation is a simple addition of x_j through both the power and voltage channels. From the simulations with various parameter values such as ρ_j, Δ^k, τ_j^1, and a, we obtained the same solutions, indicating that the proposed algorithm is robust as well as efficient. In addition, the proposed model does not require any partitioning.

Comparison to a heuristic central OPF solver, MATPOWER

In comparing the convergence between the proposed algorithm and the central OPF, the number of variables and N_{iter} were examined. The maximum cardinality $n_{\text{var}}^{\max}$ of μ_j increases with the system size in $n_{\text{var}}^{\max} \propto Nb^{1/4}$ (See Fig 4). Fig 9 illustrates the number of variables in the central OPF as well as the maximum cardinality of the nodal variables. The blue line is the best-fit line for the central OPF ($n_{\text{var}}^{OPF} = 2Nb + 2Ng \propto Nb^{0.96}$), whereas the red line is the best-fit line for the distributed algorithm. It is clear that n_{var}^{OPF} increases with Nb in a much faster way than $n_{\text{var}}^{\max}$ does. The black line is the boundary at which n_{var}^{OPF} equals $n_{\text{var}}^{\max}$. If enough computation cores are available, the proposed algorithm involves reduced computation costs per iteration for systems larger than or equal to IEEE 24-bus. Whereas n_{var}^{OPF} almost linearly increases with Nb because $Ng \ll Nb$ in most systems, $n_{\text{var}}^{\max}$ does not necessarily increase theoretically. A key observation is that the proposed approach can be a scalable algorithm if N_{iter} does not rapidly increase with the system size.

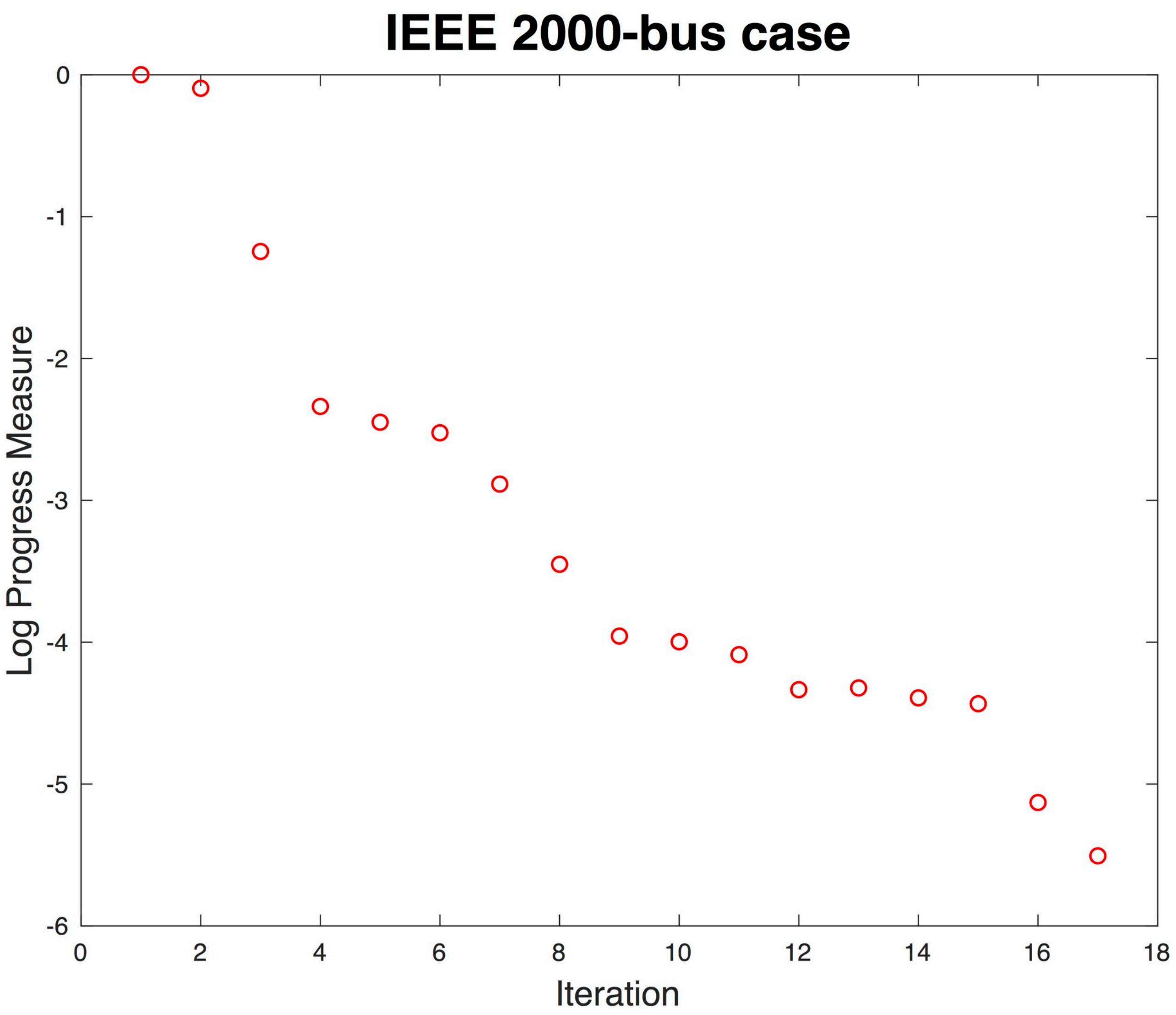

Fig 8. Convergence of proposed algorithm for synthetic 2,000-bus system on a footprint of Texas.

https://doi.org/10.1371/journal.pone.0251948.g008

Fig 10 presents N_{iter} for the central OPF (Nit_{cent}) and of the proposed algorithm (Nit_{dist}) on the tested systems. A visible increase in Nit_{cent} is observed with Nb, but the dependence of Nit_{dist} on Nb is not evident—the solid lines are best-fit curves (log-log plot) that indicate $Nit_{cent} = Nb^{0.16}$ and $Nit_{dist} = Nb^{-0.007}$, respectively. The dotted lines are average N_{iter}, and the values are 15 and 23, respectively.

From the comparisons (the number of variables and N_{iter}), we conclude that the computation cost of the distributed algorithm increases with Nb at a much slower rate than that of the central OPF if sufficient computational resources are available.

Scalability of the algorithm

The computation efficiency of a distributed computation depends on the number of iterations and the computational cost per iteration. The cost is determined by the local computation

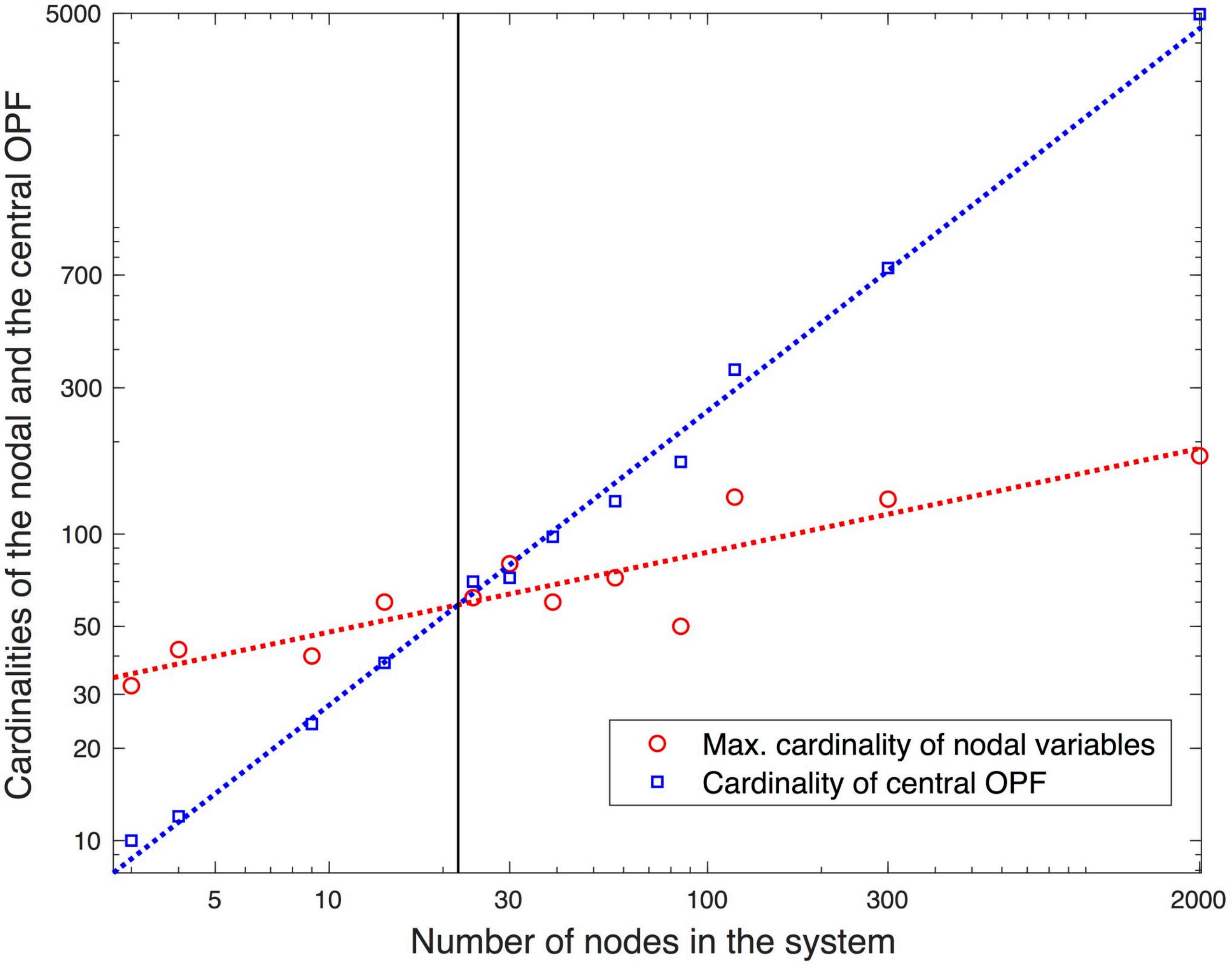

Fig 9. Cardinalities of central OPF and proposed algorithm for the test systems.

https://doi.org/10.1371/journal.pone.0251948.g009

with the largest number of variables. Fig 10 shows that the number of iterations does not increase with system size.

Fig 11 illustrates the maximum nodal computation time depending on the maximum cardinality of the nodal variables. This performance dependence on the number of nodal variables may shed light on the claim by Loukarakis, Bialek, and Dent [30] that a larger system does not necessarily imply an inferior convergence performance. The dashed line indicates that the maximum nodal computation time is proportional to the maximum cardinality of the nodal variables with the power of 2.64. The cardinality to the power of 2.64 observed in this study is close to the theoretically estimated 3 for the SDP solver. Note that it is not necessary for the maximum cardinality of the nodal variables to be positively correlated with the system size. Rather, the maximum cardinality depends on the local grid topology of the system. For the tested systems, $n_{\text{var}}^{\max} \propto Nb^{1/4}$ and $CT \propto \left(n_{\text{var}}^{\max}\right)^{2.64}$, which results in $CT \propto Nb^{2.64/4.} = Nb^{0.66}$ where CT represents the maximum nodal computation time. The total computation cost is bounded by the product between N_{iter} and CT. The total computation cost is bounded by $CT \propto \vartheta(\lceil Nb/N_{core} \rceil Nb^{0.66})$ where N_{core} is the number of cores. If the computational resource is

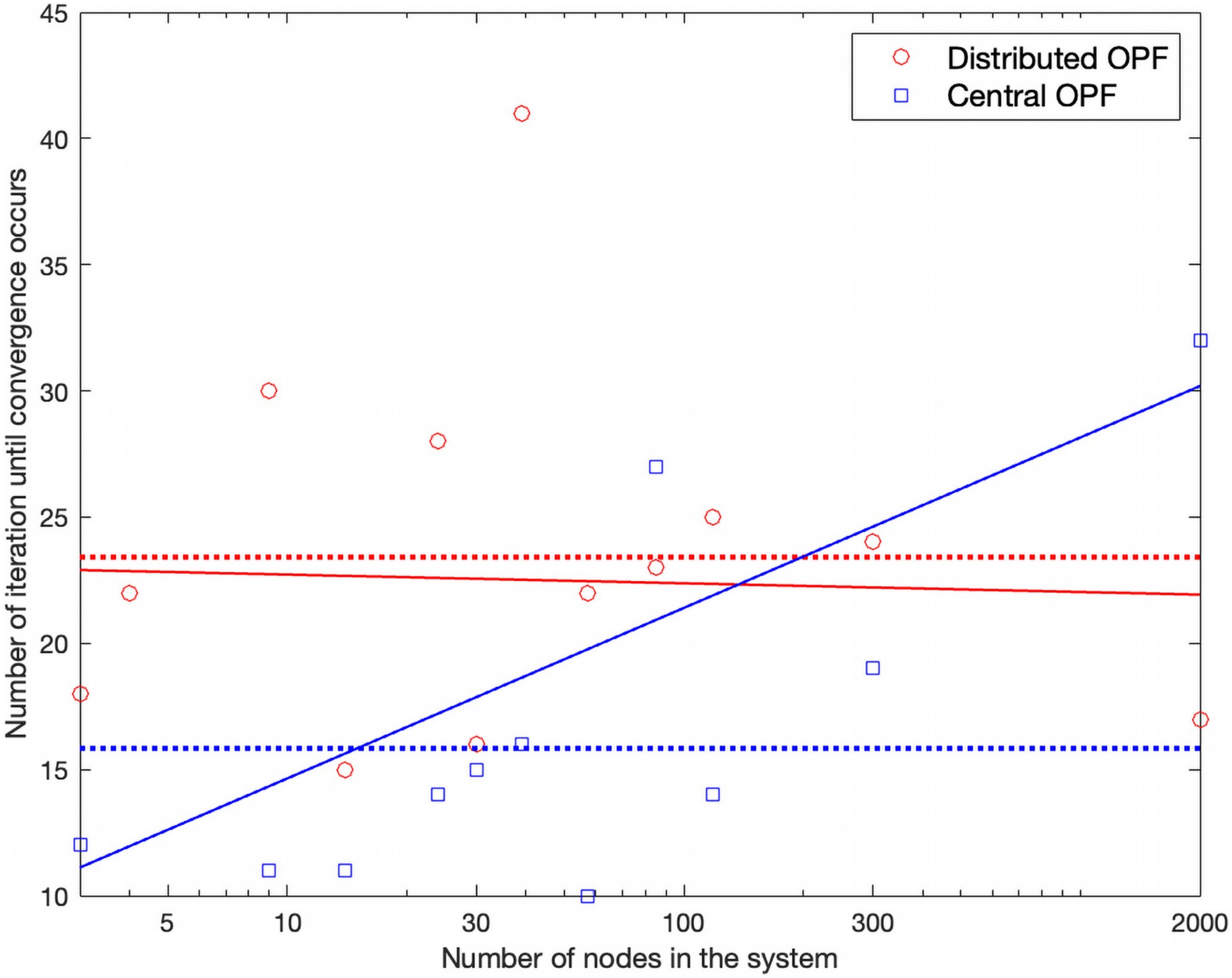

Fig 10. N_{iter} for central OPF and of the proposed algorithm on the tested systems. Dotted lines are averages, and solid lines are best-fit curves.

https://doi.org/10.1371/journal.pone.0251948.g010

sufficient, $CT \propto \vartheta(Nb^{0.66})$. From this observation, the computation cost increases sub-linearly for a large-scale network.

If a single core is available for computation, the computation cost is in $\vartheta(Nb^{1+p_{SDP}/4})$ where p_{SDP} is the computational complexity for a nodal SDP. For the state-of-art central heuristic OPF solvers, the computation cost is in $\vartheta(Nb^{1.5})$. Therefore, when the computation resources are highly limited, the proposed algorithm would still be efficient with a convex problem solver that yields $p_{SDP} \leq 2$. A potential improvement in p_{SDP} of the SDP solver is to explore the sparse structure of the matrices [31] or to utilize a commercial solver such as MOSEK.

Nomenclature

$\Phi_j^{M_{eq}}, \Phi_j^{M_{in}}$ eigenvectors of M_{eq}, M_{in} scaled by $diag\left(\sqrt{\lambda_{eq}}\right)$ and $diag\left(\sqrt{\lambda_{in}}\right)$

Ω_m, Ω_m^c Set and the complement of the set of Case m

α_j, β_j nodal variable associated with real and with reactive power injection, $\alpha_j^{4\times 1} = \Phi_{\breve{S}_j}^T v$ and $\beta_j^{4\times 1} = \Phi_{\bar{S}_j}^T v$

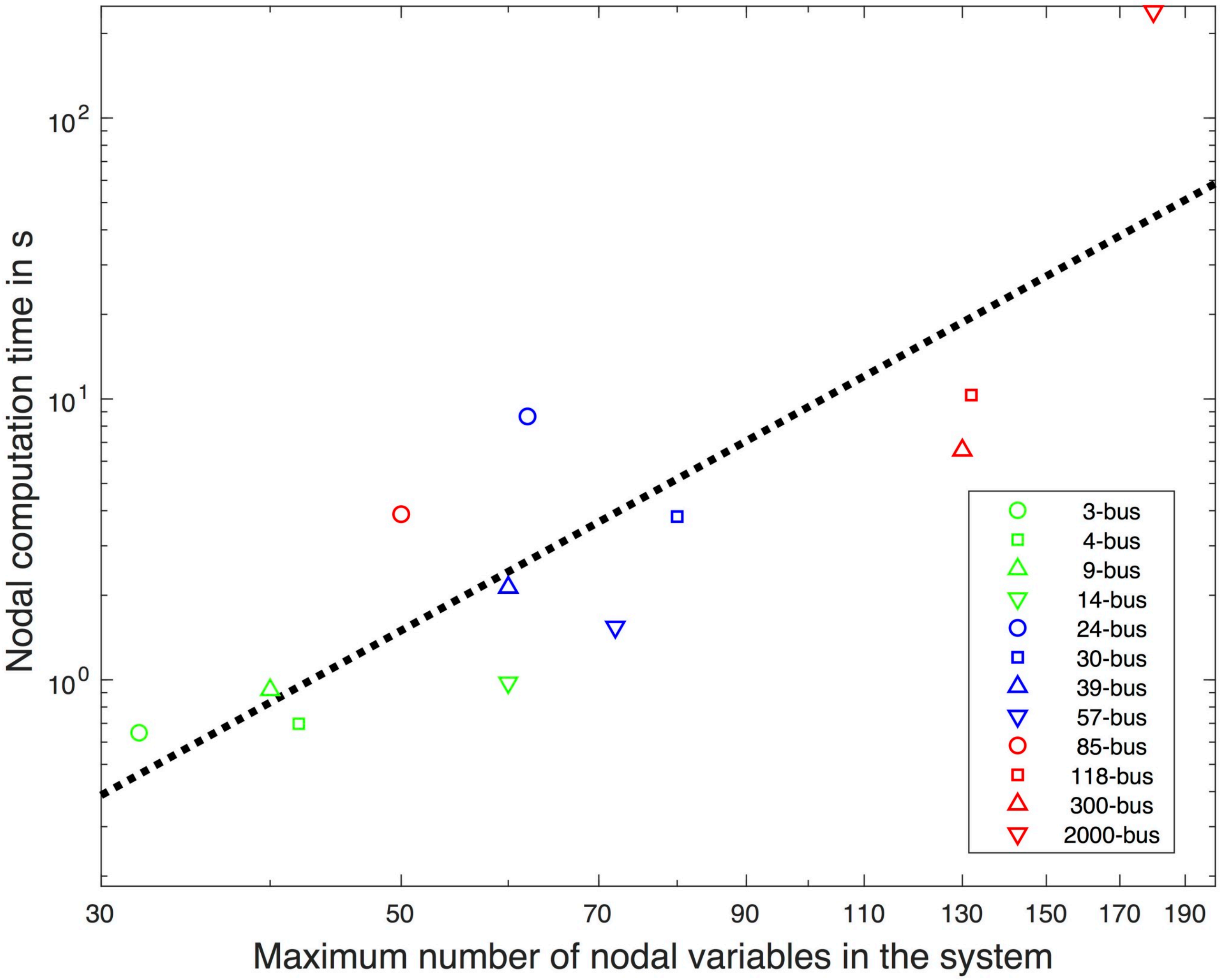

Fig 11. Computation times in seconds for nodal OPF of the maximum cardinality of the nodal variables. Dotted line is the best-fit line with the slope of 2.64.

https://doi.org/10.1371/journal.pone.0251948.g011

$\gamma_{j\text{-}l,j}$, $\delta_{j\text{-}l,j}$ nodal variable with real and with reactive power flow over the line *j-l* at the side of the j^{th} node, $\gamma_{j-l,j}^{4\times1} = \Phi_{\breve{F}^{j}_{j-l}}^{T} v$ and $\delta_{j-l,j}^{4\times1} = \Phi_{\bar{F}^{j}_{j-l}}^{T} v$

ω_j nodal variable with E_j, $\omega_j^{2\times1} = \left(e_j^T + e_{j+Nb}^T\right)^T v$

B, *Br*, *G* sets of nodes, branches, and generators

Br{j}, *G{j}* sets of branches and of generators at node *j*

E_j voltage magnitude square at Node *j*, $\boldsymbol{v}_j^2$

II_j^{eq}, II_j^{in} diagonal matrix of ±1 associated with equality and inequality constraints

I identity matrix

J matrix $[I^T\, \mathbf{j}\, I^T]^T$

M_{eq}, M_{in} symmetric matrices with equality and inequality constraints

$M_j^{eq} = \Phi_j^{M_{eq}} II_j^{eq} \Phi_j^{M_{eq}T}$, $M_j^{iin} = \Phi_j^{M_{in}} II_j^{in} \Phi_j^{M_{in}T}$

$\boldsymbol{Y_{bus}}$, $\boldsymbol{Y_{br}}$ nodal and branch admittance matrices

Nb, *Nl*, *Ng* number of nodes, lines, and generators at the system of interest

N_{iter} number of iterations until convergence occurs

cap_l thermal limit of flow over l

$\boldsymbol{d}_j$ real and reactive power demand at j,$\boldsymbol{d}_j = \breve{d}_j + \mathbf{j}\bar{d}_j$

e_j j^{th} column vector in the identity matrix

$\boldsymbol{f}^j_{j-l}$ power flow over j-l at the j^{th} node,$\boldsymbol{f}^j_{j-l} = \breve{f}^j_{j-l} + \mathbf{j}\bar{f}^j_{j-l}$

$\boldsymbol{g}_j$ generation at the j^{th} node, $\boldsymbol{g}_j = \breve{g}_j + \mathbf{j}\bar{g}_j$

$\mathbf{j}$ $\sqrt{-1}$

nl_j, ng_j number of lines and generators at Node j

$nl_{\max}$ maximum number among nl_j in the system

v_L, v_M voltages at the power and voltage channels

$\boldsymbol{v}$, v, v_x, v_y complex voltage vector, voltage $v = \begin{bmatrix} v_x^T & v_y^T \end{bmatrix}^T$, real and imaginary part of voltage, $\boldsymbol{v} = v_x + \mathbf{j}v_y = \mathbf{j}^T v$

$\boldsymbol{x}$, x, $\breve{x}$, $\bar{x}$ complex variable, real, and imaginary parts of $\boldsymbol{x}$, $\boldsymbol{x} = \breve{x} + \mathbf{j}\bar{x}$ and $x = [\breve{x}^T \quad \bar{x}^T]^T$

Nodal variables and their cardinality

$\alpha_j^{4\times1}$ nodal variable associated with real power injection at Node j

$\beta_j^{4\times1}$ nodal variable associated with reactive power injection at Node j

$\gamma_{jk}^{4\times1}$ nodal variable associated with real power flow $\breve{f}^j_{j-k}$ over a line j-k at the side of Node j

$\gamma_j^{4nl_j\times1}$ nodal variable associated with real power flow over the lines connected to Node j

$\delta_{jk}^{4\times1}$ nodal variable associated with reactive power flow $\bar{f}^j_{j-k}$ over a line j-k at the side of Node j

$\delta_j^{4nl_j\times1}$ nodal variable associated with reactive power flow over the lines connected to Node j

$\omega_j^{2\times1}$ nodal variable associated with the voltage magnitude at Node j

$\breve{f}^j_{j-k}$ scalar representing real power flow over a line j-k at the side of Node j

$-f^j_{j-k}$ scalar representing reactive power over a line j-k at the side of Node j

$p_j^{ng_j\times1}$ nodal real power generation vector at Node j

$q_j^{ng_j\times1}$ nodal reactive power generation vector at Node j

Conclusions and future research directions

From the tensor analysis of the power flow, we developed a star and linear model to achieve a scalable distributed computation. The new network model allows the direct communication between the nodal variables and the central voltages. In the model, the PL remains at unity regardless of the system size. On the other hand, the Kirchhoff's laws and voltage magnitudes are expressed in terms of nodal variables that are linear in the voltages. Therefore, the model makes it possible to keep the size of a nodal OPF small regardless of the size of the system, while the communication costs remain manageable. This new aspect of the model allows us to construct a scalable algorithm that converges to the same solution as the nonconvex OPF. We proposed the DROHS algorithm to find a local minimum using a convex surrogate function. Among the nodal OPF solutions, only near-feasible solutions (Rule 4) are selected for updating. In addition to the high quality of the solution, it also achieves computational efficiency and robustness. We tested the DROHS algorithm for the 3-, 4-, 9-, 14-, 24-, 30-, 39-, 57-, 85-, 118-, 300-, and 2,000-bus systems. The proposed algorithm achieves 1) fast and uniform convergence, 2) provable convergence, 3) the same problem formulation as the central OPF problem without ignoring any constraints, 4) guaranteed convergence to a local minimum, rather than a maximum or saddle point, that meets the first-order necessary conditions for optimality, and 5) a completely distributed algorithm (i.e., a scalable algorithm, which has never been achieved before in the literature). The challenges that the proposed algorithm faces are 1) an increased number of nodal variables that may be higher than that of the variables in the central OPF for a small system, 2) an increased number of iterations when highly connected nodes involve solutions far from the feasible regions, and 3) a prolonged wait time for nodes with low cardinalities. Therefore, the proposed algorithm is an efficient alternative to the central OPF for a large-scale network. Future research directions include the development of 1) a way to accommodate the impact of the rejected solutions in updating the x-variables if the corresponding nodes are highly connected, 2) an efficient computation to solve the nodal SDP, particularly a way to explore the sparse structure of the nodal OPF, and 3) asynchronous distributed optimization for improving the computational efficiency where the scheduling of the distributed computation is identified in terms of a knapsack problem. We also present the proof showing that the surrogate function improves at every iteration and that the iteration converges to a fixed point of the nonconvex OPF. The numerical results exhibit rapid convergence, and the convergence behavior is discussed.

Supporting information

S1 Appendix. Proof of claims.
(DOCX)

S2 Appendix. Proof of convergence.
(DOCX)

Acknowledgments

We thank Dr. Charles Van Loan who provided insight on tensor computation. We would like to express our gratitude to Dr. Robert J. Thomas and Mr. Gilbert Bindewald for their expertise on the computation for the power system analysis during this research, and we also thank 5 anonymous reviewers whose comments and suggestions helped improve and clarify this manuscript.

Author Contributions

Conceptualization: HyungSeon Oh.

Data curation: HyungSeon Oh.

Formal analysis: HyungSeon Oh.

Investigation: HyungSeon Oh.

Methodology: HyungSeon Oh.

Software: HyungSeon Oh.

Validation: HyungSeon Oh.

Visualization: HyungSeon Oh.

Writing – original draft: HyungSeon Oh.

Writing – review & editing: HyungSeon Oh.

References

1. Verma DB and A. Strong NP-hardness of AC power flows feasibility. arXiv:151207315.
2. Molzahn DK, Dörfler F, Sandberg H, Low SH, Chakrabarti S, Baldick R, et al. A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems. IEEE Transactions on Smart Grid. 2017. https://doi.org/10.1109/TSG.2017.2720471
3. Wang Y, Wang S, Wu L. Distributed optimization approaches for emerging power systems operation: A review. Electric Power Systems Research. 2017. https://doi.org/10.1016/j.epsr.2016.11.025
4. Abboud A, Couillet R, Debbah M, Siguerdidjane H. Asynchronous alternating direction method of multipliers applied to the direct-current optimal power flow problem. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing—Proceedings. 2014.
5. Kraning M. Dynamic Network Energy Management via Proximal Message Passing. Found Trends® Optim. 2014. https://doi.org/10.1561/2400000002
6. Chakrabarti S, Kraning M, Chu E, Baldick R, Boyd S. Security Constrained Optimal Power Flow via proximal message passing. 2014 Clemson University Power Systems Conference, PSC 2014. 2014.
7. Gan L, Li N, Topcu U, Low SH. Exact Convex Relaxation of Optimal Power Flow in Radial Networks. IEEE Trans Automat Contr. 2015. https://doi.org/10.1109/TAC.2014.2332712
8. Peng Q, Low SH. Distributed optimal power flow algorithm for radial networks, I: Balanced single phase case. IEEE Trans Smart Grid. 2018. https://doi.org/10.1109/TSG.2016.2546305
9. Peng Q, Low SH. Distributed algorithm for optimal power flow on a radial network. Proceedings of the IEEE Conference on Decision and Control. 2014.
10. Zheng W, Wu W, Zhang B, Sun H, Liu Y. A Fully Distributed Reactive Power Optimization and Control Method for Active Distribution Networks. IEEE Trans Smart Grid. 2016. https://doi.org/10.1109/TSG.2015.2396493
11. Dall'Anese E, Zhu H, Giannakis GB. Distributed optimal power flow for smart microgrids. IEEE Trans Smart Grid. 2013. https://doi.org/10.1109/TSG.2013.2248175
12. Sun AX, Phan DT, Ghosh S. Fully decentralized AC optimal power flow algorithms. IEEE Power and Energy Society General Meeting. 2013. https://doi.org/10.1109/PESMG.2013.6672864
13. Watts DJ, Strogatz SH. Collective dynamics of 'small-world9 networks. Nature. 1998. https://doi.org/10.1038/30918 PMID: 9623998
14. Erseghe T. Distributed optimal power flow using ADMM. IEEE Trans Power Syst. 2014. https://doi.org/10.1109/TPWRS.2014.2306495
15. T. Erseghe. Distributed processing. http://dgt.dei.unipd.it/pages/read/93/
16. Guo J, Hug G, Tonguz OK. A Case for Nonconvex Distributed Optimization in Large-Scale Power Systems. IEEE Trans Power Syst. 2017. https://doi.org/10.1109/TPWRS.2016.2613479 PMID: 28824226
17. Zhang M, Kar RS, Miao Z, Fan L. New auxiliary variable-based ADMM for nonconvex AC OPF. Electr Power Syst Res. 2019. https://doi.org/10.1016/j.epsr.2019.105867
18. Engelmann A, Jiang Y, Muhlpfordt T, Houska B, Faulwasser T. Toward distributed OPF using ALADIN. IEEE Trans Power Syst. 2019. https://doi.org/10.1109/TPWRS.2018.2867682
19. Madani R, Kalbat A, Lavaei J. ADMM for sparse semidefinite programming with applications to optimal power flow problem. Proceedings of the IEEE Conference on Decision and Control. 2015.
20. Guo J, Hug G, Tonguz OK. Intelligent Partitioning in Distributed Optimization of Electric Power Systems. IEEE Trans Smart Grid. 2016. https://doi.org/10.1109/TSG.2015.2490553
21. Vandenberghe L, Balakrishnan VR, Wallin R, Hansson A, Roh T. Interior-point algorithms for semidefinite programming problems derived from the KYP lemma. Lect Notes Control Inf Sci. 2005. https://doi.org/10.1007/10997703_12
22. Oh H. A Unified and Efficient Approach to Power Flow Analysis. Energies. 2019; 12: 2425. https://doi.org/10.3390/en12122425
23. Bauschke HH, Bolte J, Teboulle M. A descent lemma beyond Lipschitz gradient continuity: First-order methods revisited and applications. Math Oper Res. 2017. https://doi.org/10.1287/moor.2016.0817
24. Scutari G, Sun Y. Parallel and distributed successive convex approximation methods for big-data optimization. Lecture Notes in Mathematics. 2018. https://doi.org/10.1007/978-3-319-97142-1_3
25. Zimmerman RD, Murillo-Sánchez CE, Thomas RJ. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans Power Syst. 2011. https://doi.org/10.1109/TPWRS.2010.2051168
26. M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming. http://cvxr.com/cvx

27. Von Luxburg U. A tutorial on spectral clustering. Stat Comput. 2007. https://doi.org/10.1007/s11222-006-9015-6 PMID: 18836571

28. Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems. 2002.

29. Nocedal J, Wright S. Numerical optimization, series in operations research and financial engineering. Springer. 2006.

30. Loukarakis E, Bialek JW, Dent CJ. Investigation of Maximum Possible OPF Problem Decomposition Degree for Decentralized Energy Markets. IEEE Trans Power Syst. 2015. https://doi.org/10.1109/TPWRS.2014.2365959

31. Molzahn DK, Holzer JT, Lesieutre BC, DeMarco CL. Implementation of a large-scale optimal power flow solver based on semidefinite programming. IEEE Trans Power Syst. 2013. https://doi.org/10.1109/TPWRS.2013.2258044