

Globally solving a class of optimal power flow problems in radial networks by tree reduction

Amir Beck¹ · Yuval Beck² · Yoash Levron³ ·
Alex Shtof⁴ · Luba Tetrushvili⁴

Received: 4 August 2017 / Accepted: 10 April 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract We devise an algorithm for finding the global optimal solution of the so-called optimal power flow problem for a class of power networks with a tree topology, also called radial networks, for which an efficient and reliable algorithm was not previously known. The algorithm we present is called the tree reduction/expansion method, and is based on an equivalence between the input network and a single-node network. Finally, our numerical experiments demonstrate the reliability and robustness of our algorithm.

1 Introduction

Power systems are large networks which transport electrical power from generators to consumers. The optimal power flow problem is a mathematical optimization problem concerned with finding the state of the system that minimizes a certain objective, which might involve

The research of Amir Beck was partially supported by the Israel Science Foundation grant 1821/16.

✉ Amir Beck
becka@tauex.tau.ac.il

Yuval Beck
beck@post.tau.ac.il

Yoash Levron
yoashl@ee.technion.ac.il

Alex Shtof
alexshftf@campus.technion.ac.il

Luba Tetrushvili
lubate@tx.technion.ac.il

¹ School of Mathematical Sciences, Tel Aviv University, 6997801 Tel Aviv, Israel

² Physical Electronics Department, Faculty of Engineering, Tel-Aviv University, 6997801 Tel Aviv, Israel

³ Faculty of Electrical Engineering, Technion - Israel Institute of Technology, Haifa 3200003, Israel

⁴ Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, 3200003 Haifa, Israel

the total cost of generation, stability and reliability of the system, subject to the (nonconvex) constraints imposed by the laws of physics, and additional operational constraints. The decision variables, constraints and objective involve physical quantities which arise in the analysis of electrical circuits, such as voltage, current and power. Readers are referred to [11] for a survey on formulations and properties of OPF problems.

The nonconvexity of the feasible set makes the problem computationally intractable in general. In spite of this, the substantial economic and engineering importance of this problem led to the development of various methods for globally solving special instances of the problem, or even more common, to the derivation of approximation and heuristic techniques. For a comprehensive review of these methods, the reader is referred to the extensive review [1] as well as references therein.

In this paper, we concentrate on *radial networks*, which are networks with a tree topology. Due to the nonconvexity of the constraints, even the problem of finding a feasible solution is in general difficult task. However, specific families of radial networks enjoyed some success in finding the global optimum in a computationally efficient manner, mostly via convex relaxations.

The methods proposed in [10, 12] utilize second order cone programming (SOCP) relaxations which yield a global optimum under the assumption that voltage magnitudes are unbounded above or below, and the objective function is a convex quadratic function of a specific form. The methods proposed in [2, 15] utilize semidefinite programming (SDP) relaxations, which yield a global optimum under the assumption that the power produced by the generators is unbounded below, and the power delivered to consumers is unbounded above, and assumptions on the objective function which are similar in nature to those made for the SOCP-based methods. See [16, 17] for a comprehensive tutorial on various relaxations and the conditions that ensure global optimality.

Despite the success of convex relaxations, there are many OPF instances on radial networks for which convex relaxations are incapable of producing the global optimum. For example, in contrast to popular works on convex relaxations, the authors of [14] study the conditions which ensure that SDP relaxations will *not* produce the global optimum, and create a library of such problem instances. This prompts the need for different approaches, which compute global optima for networks under assumptions which are not covered by convex relaxations.

Motivated by the above, in this paper we propose an algorithm which computes a global optimum in a reliable, and computationally efficient manner for a family of radial networks that radically differs from the families solved by convex relaxations. We assume the classical partition of the nodes to PQ and PV nodes with an additional reference node. The objective function is not assumed to have any specific structure.

The solution follows from an equivalence theorem between a radial network of a certain family, and a smaller radial network. The OPF problem is solved by applying the theorem repeatedly to reduce the input network, until an equivalent single-node terminal network is obtained. The constraints of the terminal network have a simple characterization, which is expanded, based on the same equivalence theorem, to form a computationally tractable representation of the constraints of the input network. Hence, our algorithm is called the *tree reduction/expansion method*.

Our algorithm bears similarity to an idea that was proposed in [3] to solve feasibility problems on radial networks iteratively performing a pair of tree traversals (sweeps) which propagate information: from the root to the leaves and from the leaves to the root. The idea was further developed into a family of algorithms known as forward-backward sweep methods. Readers are referred to [5, 6, 13] for more examples and to [8] for an extensive survey. Despite

the apparent resemblance, our method differs from the forward-backward sweep methods in several key aspects.

First, the above-mentioned methods solve feasibility problems, while we solve an optimization problem. In addition, these methods are iterative in nature, and perform several sweeps of the network until convergence. Our method, on the other hand, is not iterative, and it characterizes the entire feasible set of the given problem in a single sweep. Finally, to the best of our knowledge, these iterative methods have no mathematically proven guarantee of convergence to a feasible solution, despite empirical results which demonstrate their performance on many examples. In contrast, our method is backed by a theorem that ensures, under mild assumptions, that our method is correct up to user-controlled approximation error.

1.1 Paper layout

First, in Sect. 2 we explain in detail what are power systems and how they are mathematically modeled. Section 3 defines the mathematical notation and formalism we use to compactly describe power networks. Then, we define the OPF problem in terms of this notation, and state the assumptions required for our algorithm. In Sect. 4 we state the main theorem of this paper, called the “tree reduction theorem”, which establishes an equivalence between a given network and a smaller network leading to the development of the tree reduction/expansion method. Finally, Sect. 5 is devoted to the proof of the tree reduction theorem, and in Sect. 6 we present the results of several numerical experiments, which serve as an empirical evidence for our claim of reliably computing the global optimum.

1.2 Mathematical notation

Italics, e.g. a, b, V, I , are used for scalars. Capital italics are also used to describe sets. Bold letters, e.g. $\mathbf{p}, \mathbf{q}, \mathbf{Y}$, are used for column vectors and matrices; Indexed italics, e.g. Y_{ij} , are used to denote scalars or matrix/vector elements. The sets of real and complex numbers are denoted by \mathbb{R} and \mathbb{C} respectively and the set of natural numbers (excluding 0) is denoted by \mathbb{N} . The symbol i is the complex imaginary unit, e.g. $i^2 = -1$. We use $\text{re}(z)$, $\text{im}(z)$ to denote the real and imaginary parts of $z \in \mathbb{C}$, and $|z|$, $\arg(z)$ to denote its absolute value and angle (argument) in radians. For vectors and matrices $\text{re}(\cdot)$, $\text{im}(\cdot)$, $|\cdot|$, and $\arg(\cdot)$ are naturally extended to be component-wise operations. We also use \mathbf{e} to denote the vector of ones, and $\mathbf{0}$ to denote the zeros vector.

We also define some notation related to an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with \mathcal{V} being the node set and \mathcal{E} being the edge set. The edge connecting nodes k and j (if exists) is denoted by $\{k, j\} \in \mathcal{E}$. In addition, for each $k \in \mathcal{V}$ we denote by N_k the set of the neighbors of k . That is, $N_k = \{j \in \mathcal{V} : \{k, j\} \in \mathcal{E}\}$. A tree $T = (\mathcal{V}, \mathcal{E})$ is an undirected graph without cycles. If the tree is rooted at some node, then parent-child relationships are automatically defined. We denote by $\mathbf{root}(T)$ the root node of T , by $\mathbf{Pa}_T(j)$ the parent of node $j \in \mathcal{V}$, by $\mathbf{Ch}_T(j)$ the set of (direct) children of node $j \in \mathcal{V}$. We also denote by $\mathbf{L}(T)$ the set of leaves of a rooted tree T , which are nodes without children: $\mathbf{L}(T) = \{k \in \mathcal{V} : \mathbf{Ch}_T(k) = \emptyset\}$. Note that the root of the tree may be a leaf in the graph-theoretic sense, but is usually not a leaf in the rooted tree sense. Specifically, $\mathbf{root}(T) \in \mathbf{L}(T)$ if and only if $\mathbf{root}(T)$ is the only node of T .

Finally, we define some notation related to functions. For a function $f : S \rightarrow D$, we denote its domain by $\mathbf{dom}(f) := S$ and its image by

$$\mathbf{image}(f) \equiv \{f(x) : x \in \mathbf{dom}(f)\}.$$

For a set of functions f_1, \dots, f_m we extend the operator **image**(\cdot) as follows:

$$\mathbf{image}(f_1, \dots, f_m) \equiv \{(f_1(x), \dots, f_m(x)) : x \in \mathbf{dom}(f_1) \cap \dots \cap \mathbf{dom}(f_m)\}.$$

Any function f which maps scalars to scalars is extended to operate on vectors in a component-wise manner. That is, given a vector $\mathbf{p} \in \mathbb{R}^n$ the expression $f(\mathbf{p})$ is a vector in \mathbb{R}^n such that $(f(\mathbf{p}))_i = f(p_i)$. The scalar fraction operator is also extended in a similar manner, and for two vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^n$ the expression $\frac{\mathbf{p}}{\mathbf{q}}$ is a vector of component-wise fractions. That is, $(\frac{\mathbf{p}}{\mathbf{q}})_i = \frac{p_i}{q_i}$. The power, absolute value, and complex conjugate operations are extended in a similar manner.

2 Power system modeling

An electric power system is a physically large electric circuit consisting of sources of energy and consumers of energy (generators and loads) that are interconnected through transmission and distribution networks. One can think of such a system as a graph $G = (\mathcal{V}, \mathcal{E})$ consisting of nodes and edges. The nodes refer to the network's *buses*, and represent points of equal voltage. Edges represent the network transmission lines. Each generator or load in the system is connected to a specific node. We refer to G as the *network topology* graph. Unless the nodes of a power network's topology graph G are defined otherwise, we use the convention that $\mathcal{V} = \{1, 2, \dots, n\}$ for some positive integer n .

The voltages and currents in the system are typically sinusoidal (AC) signals with frequency of nearly 50 or 60 Hz. The magnitude and phase of these sinusoidal signals are represented by complex numbers called *phasors*. In this paper, $v_k \in \mathbb{C}$ is the voltage phasor at node k , and $I_k \in \mathbb{C}$ is a phasor describing the current injected into node k . Additional quantities associated with each node k are its *active power* $p_k \in \mathbb{R}$ and *reactive power* $q_k \in \mathbb{R}$. These are combined to form the *apparent power* $s_k = p_k + iq_k \in \mathbb{C}$. Associated with each edge $\{k, j\} \in \mathcal{E}$ is the *impedance* z_{kj} , which is a complex number that measures the conductivity of the power transmission line. Figure 1 illustrates a simple power network and its corresponding graph.

Balanced three-phase networks are characterized by a matrix \mathbf{Y} , named the *nodal admittance matrix*. This matrix expresses the relationships between node voltages and currents via the following set of equations:

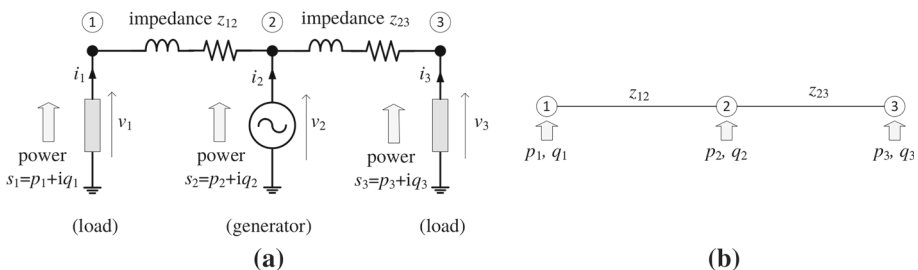


Fig. 1 **a** Electric circuit representing one phase in a balanced three-phase power system. Loads are connected to nodes 1, 3, and a synchronous generator is connected to node 2. **b** An equivalent graph, showing nodes, edges, and injected powers

$$I_k = \sum_{j=1}^n Y_{k,j} v_j, \quad k = 1, \dots, n \tag{1}$$

Taking the conjugate of both sides of (1), multiplying both sides by v_k , and using the power formula $s_k = v_k I_k^*$ results in:

$$s_k = p_k + iq_k = \sum_{j=1}^n v_k v_j^* Y_{k,j}^*, \quad k = 1, 2, \dots, n. \tag{2}$$

These equalities are known as the *power flow equations* (PFE). In case the system is represented by per-unit quantities, such that all transformers have a ratio of 1:1, and all shunt elements are modeled as part of generators or loads, the admittance matrix is given by

$$Y_{k,j} = \begin{cases} -\frac{1}{z_{kj}}, & \{k, j\} \in \mathcal{E}, \\ \sum_{\ell: \{k, \ell\} \in \mathcal{E}} \frac{1}{z_{k\ell}}, & k = j, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

An *impedance function* $Z : \mathcal{E} \rightarrow \mathbb{C}$ is a function that assigns an impedance to each edge. For consistency and readability, we use the indexed notation z_{kj} to denote $Z(\{k, j\})$. The power flow equations defined in (2) are a key physical property of any power transmission system, and therefore for any graph G with n nodes and a corresponding impedance function Z we define the set $\text{PFE}(G, Z)$ to be the set of voltages and powers associated with all nodes which satisfy the power-flow equations. Formally,

$$\text{PFE}(G, Z) \equiv \left\{ (\mathbf{v}, \mathbf{s}) \in \mathbb{C}^n \times \mathbb{C}^n : s_k = \sum_{j=1}^n v_k v_j^* Y_{k,j}^*, \quad k = 1, 2, \dots, n \right\},$$

where $Y_{k,j}$ are defined in (3). The quadratic transformation that maps the voltages \mathbf{v} to the powers \mathbf{s} is denoted by \mathcal{Q}_Z :

$$\mathcal{Q}_Z(\mathbf{v}) \equiv \left(\sum_{j=1}^n v_k v_j^* Y_{k,j}^* \right)_{k=1}^n.$$

In this notation, we can rewrite the set $\text{PFE}(G, Z)$ as

$$\text{PFE}(G, Z) = \{(\mathbf{v}, \mathbf{s}) \in \mathbb{C}^n \times \mathbb{C}^n : \mathbf{s} = \mathcal{Q}_Z(\mathbf{v})\}.$$

3 Terminology and problem statement

In this paper we present an algorithm for finding a global optimal solution of a special, yet challenging, case of the *optimal power flow* problem. In this section we describe the basic formalism used in the paper, define the optimal power flow problem in terms of this formalism, and finally, present the class of *restricted radial networks* for which the optimal power flow problem will be shown to be tractable under some verifiable conditions.

3.1 Power networks

We now define the concept of a *power network* that is described by the topology graph of a power transmission system, as well as additional constraint sets corresponding to each of the nodes.

Definition 1 (*power network*) The tuple $P = (G, Z, \{\mathcal{C}_k\}_{k \in \mathcal{V}})$ is called a **power network** if:

- $G = (\mathcal{V}, \mathcal{E})$ is an undirected graph with $\mathcal{V} \subseteq \mathbb{N}$ and $1 \in \mathcal{V}$;
- $Z : \mathcal{E} \rightarrow \mathbb{C}$ is the **impedance function** of G ;
- For any $k \in \mathcal{V}$ we have $\mathcal{C}_k \subseteq \mathbb{R} \times \mathbb{C}$. Each set \mathcal{C}_k is called the **constraint set associated with node k** .

We use the sets $\{\mathcal{C}_k\}_{k \in \mathcal{V}}$ of a power network to constrain the network’s capacity to produce and consume power by constraining the voltage absolute value and the power injected into each node. In addition, we designate node “1” as the *reference node*, for which the voltage angle is zero. We are now ready to define formally the notions of *feasible pairs* and *feasible vectors* of a power network. Note that since the powers and voltages satisfy the relation $\mathbf{s} = \mathcal{Q}_Z(\mathbf{v})$, feasibility can also be defined solely with respect to the voltages vector.

Definition 2 (*feasible pairs and vectors*) Let $P = (G, Z, \{\mathcal{C}_k\}_{k \in \mathcal{V}})$ be a power network with $\mathcal{V} = \{1, 2, \dots, n\}$.

- A pair of vectors $(\mathbf{v}, \mathbf{s}) \in \mathbb{C}^n \times \mathbb{C}^n$ is called a **feasible pair of the network P** if

$$\begin{aligned} \arg(v_1) &= 0, \\ (\mathbf{v}, \mathbf{s}) &\in \text{PFE}(G, Z), \\ (|v_k|, s_k) &\in \mathcal{C}_k, \quad k \in \mathcal{V}. \end{aligned}$$

- A vector $\mathbf{v} \in \mathbb{C}^n$ is called a **feasible vector of the network P** if

$$\begin{aligned} \arg(v_1) &= 0, \\ (|v_k|, (\mathcal{Q}_Z(\mathbf{v}))_k) &\in \mathcal{C}_k, \quad k \in \mathcal{V}. \end{aligned}$$

- The **set of all feasible pairs of the network P** is denoted by $\mathcal{F}_{vs}(P)$, and formally defined by

$$\mathcal{F}_{vs}(P) \equiv \{(\mathbf{v}, \mathbf{s}) \in \mathbb{C}^n \times \mathbb{C}^n : \arg(v_1) = 0, (\mathbf{v}, \mathbf{s}) \in \text{PFE}(G, Z), (|v_k|, s_k) \in \mathcal{C}_k, k \in \mathcal{V}\}.$$

- The **set of all feasible vectors of the network P** is denoted by $\mathcal{F}_v(P)$, and formally defined by

$$\mathcal{F}_v(P) \equiv \{\mathbf{v} \in \mathbb{C}^n : \arg(v_1) = 0, (|v_k|, (\mathcal{Q}_Z(\mathbf{v}))_k) \in \mathcal{C}_k, k \in \mathcal{V}\}.$$

Obviously \mathbf{v} is a feasible vector of P if and only if $(\mathbf{v}, \mathcal{Q}_Z(\mathbf{v}))$ is a feasible pair and the two feasible sets $\mathcal{F}_v(P)$ and $\mathcal{F}_{vs}(P)$ are connected by the relation

$$\mathcal{F}_v(P) = \{\mathbf{v} \in \mathbb{C}^n : (\mathbf{v}, \mathcal{Q}_Z(\mathbf{v})) \in \mathcal{F}_{vs}(P)\}.$$

When we list the sets C_k directly inside the tuple describing the power network, we implicitly assume that their indices are the vertices of the topology graph. For example, when we denote a power network by (G, Z, C_1, \dots, C_n) , it is assumed that the set of vertices of G is $\{1, \dots, n\}$.

The following node constraint types are widely used in power-flow problems:

PQ constraint If

$$C_k = [V_k^{min}, V_k^{max}] \times \{\hat{s}_k\},$$

where $\hat{s}_k \in \mathbb{C}$, $V_k^{min} \in \mathbb{R}_+$, and $V_k^{max} \in \mathbb{R}_{++} \cup \{+\infty\}$ are given constants, then C_k is called a *PQ constraint*. That is, with PQ constraints we exactly specify the apparent power, while constraining the voltage absolute value to lie in a given interval, which may be unbounded.

PV constraint If

$$C_k = \{\hat{v}_k\} \times \{s \in \mathbb{C} : \text{re}(s) = \hat{p}_k, \text{im}(s) \in [Q_k^{min}, Q_k^{max}]\},$$

where $\hat{v}_k \in \mathbb{R}_{++}$, $\hat{p}_k \in \mathbb{R}$, $Q_k^{min} \in \mathbb{R} \cup \{-\infty\}$, and $Q_k^{max} \in \mathbb{R} \cup \{+\infty\}$ are given constants, then C_k is called a *PV constraint*. That is, with PV constraints we exactly specify the voltage absolute value and active power, while constraining the reactive power to lie in a given interval, which may be unbounded.

3.2 The power flow problem

In the power-flow problem on a power network P , the goal is to either find $(\mathbf{v}, \mathbf{s}) \in \mathcal{F}_{vs}(P)$ or deduce that $\mathcal{F}_{vs}(P) = \emptyset$. In most cases, the sets C_k are chosen such that all the equality constraints defining $\mathcal{F}_{vs}(P)$ comprise $4n$ equations in $4n$ variables (n being the number of nodes). One particular widely used choice is:

- For any $k \in \mathcal{V}$, $k \neq 1$, the set C_k is either a PV or a PQ constraint.
- The set C_1 is defined by

$$C_1 = \{\hat{v}_1\} \times \{s \in \mathbb{C} : \text{re}(s) \in [P_1^{min}, P_1^{max}], \text{im}(s) \in [Q_1^{min}, Q_1^{max}]\}.$$

In other words, the voltage absolute value is exactly specified, while the active and reactive powers are constrained to an interval.

3.3 The optimal power flow problem

The purpose of the optimal power flow problem on a power network $P = (G, Z, C_1, \dots, C_n)$ is to minimize a certain objective function of $(\mathbf{v}, \mathbf{s}) \in \mathbb{C}^n \times \mathbb{C}^n$, subject to the physical constraints of the system, described by $\mathcal{F}_{vs}(P)$ and some additional constraints. The objective function might, for example, include elements related to the actual cost of energy production, stability and reliability. In most cases of interest, the equality constraints define less than $4n$ equations, to leave some degrees of freedom. Formally, the optimal power-flow problem is described below.

The Optimal Power Flow Problem

Problem data:

- a power network $P = (G, Z, C_1, \dots, C_n)$;
- the objective function $f : \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{R}$ that assigns a cost $f(\mathbf{v}, \mathbf{s})$ to each instance of the voltage and power vectors \mathbf{v}, \mathbf{s} ;
- a set of additional constraints $\mathcal{C} \subseteq \mathbb{C}^n \times \mathbb{C}^n$.

Objective: find an optimal solution, if one exists, of the following optimization problem:

$$\begin{aligned} \min_{\mathbf{v}, \mathbf{s} \in \mathbb{C}^n} \quad & f(\mathbf{v}, \mathbf{s}) && \text{(OPF)} \\ \text{s.t.} \quad & (\mathbf{v}, \mathbf{s}) \in \mathcal{F}_{vs}(P), \\ & (\mathbf{v}, \mathbf{s}) \in \mathcal{C}. \end{aligned}$$

3.4 Restricted radial networks

The OPF problem is a difficult nonconvex problem since, for example, it includes quadratic equality constraints. Thus, it seems to be an intractable problem in general. In this paper we will show how to solve the problem for a class of power networks with a tree topology which we call *restricted radial networks*.

Definition 3 A power network $P = (T, Z, \{C_k\}_{k \in \mathcal{V}})$ is called a **restricted radial network** if the following conditions hold:

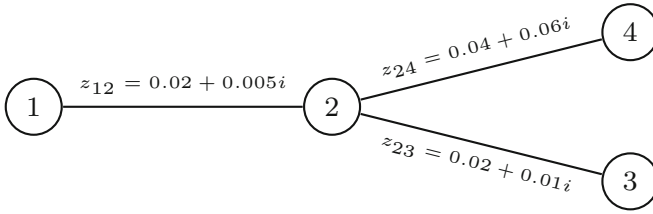
- $T = (\mathcal{V}, \mathcal{E})$ is a rooted tree with $\text{root}(T) = 1$ and $|T| \geq 2$;
- for any $k \in \mathcal{V} \setminus \mathbf{L}(T)$, $k \neq 1$ the set C_k is a PQ constraint;
- for any $k \in \mathbf{L}(T)$ the set C_k is either a PQ or a PV constraint. The corresponding interval, $[V_k^{\min}, V_k^{\max}]$ for a PQ constraint or $[Q_k^{\min}, Q_k^{\max}]$ for a PV constraint, is bounded;
- the root constraint set is $C_1 = [V_1^{\min}, V_1^{\max}] \times \mathbb{C}$, where $V_1^{\min} \in \mathbb{R}_+$ and $V_1^{\max} \in \mathbb{R}_{++} \cup \{+\infty\}$ are given constants. That is, the root voltage is constrained to lie in a (possibly unbounded) interval, while the root injected power is unconstrained.

The network in Fig. 2 is an example of a restricted radial network. The goal of this paper is to describe an efficient algorithm for solving the OPF problem on restricted radial networks. The restriction that PV constrained nodes cannot be non-leaves was introduced to simplify the presentation of our algorithm as a concept, and indeed in Sect. 4.8 show how this restriction can be eliminated.

4 The tree reduction/expansion method

In contrast to popular methods, our algorithm for solving the OPF problem on a restricted radial network P does not, by nature, generate a recurrent sequence which converges to the optimal solution. Instead, our algorithm is based on a technique that essentially fully characterizes the feasible set $\mathcal{F}_v(P)$ (or $\mathcal{F}_{vs}(P)$) and then picks out of this set a vector (or vectors) corresponding to the minimal objective function value which also satisfies the additional constraints described in \mathcal{C} .

The method comprises two stages. The aim of the first stage is to find a representation of $\mathcal{F}_v(P)$ that will enable us to evaluate all the points in $\mathcal{F}_v(P)$ up to some discretization. This is done by a sequence of reductions of the power network up to the point that we



$\mathcal{C}_1 = [0.97, \infty] \times \mathbb{C}$
 \nwarrow Root constraint: $V_1^{min} = 0.97, V_1^{max} = \infty$
 $\mathcal{C}_2 = [0.9, 1.1] \times \{-0.2 - 0.1i\}$
 \nwarrow PQ constraint: $V_2^{min} = 0.9, V_2^{max} = 1.1, \hat{s}_2 = -0.2 - 0.1i$
 $\mathcal{C}_3 = [0.9, 1.1] \times \{-0.4 - 0.3i\}$
 \nwarrow PQ constraint: $V_3^{min} = 0.9, V_3^{max} = 1.1, \hat{s}_2 = -0.4 - 0.3i$
 $\mathcal{C}_4 = \{1\} \times \{s \in \mathbb{C} : \text{re}(s) = 0.25, \text{im}(s) \in [-1, 1]\}$
 \nwarrow PV constraint: $\hat{v}_4 = 1, \hat{p} = 0.25, Q_4^{min} = -1, Q_4^{max} = 1$

Fig. 2 Example of a restricted radial network

reach an elementary network with a single node. This sequence of reductions is done by the *tree reduction method*. In the second stage, we use the outcomes of the first stage in order to find all points in $\mathcal{F}_v(P)$ up to some discretization, using a method that we call *the tree expansion method*, and pick the optimal solution out of these points. The combination of the two methods is the *tree reduction/expansion method*.

This section is devoted to the derivation of the tree reduction/expansion method and its practical implementation. We derive our method under a technical assumption on the network, which somewhat limits the applicability of our algorithm. Then, we show how this assumption can be substantially weakened, making our algorithm applicable to almost any restricted radial network.

4.1 Tree reductions

We begin with the definition of a *tree reduction*, illustrated in Fig. 3.

Definition 4 (*Tree reduction*)

1. A node $j \in \mathcal{V}$ in a rooted tree $T = (\mathcal{V}, \mathcal{E})$ is **reducible** if $\text{Ch}_T(j) \subseteq \mathbf{L}(T)$, and $\text{Ch}_T(j) \neq \emptyset$. That is, j 's children form a non-empty set of leaves.
2. A rooted tree T is a **reduction of a rooted tree S via the node j** , if j is reducible in S and T was constructed by removing $\text{Ch}_S(j)$ and all their associated edges from S . A tree T is a reduction of S if it is a reduction of S via some reducible node of S .
3. A tree T is an **indirect reduction of a rooted tree S** , if there exists a sequence of rooted trees $S = T_1, T_2, \dots, T_m = T$ such that T_{k+1} is a reduction of T_k for any $k = 1, 2, \dots, m - 1$.

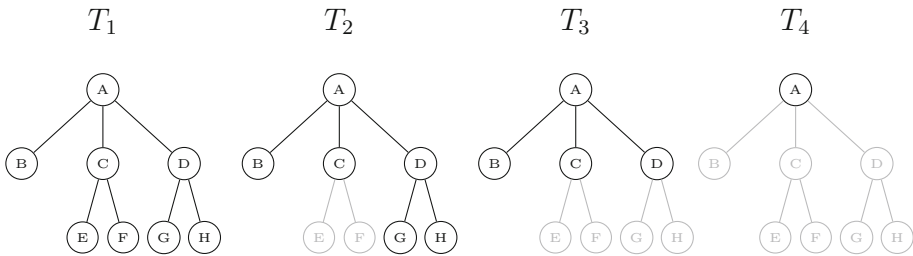


Fig. 3 A sequence of reductions. Grayed nodes and edges were discarded by the reductions. T_1 consists of the nodes $\{A, B, C, D, E, F, G, H\}$ and rooted at A . T_2 is a reduction of T_1 via C . T_3 is a reduction of T_2 via D . Finally, T_4 is a reduction of T_3 via A . Also, for any $k < j$ the tree T_j is an indirect reduction of T_k

4.2 Curved radial networks

Another step that is required to derive the tree reduction method is to define a slight generalization of restricted radial networks.

Definition 5 (*Curved Radial Network*) A power network $P = (T, Z, \{C_k\}_{k \in \mathcal{V}})$ is a **curved radial network** (or a CRN) if the following conditions hold:

- $T = (\mathcal{V}, \mathcal{E})$ is a rooted tree with $\mathbf{root}(T) = 1$.
- For any $k \neq 1, k \notin \mathbf{L}(T)$ the set C_k is a PQ constraint.
- For any $k \in \mathbf{L}(T)$ we have

$$C_k = \mathbf{image}(v_k, \sigma_k),$$

where $v_k : [0, 1] \rightarrow \mathbb{R}_{++}$ and $\sigma_k : [0, 1] \rightarrow \mathbb{C}$ are given continuous functions. That is, for any leaf k , the set C_k is a continuous one-dimensional curve in $\mathbb{R} \times \mathbb{C}$.

- If $1 \notin \mathbf{L}(T)$, the root constraint is $C_1 = [V_1^{min}, V_1^{max}] \times \mathbb{C}$, where $V_1^{min} \in \mathbb{R}_+$ and $V_1^{max} \in \mathbb{R}_{++} \cup \{+\infty\}$ are given constants. That is, the root voltage is constrained to lie in a (possibly unbounded) interval.

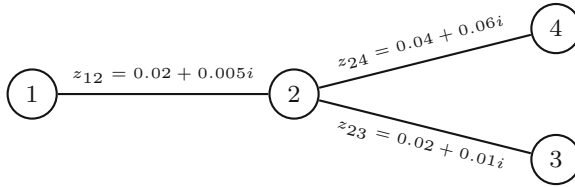
To see why a CRN is indeed a generalization of a restricted radial network, we point out two observations. First, in a CRN we allow $|T| = 1$, while in a restricted radial network we require $|T| \geq 2$. Second, when $|T| \geq 2$, the only difference between the two network types is in the definition of C_k for $k \in \mathbf{L}(T)$ as one-dimensional curves. However, these sets in a restricted radial network are essentially one-dimensional line segments, and thus can be represented as one-dimensional curves. Indeed, if $(T, Z, \{C_k\}_{k \in \mathcal{V}})$ is a restricted radial network, then for any $k \in \mathbf{L}(T)$, we can represent the constraint set C_k as $C_k = \mathbf{image}(v_k, \sigma_k)$ with the following functions for PQ constraints (recalling that V_k^{min}, V_k^{max} are finite real numbers):

$$v_k(t) = (1 - t)V_k^{min} + tV_k^{max}, \quad \sigma_k(t) = \hat{s}_k,$$

and the following functions for PV constraints (recalling that Q_k^{min}, Q_k^{max} are finite real numbers):

$$v_k(t) = \hat{v}_k, \quad \sigma_k(t) = \hat{p}_k + i \left[(1 - t)Q_k^{min} + tQ_k^{max} \right].$$

For example, applying the transformation described above to the network in Fig. 2 results in the CRN illustrated in Fig. 4.



$C_1 = [0.97, \infty] \times \mathbb{C}$	
$C_2 = [0.9, 1.1] \times \{-0.2 - 0.1i\}$	
$C_3 = \text{image}(\nu_3, \sigma_3)$	$\nu_3(t) = 0.9 + 0.2t, \sigma_3(t) = -0.4 - 0.3i$
$C_4 = \text{image}(\nu_4, \sigma_4)$	$\nu_4(t) = 1, \sigma_4(t) = 0.25 + i \cdot [-1 + 2t]$

Fig. 4 A CRN equivalent to the network in Fig. 2

4.3 The tree reduction theorem

The next theorem is the main result in the paper and constitutes the theoretical justification for the tree reduction/expansion method. The result establishes a relationship between a CRN with some topology tree T , and a smaller CRN whose topology graph is a reduction of T . For readability reasons, its proof is provided in a separate section later in this paper (Sect. 5).

Theorem 1 (the tree reduction theorem) *Let $P = (T, Z, C_1, \dots, C_n)$ be a CRN with $n \geq 2$. Let j be a reducible node in T , and assume w.l.o.g. that $\mathbf{Ch}_T(j) = \{j + 1, \dots, n\}$. Let:*

- T' be the reduction of T via j ;
- $\{\nu_k, \sigma_k\}_{k=j+1}^n$ be the functions describing the curves $C_k, k = j + 1, \dots, n$;
- $V_j^{min}, V_j^{max}, \hat{s}_j$ be the scalars for which $C_j = [V_j^{min}, V_j^{max}] \times \{\hat{s}_j\}$.

Define the functions $\tilde{\nu}_k : [0, 1] \rightarrow \mathbb{R}_+$ and $\tilde{\sigma}_k : [0, 1] \rightarrow \mathbb{C}$, and the set U_j by:

$$\tilde{\nu}_k(t) = \left| \nu_k(t) - \frac{(\sigma_k(t))^* z_{kj}}{\nu_k(t)} \right|, \quad k = j + 1, \dots, n, \tag{4}$$

$$\tilde{\sigma}_k(t) = \sigma_k(t) - z_{kj} \frac{|\sigma_k(t)|^2}{(\nu_k(t))^2}, \quad k = j + 1, \dots, n, \tag{5}$$

$$U_j = [V_j^{min}, V_j^{max}] \cap \left(\bigcap_{k=j+1}^n \text{image}(\tilde{\nu}_k) \right). \tag{6}$$

Assume that the functions $\tilde{\nu}_k$ are invertible, and let

$$\phi_k \equiv \tilde{\sigma}_k \circ \tilde{\nu}_k^{-1}. \tag{7}$$

If $U_j = \emptyset$, then $\mathcal{F}_v(P) = \emptyset$. Otherwise, set $[v_j^{min}, v_j^{max}] = U_j$ and we have that $\mathbf{v} \in \mathcal{F}_v(P)$ if and only if all of the following hold:

1. $\mathbf{v}' = (v_1, \dots, v_j)^T \in \mathcal{F}_v(P')$, where $P' = (T', Z', C_1, \dots, C_{j-1}, C'_j)$ is a CRN in which T' is a reduction of T via j , Z' is the restriction of the function Z on the edge set of T' and the curve C'_j is defined by the following pair of functions:

$$v_j(t) = (1 - t) \cdot v_j^{min} + t \cdot v_j^{max}, \tag{8}$$

$$\sigma_j(t) = \begin{cases} \hat{s}_j + \sum_{k=j+1}^n (\phi_k \circ v_j)(t), & j \neq 1, \\ 0, & j = 1; \end{cases} \tag{9}$$

2. for any $k = j + 1, \dots, n$ we have

$$v_k = v_j + z_{kj} \left(\frac{\phi_k(|v_j|)}{v_j} \right)^*. \tag{10}$$

The tree reduction theorem essentially states that the problem of finding the feasible vectors of a given CRN $P = (T, Z, C_1, \dots, C_n)$ can be reduced into the problem of finding the feasible vectors of a smaller CRN $P' = (T', Z', C_1, \dots, C_{j-1}, C'_j)$. Assuming that we found the set of feasible vectors of P' , we can find the voltages of the larger CRN by using the functions ϕ_k given in (7). These functions are called the *voltage transfer functions* since they allow us, using Eq. (10), to map the voltage of the node through which the reduction was performed to the voltages of its children nodes.

4.3.1 Example

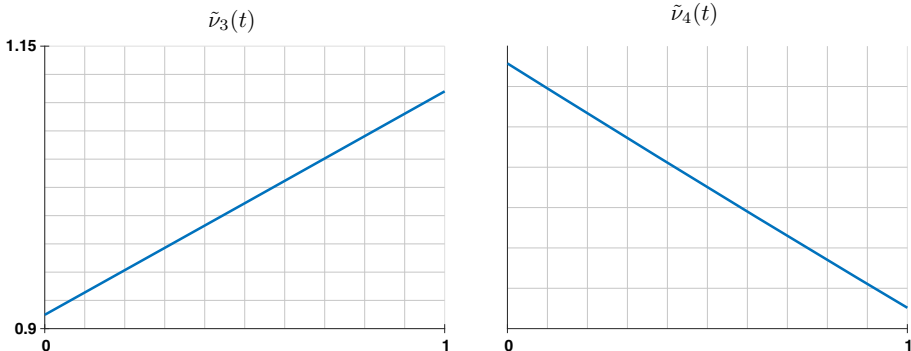
We illustrate the reduction described in Theorem 1 on the CRN described in Fig. 4. We choose the reducible node $j = 2$, for which we have $\mathbf{Ch}_T(2) = \{3, 4\}$. We begin by computing the functions $\tilde{v}_3, \tilde{\sigma}_3, \tilde{v}_4, \tilde{\sigma}_4$, verify that \tilde{v}_3 and \tilde{v}_4 are invertible, and then proceed to compute the set U_2 . The formulas in (4) and (5) with $j = 2$ and $k = 3$ yield:

$$\begin{aligned} \tilde{v}_3(t) &= \left| v_3(t) - \frac{(\sigma_3(t))^* z_{23}}{v_3(t)} \right| = \left| 0.9 + 0.2t - \frac{(-0.4 + 0.3i)(0.02 + 0.01i)}{0.9 + 0.2t} \right| \\ &= \sqrt{\frac{0.000125}{(0.2t + 0.9)^2} + (0.2t + 0.9)^2 + 0.022}, \\ \tilde{\sigma}_3(t) &= \sigma_3(t) - z_{23} \frac{|\sigma_3(t)|^2}{(v_3(t))^2} = -0.4 - 0.3i - (0.02 + 0.01i) \frac{0.25}{(0.9 + 0.2t)^2} \\ &= \left[-\frac{0.005}{(0.2t + 0.9)^2} - 0.4 \right] + i \times \left[-\frac{0.0025}{(0.2t + 0.9)^2} - 0.3 \right]. \end{aligned} \tag{11}$$

Similarly, computing \tilde{v}_4 and $\tilde{\sigma}_4$ yields:

$$\begin{aligned} \tilde{v}_4(t) &= \sqrt{0.0208t^2 - 0.2608t + 1.105525}, \\ \tilde{\sigma}_4(t) &= (-0.16t^2 + 0.16t + 0.2075) + i \cdot (-0.24t^2 + 2.24t - 1.06375). \end{aligned}$$

It is easy to show that \tilde{v}_3 and \tilde{v}_4 are monotone; this can also be illustrated graphically:



Since \tilde{v}_3 is increasing and \tilde{v}_4 is decreasing, we have

$$\begin{aligned} \mathbf{image}(\tilde{v}_3) &= [\tilde{v}_3(0), \tilde{v}_3(1)] = [0.9122, 1.1100], \\ \mathbf{image}(\tilde{v}_4) &= [\tilde{v}_4(1), \tilde{v}_4(0)] = [0.9303, 1.0514]. \end{aligned}$$

Using Eq. (6) we compute:

$$U_2 = [0.9, 1.1] \cap [0.9122, 1.1100] \cap [0.9303, 1.0514] = [0.9303, 1.0514]. \tag{12}$$

The assumptions of the tree reduction theorem indeed hold (invertibility of \tilde{v}_3 and \tilde{v}_4), and therefore $(v_1, v_2, v_3, v_4)^T \in \mathcal{F}_v(P)$ if and only if:

- $(v_1, v_2)^T \in \mathcal{F}_v(P')$, where $P' = (T', Z, C_1, C'_2)$, T' is the reduction of T via node $j = 2$, and $C'_2 = \mathbf{image}(v_2, \sigma_2)$ with v_2, σ_2 defined by

$$\begin{aligned} v_2(t) &= v_2^{min} \cdot (1 - t) + v_2^{max} \cdot t = 0.9303 \cdot (1 - t) + 1.0514 \cdot t \\ \sigma_2(t) &= \hat{\sigma}_2 + \phi_3(v_2(t)) + \phi_4(v_2(t)), \end{aligned}$$

where $\phi_3 = \tilde{\sigma}_3 \circ \tilde{v}_3^{-1}$ and $\phi_4 = \tilde{\sigma}_4 \circ \tilde{v}_4^{-1}$.

- We have

$$\begin{aligned} v_3 &= v_2 + z_{23} \left(\frac{\phi_3(|v_2|)}{v_2} \right)^*, \\ v_4 &= v_2 + z_{24} \left(\frac{\phi_4(|v_2|)}{v_2} \right)^*. \end{aligned}$$

Next, we need to compute ϕ_3 and ϕ_4 , which participate in the conclusion above. Let $v \in \mathbb{R}_+$; we will compute $\phi_3(v) = (\tilde{\sigma}_3 \circ \tilde{v}_3^{-1})(v)$. Denoting $t = \tilde{v}_3^{-1}(v)$, we have that $\tilde{v}_3(t) = v$, meaning that:

$$\sqrt{\frac{0.000125}{(0.2t + 0.9)^2} + (0.2t + 0.9)^2 + 0.022} = v,$$

which readily implies that

$$(0.9 + 0.2t)^2 = \frac{v^2 - 0.022 + \sqrt{v^4 - 0.044v^2 - 0.000016}}{2}.$$

Therefore, taking into account the expression for $\tilde{\sigma}_3$ given in (11),

$$\begin{aligned} \phi_3(v) = \tilde{\sigma}_3(t) &= \left[-0.4 - \frac{0.1}{v^2 - 0.022 + \sqrt{v^4 - 0.044v^2 - 0.000016}} \right] \\ &+ i \left[-0.3 - \frac{0.05}{v^2 - 0.022 + \sqrt{v^4 - 0.044v^2 - 0.000016}} \right] \end{aligned}$$

In a similar manner, ϕ_4 is computed from \tilde{v}_4 and $\tilde{\sigma}_4$:

$$\begin{aligned} \phi_4(v) &= 0.017751\sqrt{520000.0v^2 - 149777.0} - 7.6923v^2 - 2.8624 \\ &+ i \times (0.0073964\sqrt{520000.0v^2 - 149777.0} - 11.538v^2 + 6.8698). \end{aligned}$$

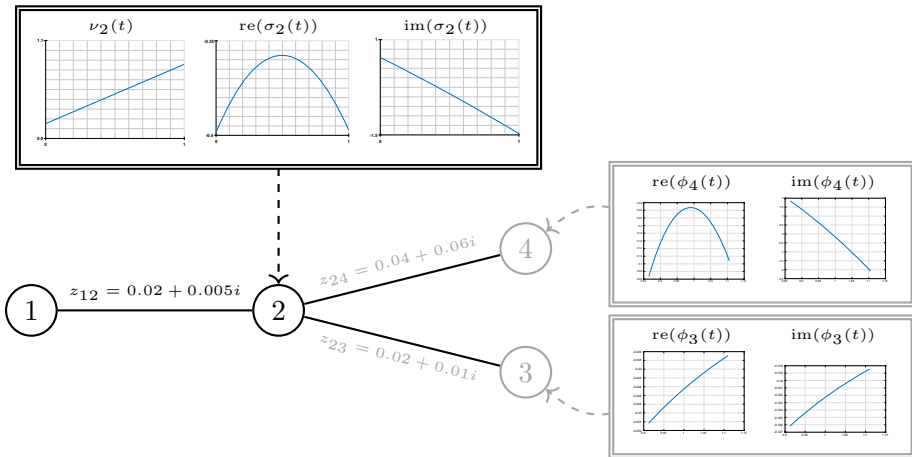


Fig. 5 The network $P' = (T', Z, C_1, C'_2)$, where T' are the black nodes and edges. The gray nodes and edges that were discarded by the reduction

To summarize, having expressed the functions $\phi_3, \phi_4, v_2, \sigma_2$, and remembering that $P' = (T, Z, C_1, C'_2)$ with $C'_2 = \mathbf{image}(v_2, \sigma_2)$, we have the following mathematical relationship:

$$\begin{aligned}
 (v_1, v_2, v_3, v_4)^T &\in \mathcal{F}_v(P) \\
 &\iff \\
 (v_1, v_2)^T &\in \mathcal{F}_v(P'), \tag{13} \\
 v_3 &= v_2 + z_{23} \left(\frac{\phi_3(|v_2|)}{v_2} \right)^*, \\
 v_4 &= v_2 + z_{24} \left(\frac{\phi_4(|v_2|)}{v_2} \right)^*.
 \end{aligned}$$

The formulas for v_3 and v_4 above follow from (10) for $j = 2$ and $k = 3, 4$. All the functions involved are graphically illustrated in Fig. 5.

4.4 The tree reduction method

The idea in the tree reduction method is to successively employ the reductions described in the tree reduction theorem (Theorem 1) in order to get a sequence of CRNs $P^{(0)} \equiv P, P^{(1)}, \dots, P^{(m)}$ until we reach the CRN $P^{(m)}$ consisting only of the single node 1. During this process, we compute the voltage transfer functions $\{\phi_k\}_{k=2}^n$ and the interval $[v_1^{min}, v_1^{max}]$ of possible values of the voltage at node 1 (recall that we assume that v_1 is real). In addition, we test for the nonemptiness of $\mathcal{F}_v(P)$ by observing that $\mathcal{F}_v(P) \neq \emptyset$ if and only if all of the sets U_j computed during this phase are nonempty. The set of feasible vectors of the final CRN $P^{(m)}$ (consisting of the single node 1) is extremely simple and is given by

$$\mathcal{F}_v(P^{(m)}) = [v_1^{min}, v_1^{max}].$$

The tree reduction method is formally summarized in Algorithm 1 below. We assume that we have a procedure named COMPUTE-REDUCTION-ORDER, which computes the sequence of

nodes u_1, \dots, u_w , with $u_w = 1$, via which the reductions of the original tree are performed. Such a procedure can be easily implemented, for instance, by employing a Breadth-First Search algorithm.

Algorithm 1 The Tree Reduction Method

Input: A CRN $P = (T, Z, C_1, \dots, C_n)$

Output: the reduction order $(u_1, \dots, u_w = 1)$; the functions $\{\phi_k\}_{k=2}^n$; the interval $[v_1^{min}, v_1^{max}]$ of feasible values of v_1 .

Steps:

```

1:  $u_1, \dots, u_w = 1 \leftarrow \text{COMPUTE-REDUCTION-ORDER}(T)$ 
2: for all  $j \in \{u_1, \dots, u_w\}$  do                                      $\triangleright$  Invariant:  $v_k, \sigma_k$  exist for  $k \in \text{Ch}_T(j)$ 
3:   for all  $k \in \text{Ch}_T(j)$  do
4:     Compute  $\tilde{v}_k$  as:  $\tilde{v}_k(t) \equiv \left| v_k(t) - z_{kj} \frac{(\sigma_k(t))^*}{v_k(t)} \right|$                                       $\triangleright$  Eq (4)
5:     Compute  $\tilde{\sigma}_k$  as:  $\tilde{\sigma}_k(t) \equiv \sigma_k(t) - z_{kj} \frac{|\sigma_k(t)|^2}{(v_k(t))^2}$                                       $\triangleright$  Eq (5)
6:     Compute  $\phi_k$  as:  $\phi_k \equiv \tilde{\sigma}_k \circ \tilde{v}_k^{-1}$ 
7:   end for
8:    $U_j \leftarrow [V_j^{min}, V_j^{max}] \cap \left( \bigcap_{k \in \text{Ch}_T(j)} \text{image}(\tilde{v}_k) \right)$                                       $\triangleright$  Eq (6)
9:   if  $U_j = \emptyset$  then
10:    Terminate with an error:  $\mathcal{F}_v(P) = \emptyset$ 
11:   else
12:     $[v_j^{min}, v_j^{max}] \leftarrow U_j$ 
13:   end if
14:   if  $j \neq 1$  then
15:     Compute  $v_j$  as:  $v_j(t) \equiv v_j^{min} \cdot (1 - t) + v_j^{max} \cdot t$                                       $\triangleright$  Eq (8)
16:     Compute  $\sigma_j$  as:  $\sigma_j(t) \equiv \hat{s}_j + \sum_{k \in \text{Ch}_T(j)} \phi_k(v_j(t))$                                       $\triangleright$  Eq (9)
17:   end if
18: end for

```

Note that the tree reduction method, as described above, is conceptual in nature, since it assumes that we have an efficient way to compute and represent the functions σ_k, v_k and ϕ_k . Section 4.7 will describe an implementable version of the method.

We can now formally state the technical assumption which must hold for a CRN P , and is essential for the correctness of the tree reduction method applied on P .

Assumption (Invertibility assumption) During the execution of the tree reduction method on the network P , the functions \tilde{v}_k computed on line 4 are invertible.

It does not seem to be possible to verify the validity of the invertibility assumption on a given CRN P a priori using only the data of the problem. However, this condition can be verified during the execution of the tree reduction method. Our numerical simulations suggest that when the impedances are small enough, the assumption holds. In any case, in Sect. 4.9 we describe a way to eliminate the above assumption.

4.4.1 Example continued

The example from Sect. 4.3.1 described the first reduction used in the tree reduction method in which the CRN P with the four nodes $\{1, 2, 3, 4\}$ was reduced into the CRN P' with the two nodes $\{1, 2\}$. The tree reduction method will conduct at this point a second reduction—a reduction of the CRN $P' = (T', Z', C_1, C'_2)$ via the node 1, which is reducible in T' . We have $\text{Ch}_{T'}(1) = \{2\}$. Skipping the tedious computations, the process of computing the functions

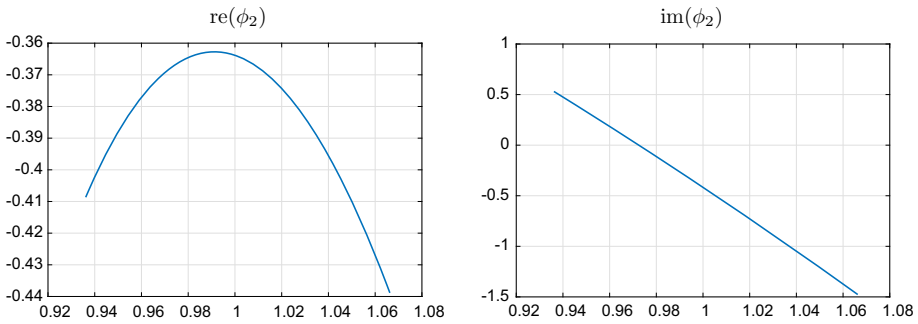


Fig. 6 Real and imaginary parts of ϕ_2

Algorithm 2 The Tree Expansion Method

Input: a CRN $P = (T, Z, C_1, \dots, C_n)$; two of the outputs of the tree reduction method – the reduction order $(u_1, \dots, u_w = 1)$, the voltage transfer functions $\{\phi_k\}_{k=2}^n$; a scalar $\alpha \in \mathbb{R}_+$

Output: a vector $(\mathbf{v}, \mathbf{s}) \in \mathcal{F}_{vS}(P)$ for which $v_1 = \alpha$.

Steps:

- 1: $v_1 \leftarrow \alpha$
 - 2: **for all** $j \in u_w, \dots, u_1$ **do** ▷ Invariant: v_j was already computed
 - 3: **for all** $k \in \text{Ch}_T(j)$ **do**
 - 4: $v_k \leftarrow v_j + z_{kj} \left(\frac{\phi_k(|v_j|)}{v_j} \right)^*$ ▷ Eq. (10)
 - 5: **end for**
 - 6: **end for**
 - 7: $\mathbf{s} \leftarrow \mathcal{Q}_Z(\mathbf{v})$
 - 8: **Return:** (\mathbf{v}, \mathbf{s})
-

\tilde{v}_2 and $\tilde{\sigma}_2$ and the nonempty set U_1 , verifying the invertibility of \tilde{v}_2 by observing that it is strictly increasing, computing the endpoints of the non-empty interval $U_1 = [v_1^{min}, v_1^{max}]$, and finally utilizing Eqs. (8) and (9) yields:

$$[v_1^{min}, v_1^{max}] = [0.97, \infty] \cap [\tilde{v}_2(0), \tilde{v}_2(1)] = [0.97, 1.0663].$$

In addition, computing $\phi_2 = \tilde{\sigma}_2 \circ \tilde{v}_2^{-1}$ results in the function described in Fig. 6. By defining $P'' = (T'', Z'', C'_1)$ with $C'_1 = [v_1^{min}, v_1^{max}] \times \{0\}$, we can replace the statement $(v_1, v_2) \in \mathcal{F}_v(P')$ in Eq. (13) with equivalent statements which follow from the CRN reduction theorem applied to P' with $j = 1$. The conclusion is that $(v_1, v_2, v_3, v_4)^T \in \mathcal{F}_v(P)$ if and only if

$$v_1 \in \mathcal{F}_v(P'') = [v_1^{min}, v_1^{max}], \tag{14}$$

$$v_2 = v_1 + z_{12} \left(\frac{\phi_2(|v_1|)}{v_1} \right)^*, \tag{15}$$

$$v_3 = v_2 + z_{23} \left(\frac{\phi_3(|v_2|)}{v_2} \right)^*, \tag{16}$$

$$v_4 = v_2 + z_{24} \left(\frac{\phi_4(|v_2|)}{v_2} \right)^*. \tag{17}$$

4.5 The tree expansion method

The system (14)–(17) concluding Example 4.4.1 illustrates how the outcomes of the tree reduction method, which are the reduction order, the voltage transfer functions $\{\phi_k\}_{k \geq 2}$ and the interval of possible values of v_1 , can be the basis of an algorithm for computing vectors in $\mathcal{F}_v(P)$. Indeed, to obtain a feasible vector we just need to choose $v_1 \in \mathcal{F}_v(P'') = [0.97, 1.0663]$ and then compute v_2, v_3, v_4 using the relations (15), (16) and (17). After finding the four voltages $\mathbf{v} = (v_1, v_2, v_3, v_4)^T$, the corresponding powers vector can be computed by the relation $\mathbf{s} = \mathcal{Q}_Z(\mathbf{v})$. The *tree expansion method* that computes a feasible pair $(\mathbf{v}, \mathbf{s}) \in \mathcal{F}_{vs}(P)$ given a value of $v_1 \in [v_1^{min}, v_1^{max}]$ is now described. Note that applying Eq. (10) is done in the order which is opposite to the reduction order.

4.6 The tree reduction/expansion method for solving the OPF problem

We can solve the OPF problem on a given restricted radial network by following the next steps: (i) represent the network as a CRN; (ii) employ the tree reduction method to obtain the reduction order, the voltage transfer functions and an interval of possible values of v_1 ; (iii) use the tree expansion method to compute a discretized version of $\mathcal{F}_{vs}(P)$ and (iv) pick the optimal solution out of the list of discretized pairs in $\mathcal{F}_{vs}(P)$. The algorithm executing these four steps is called the tree reduction/expansion method, and its details are given below.

Algorithm 3 The Tree Reduction/Expansion Method

Input: a restricted radial network P with n nodes; additional set of constraints $\mathcal{C} \subseteq \mathbb{C}^n \times \mathbb{C}^n$; the grid density parameter $m \in \mathbb{N}$, $m \geq 1$; objective function $f : \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{R}$.

Output: an optimal solution of (OPF) (up to discretization) on the network P with the additional constraints \mathcal{C} .

Steps:

1. Represent P as a CRN.
2. **Reduction:** Run the tree reduction method (Algorithm 1) on P and obtain the voltage transfer functions $\{\phi_k\}_{k=2}^n$, the reduction order $u_1, \dots, u_w = 1$ and the interval of possible values of $v_1, [v_1^{min}, v_1^{max}]$.
3. **Expansion:** Construct a discretized version of $\mathcal{F}_{vs}(P)$: for any $r = 1, 2, \dots, m$ employ the tree expansion method with input $(P, (u_1, \dots, u_w), \{\phi_k\}_{k \geq 2}, v_1^{min}(1 - t_r) + v_1^{max}t_r)$ where $t_r = \frac{r-1}{m-1}$ and obtain an output $(\mathbf{v}^r, \mathbf{s}^r) \in \mathcal{F}_{vs}(P)$.
4. Return an element of

$$\arg \min_{\mathbf{v}, \mathbf{s}} \{f(\mathbf{v}, \mathbf{s}) : (\mathbf{v}, \mathbf{s}) \in \mathcal{S} \cap \mathcal{C}\},$$

where $\mathcal{S} = \{(\mathbf{v}^r, \mathbf{s}^r) : r = 1, 2, \dots, m\}$.

4.7 Implementation of the tree reduction method

In contrast to the example detailed in Sects. 4.3.1 and 4.4.1, an actual implementation of the tree reduction method will not rely on symbolic expressions to represent the functions v_k, σ_k and ϕ_k . The reason for that is twofold: first, verifying that \tilde{v}_k are invertible based on symbolic expressions is a challenging task on its own; second, inverting \tilde{v}_k , an operation which is required in order to compute the voltage transfer function ϕ_k , requires analytically solving high-order polynomial equations, which is known to be generally impossible for any polynomial of degree 5 and above. Therefore, an actual implementation of the algorithm will involve replacing all the one-dimensional functions by appropriate approximations. This is

of course a modification of the algorithm, but fortunately, one-dimensional functions can be well approximated. Below we provide the exact details of how the one-dimensional functions were represented.

Representing $v_k, \tilde{v}_k, \sigma_k,$ and $\tilde{\sigma}_k$ The functions $v_k, \tilde{v}_k, \sigma_k,$ and $\tilde{\sigma}_k$ are represented by discretizing their argument $t \in [0, 1]$ at $d \geq 4$ ($d \in \mathbb{N}$) uniformly spaced points, where the discretization density d is a parameter of the algorithm. In other words, we define the vector

$$\mathbf{t} = \left(\frac{0}{d-1}, \frac{1}{d-1}, \frac{2}{d-1}, \dots, \frac{d-1}{d-1} \right)^T.$$

We represent the functions v_k, σ_k with the vectors $\mathbf{v}_k, \boldsymbol{\sigma}_k \in \mathbb{R}^d$ defined as

$$\mathbf{v}_k = v_k(\mathbf{t}), \boldsymbol{\sigma}_k = \sigma_k(\mathbf{t}), \tag{18}$$

where the functions are applied component-wise to the vector \mathbf{t} . Correspondingly, $\tilde{v}_k, \tilde{\sigma}_k$ with $j = \mathbf{Pa}_T(k)$ are represented by the vectors $\tilde{\mathbf{v}}_k, \tilde{\boldsymbol{\sigma}}_k \in \mathbb{R}^d$, constructed by applying Eq. (5) and (4), component-wise, to the vectors defined in Eq. (18):

$$\begin{aligned} \tilde{\mathbf{v}}_k &= \left| \mathbf{v}_k - z_{kj} \frac{\boldsymbol{\sigma}_k^*}{\mathbf{v}_k} \right| \\ \tilde{\boldsymbol{\sigma}}_k &= \boldsymbol{\sigma}_k - z_{kj} \frac{|\boldsymbol{\sigma}_k|^2}{\mathbf{v}_k^2} \end{aligned} \tag{19}$$

Invertibility and image of \tilde{v}_k A continuous function that maps real numbers to real numbers is invertible if and only if it is strictly monotone. We perform approximate strict monotonicity checking of \tilde{v}_k by verifying that the components of the vector $\tilde{\mathbf{v}}_k$ form a strictly monotone sequence. We assume that for a large enough d , strict monotonicity of the sequence is a good indicator for the strict monotonicity of the function. Approximating **image**(\tilde{v}_k) is also based on the discretization above:

$$\mathbf{image}(\tilde{v}_k) \approx [\min(\tilde{\mathbf{v}}_k), \max(\tilde{\mathbf{v}}_k)]. \tag{20}$$

Representing ϕ_k Since we need to be able to compute $\phi_k(\cdot)$ at arbitrary points, we approximate these functions with cubic splines. We can verify, by induction on the structure of the tree T , that the functions $\{\phi_k\}_{k=2}^n$ are all smooth, and thus can be well-approximated by cubic splines [7, Ch. XII]. More specifically, since $\phi_k = \tilde{\sigma}_k \circ \tilde{v}_k^{-1}$, it satisfies

$$\phi_k((\tilde{\mathbf{v}}_k)_r) = (\tilde{\boldsymbol{\sigma}}_k)_r, \quad r = 1, \dots, d. \tag{21}$$

Therefore, for any $k = 2, \dots, n$ the function ϕ_k is represented using an interpolating cubic spline, whose domain is the interval in Eq. (20) and approximates the equations in (21).

The spline’s knots are the interpolation points, and end conditions are specified by the well-known ‘not a knot’ method [7, Ch. IV, pp. 43–48]. These parameters were chosen since they are widely used with spline interpolation, and have readily available implementations, such as the `csapi` MATLAB function [18].

4.8 Handling non-leaf PV-constrained nodes

To demonstrate the concept of our algorithm, we made a simplifying assumption which requires PV constrained nodes to be among the leaves of the rooted tree which defines the network. However, the tree-reduction theorem, and the resulting algorithms, can be easily extended for internal PV nodes.

It can be easily shown that in Theorem 1, replacing the assumption that C_j is a PQ constrained node with an assumption that it is a PV constrained node, defined by

$$\hat{v}_j(t) = V_j, \quad \hat{\sigma}_j(t) = \hat{p}_j + \left((1 - t)Q_j^{min} + tQ_j^{max} \right) i,$$

results in a theorem that differs from the original in the following manner:

- Equation (6) is replaced by

$$U_j = \{V_j\} \cap \left(\bigcap_{k=j+1}^n \mathbf{image}(\tilde{v}_k) \right)$$

- Equation (8) is replaced by

$$v_j(t) = V_j$$

- Equation (9) is replaced by

$$\sigma_j(t) = \hat{\sigma}_j(t) + \sum_{j=k+1}^n \phi_k(V_j)$$

In other words, after a reduction via a PV constrained node, we obtain a PV constrained leaf. The Tree Reduction / Expansion method is modified accordingly.

4.9 Eliminating the invertibility assumption

Eliminating the invertibility assumption requires us first to analyze the tree reduction/expansion method from a high level perspective. Consider a CRN P_{input} which is given as an input to the tree reduction method. Let Φ be a mapping from the nodes of P_{input} to their associated voltage transfer function, e.g. $\phi_k \equiv \Phi(k)$. The tree reduction method begins with an empty Φ , and each subsequent reduction results in a smaller CRN, while the voltage transfer functions associated with the discarded nodes are added to Φ .

Assume that at some stage during the execution of the tree reduction method on P_{input} we have the CRN $P = (T, Z, C_1, \dots, C_n)$ and the voltage transfer functions Φ . The invariant maintained by the tree reduction method, which is ensured by Theorem 1, is that at each stage the pair (P, Φ) is a full characterization of $\mathcal{F}_v(P_{input})$. Formally, we write $\mathcal{F}_v(P, \Phi) = \mathcal{F}_v(P_{input})$, where $\mathcal{F}_v(P, \Phi)$ is the set of all voltage vectors obtainable by taking any $\mathbf{v}' \in \mathcal{F}_v(P)$ and applying the tree expansion method using the voltage transfer functions Φ .

Now assume that the next reduction will be performed via node j , and for some $k \in \mathbf{Ch}_T(j)$ the function \tilde{v}_k is not invertible. Clearly, the tree reduction theorem cannot be applied to P and the method will fail. However, assuming that \tilde{v}_k has a finite number of local extremum points, we can overcome this limitation. Let $0 = z_0 < z_1 < \dots < z_\ell = 1$ be the extremum points of \tilde{v}_k . We can split the domain $\mathbf{dom}(\tilde{v}_k) = [0, 1]$ into the sub-domains

$$[z_0, z_1], [z_1, z_2], \dots, [z_{\ell-2}, z_{\ell-1}], [z_{\ell-1}, z_\ell].$$

On each of these domains \tilde{v}_k is clearly strictly monotone and therefore invertible. This observation allows us to split the CRN P into the CRNs P_1, \dots, P_ℓ , in which the invertibility assumption, at least for node k , holds. For every $r \in \{1, \dots, \ell\}$ we define the ‘reparameterized’ versions of v_k, σ_k ,

$$v_{k,r} : [0, 1] \rightarrow \mathbb{R}, \quad v_{k,r}(t) \equiv v_k((1 - t)z_{r-1} + tz_r),$$

$$\sigma_{k,r} : [0, 1] \rightarrow \mathbb{C}, \sigma_{k,r}(t) \equiv \sigma_k((1 - t)z_{r-1} + tz_r),$$

$$\mathcal{C}_{k,r} \equiv \mathbf{image}(v_{k,r}, \sigma_{k,r}),$$

and the CRN

$$P_r = (T, Z, \mathcal{C}_1, \dots, \mathcal{C}_{k-1}, \mathcal{C}_{k,r}, \mathcal{C}_{k+1}, \dots, \mathcal{C}_n).$$

By construction we have $\mathcal{C}_k = \bigcup_{r=1}^{\ell} \mathcal{C}_{k,r}$, and therefore $\mathcal{F}_v(P, \Phi) = \bigcup_{r=1}^{\ell} \mathcal{F}_v(P_r, \Phi)$. Thus, the set of pairs $\{(P_r, \Phi)\}_{r=1}^{\ell}$ remains a full characterization of $\mathcal{F}_v(P_{\text{input}})$. The splitting process can continue with respect to each of the child nodes of j , until all CRNs satisfy the assumptions of the tree reduction theorem. Then, the tree reduction method can continue by performing the reduction via node j on all the pairs $\{(P_r, \Phi)\}$ in parallel.

Equipped with this idea, we can modify the tree reduction method to operate on a set of pairs instead of a single pair. Pairs are added to the set when a CRN is split as described above. A pair (P', Φ') is removed from the set when the algorithm concludes that $\mathcal{F}_v(P') = \emptyset$.

We call a network for which the above modification of the tree reduction method encounters only functions \tilde{v}_k with a finite number of extremal points a **well behaved** network. In our experiments we did not encounter networks which are not well behaved.

4.9.1 Implementation

The modification of the tree reduction method mentioned above raises several practical issues: detection of the extremal points, efficient representation of the set of CRNs at each stage, and the memory and time complexity it induces.

Extremal points detection In an actual implementation, the functions v_k, σ_k are represented by their discrete approximations. For simplicity, our approach assumes that the extremal points of \tilde{v}_k are the extremal elements in the array \tilde{v}_k which is defined in (18) and approximates \tilde{v}_k .

Set of pairs representation Note that when a pair (P, Φ) is split, the resulting pairs share a lot of data. In fact, they share all data, except for the constraint sets (which can also be represented by curves) that were split. Instead of duplicating entire networks, we associate each node with a *set* of constraint sets or voltage transfer functions, depending on whether or not the node was already discarded by a reduction. Each network is identified by a specific choice of a constraint set \mathcal{C}_k for every leaf k out of the set of constraint sets associated with node k .

Initially, all sets are singletons. When a constraint curve \mathcal{C}_k needs to be split, it is removed from the set associated with node k and replaced with the set of constraint sets which result from the above-mentioned splitting process.

Performing a reduction via node j on all networks in parallel is done by considering all possible choices of constraint sets associated with the children of node j , since each choice identifies a CRN. Each of the constraint sets associated with the leaf k is transformed into a voltage transfer function, and then all choices of voltage transfer functions of all the child nodes of j are used to compute the possible constraint sets associated with node j . To identify which network a constraint or a voltage transfer function belongs to, we store a ‘pointer’ to the specific child voltage transfer functions that were used to compute it. This process is described in Fig. 7.

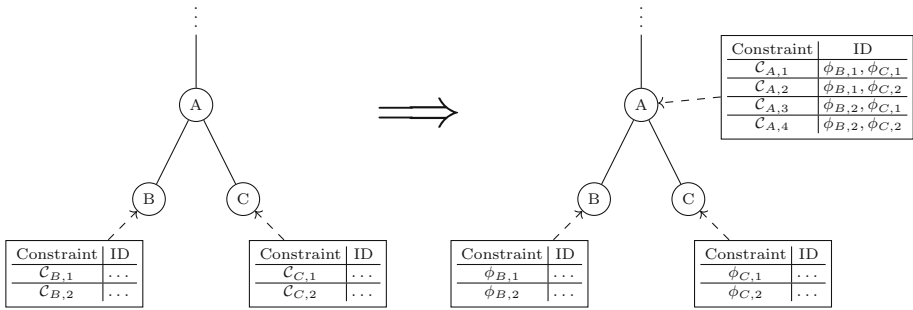


Fig. 7 A reduction via node A on all network in parallel. *On the left* two constraints associated with node B and two with node C , representing four networks. *On the right* the result of the reduction. Four networks were reduced into four smaller networks. Node A is associated with four possible constraint curves, each one has an associated network ID in the form of the specific choice of child voltage transfer functions

5 Proof of the tree reduction theorem

First, we prove the following technical Lemma:

Lemma 1 *Let $T = (\mathcal{V}, \mathcal{E})$ be the topology tree of some power network with $\mathcal{V} = \{1, \dots, n\}$, and let $Z : \mathcal{E} \rightarrow \mathbb{C}$ be the associated impedance function. Let \mathbf{Y} be defined as in (3). Suppose that $B \in \mathcal{V}$ is a leaf, and $\{A, B\} \in \mathcal{E}$ be the edge connected to B . Then $(\mathbf{v}, \mathbf{s}) \in \text{PFE}(T, Z)$ if and only if there exists $\tilde{s}_B \in \mathbb{C}$ such that $(\mathbf{v}, \mathbf{s}, \tilde{s}_B)$ satisfy the following equations:*

$$s_i = \sum_{\substack{j=1 \\ j \neq B}}^n v_i v_j^* Y_{ij}^* \quad i = 1 \dots n, i \neq A, i \neq B, \tag{22}$$

$$s_A + \tilde{s}_B = \sum_{\substack{i=1 \\ i \notin \{A, B\}}}^n v_A v_i^* Y_{A,i}^* + |v_A|^2 \left(\sum_{\substack{i \in N_A \\ i \neq B}} \frac{1}{z_{A,i}^*} \right), \tag{23}$$

$$|v_A| = \left| |v_B| - \frac{s_B^* z_{A,B}}{|v_B|} \right|, \tag{24}$$

$$\tilde{s}_B = s_B - z_{A,B} \frac{|s_B|^2}{|v_B|^2}, \tag{25}$$

$$v_B = v_A + \frac{\tilde{s}_B^* z_{A,B}}{v_A^*}. \tag{26}$$

Proof Since B is a leaf with a connecting edge $\{A, B\}$, we have $N_B = \{A\}$. Thus, according to the definition of \mathbf{Y} in (3) we have

$$\forall i \notin \{A, B\} : Y_{i,B} = Y_{B,i} = 0.$$

Hence, for $i \notin \{A, B\}$, equations (2) must hold if and only if (22) holds. We now need to prove that (2) holds for $i = A, B$ if and only if (23), (24), (25), and (26) hold. The proof is separated into the *only if* and *if* parts.

The only if part. Assume that the power flow equations (2) hold for $i = A, B$. Using the power flow equation for s_A , we can write:

$$\begin{aligned}
 s_A &= \sum_{i=1}^n v_A v_i^* Y_{A,i}^* \\
 &= \sum_{\substack{i=1 \\ i \notin \{A,B\}}}^n v_A v_i^* Y_{A,i}^* + |v_A|^2 Y_{A,A}^* + v_A v_B^* Y_{A,B}^* \\
 &= \sum_{\substack{i=1 \\ i \notin \{A,B\}}}^n v_A v_i^* Y_{A,i}^* + |v_A|^2 \left(\sum_{i \in N_A} \frac{1}{z_{A,i}^*} \right) - \frac{v_A v_B^*}{z_{A,B}^*} \tag{27} \\
 &= \sum_{\substack{i=1 \\ i \notin \{A,B\}}}^n v_A v_i^* Y_{A,i}^* + |v_A|^2 \left(\sum_{\substack{i \in N_A \\ i \neq B}} \frac{1}{z_{A,i}^*} \right) - \underbrace{\frac{v_A v_B^* - |v_A|^2}{z_{A,B}^*}}_{\tilde{s}_B}
 \end{aligned}$$

By defining

$$\tilde{s}_B = \frac{v_A v_B^* - |v_A|^2}{z_{A,B}^*}, \tag{28}$$

and re-arranging (27), we obtain (23).

According to the power flow equation for s_B , using the relations $Y_{A,B} = -\frac{1}{z_{A,B}}$ and $Y_{B,B} = \frac{1}{z_{A,B}}$ as given in (3), we have:

$$s_B = -\frac{v_B v_A^*}{z_{A,B}^*} + \frac{v_B v_B^*}{z_{A,B}^*},$$

that is,

$$s_B z_{A,B}^* = -v_B v_A^* + |v_B|^2.$$

Applying the conjugate on both sides and extracting v_A we get:

$$v_A = \frac{|v_B|^2 - s_B^* z_{A,B}}{v_B^*}. \tag{29}$$

By taking the absolute value on both sides of (29), we obtain:

$$|v_A| = \frac{||v_B|^2 - s_B^* z_{A,B}|}{|v_B|} = \left| |v_B| - \frac{s_B^* z_{A,B}}{|v_B|} \right|$$

which is exactly (24).

Substituting the expression for v_A from (29) and for $|v_A|$ from (24) into the definition of \tilde{s}_B in (28) yields:

$$\begin{aligned}
 \tilde{s}_B &= \frac{v_A v_B^* - |v_A|^2}{z_{A,B}^*} \\
 &= \frac{|v_B|^2 - s_B^* z_{A,B} - \left| |v_B| - \frac{s_B^* z_{A,B}}{|v_B|} \right|^2}{z_{A,B}^*}
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{|v_B|^2 - s_B^* z_{A,B} - \left(|v_B|^2 - s_B^* z_{A,B} - s_B z_{A,B}^* + \frac{|s_B|^2 |z_{A,B}|^2}{|v_B|^2} \right)}{z_{A,B}^*} \\
 &= \frac{s_B z_{A,B}^* - \frac{|s_B|^2 |z_{A,B}|^2}{|v_B|^2}}{z_{A,B}^*} \\
 &= s_B - z_{A,B} \frac{|s_B|^2}{|v_B|^2}, \tag{30}
 \end{aligned}$$

which proves (25),

Finally, by re-arranging Eq. (28) we obtain:

$$v_B = \frac{\tilde{s}_B^* z_{A,B} + |v_A|^2}{v_A^*} = v_A + \frac{\tilde{s}_B^* z_{A,B}}{v_A^*},$$

which is exactly (26).

The if part. Assume that (23), (24), (25), and (26) hold. We will show that (2) holds for $i = A, B$.

Beginning from (25) and following the mathematical derivation in (30) backwards, utilizing (24), we can conclude that:

$$\tilde{s}_B = \frac{|v_B|^2 - s_B^* z_{A,B} - |v_A|^2}{z_{A,B}^*}. \tag{31}$$

Extracting \tilde{s}_B from (26) we obtain:

$$\tilde{s}_B = \frac{v_B^* v_A - |v_A|^2}{z_{A,B}^*}. \tag{32}$$

Subtracting both equations, and extracting s_B from the result we obtain:

$$s_B = \frac{|v_B|^2}{z_{A,B}^*} - \frac{v_B v_A^*}{z_{A,B}^*},$$

which is exactly the power flow equation for node B .

Substituting (32) into (23), and following the mathematical derivation in (27) backwards, we obtain the power flow equation for node A :

$$s_A = \sum_{i=1}^n v_A v_i^* Y_{A,i}^*.$$

□

Note the following remarks:

1. Equations (22) and (23) are exactly the power flow equations associated with the network (\bar{T}, \bar{Z}) , where \bar{T} is the tree T with node B and the edge (A, B) removed and \bar{Z} is the impedance function restricted to \bar{T} .
2. Thus, by assuming w.l.o.g. that $B = n$ and $A = n - 1$, equations (22) and (23) can be replaced by $(\mathbf{v}', \mathbf{s}') \in PFE(\bar{T}, \bar{Z})$, where $\mathbf{v}' = (v_1, \dots, v_{n-1})$ and $\mathbf{s}' = (s_1, \dots, s_{n-2}, s_{n-1} + \tilde{s}_n)$.

The remarks above help us prove the next technical Lemma, which is directly used in the proof of the tree reduction theorem.

Lemma 2 Let $T = (\mathcal{V}, \mathcal{E})$ be the topology tree of some power network with $\mathcal{V} = \{1, \dots, n\}$, and let $Z : \mathcal{E} \rightarrow \mathbb{C}$ be the associated impedance function. Let T' be the reduction of T via the reducible node $j \in \mathcal{V}$, and assume w.l.o.g. that $\text{Ch}_T(j) = \{j + 1, \dots, n\}$. Then $(\mathbf{v}, \mathbf{s}) \in \text{PFE}(T, Z)$ if and only if all of the following hold (Z' is the impedance function restricted to the edges of T'):

$$(\mathbf{v}', \mathbf{s}') \in \text{PFE}(T', Z'), \quad \begin{aligned} \mathbf{v}' &= (v_1, \dots, v_j) \\ \mathbf{s}' &= (s_1, \dots, s_{j-1}, s'_j) \end{aligned} \tag{33}$$

$$|v_j| = \left| |v_k| - \frac{s_k^* z_{kj}}{|v_k|} \right|, \quad k = j + 1, \dots, n \tag{34}$$

$$s'_j = s_j + \sum_{k=j+1}^n \tilde{s}_k, \tag{35}$$

$$\tilde{s}_k = s_k - z_{kj} \frac{|s_k|^2}{|v_k|^2}, \quad k = j + 1, \dots, n \tag{36}$$

$$v_k = v_j + \frac{\tilde{s}_k^* z_{kj}}{v_j^*}, \quad k = j + 1, \dots, n \tag{37}$$

Proof Follows directly by repeatedly applying Lemma 1 for $B = j + 1, \dots, n$ with $A = j$. □

We are now ready to prove the tree reduction theorem (Theorem 1).

Proof of Theorem 1 The statement $\mathbf{v} \in \mathcal{F}_v(P)$, by definition, holds if and only if

$$\begin{aligned} (\mathbf{v}, \mathbf{s}) &\in \text{PFE}(T, Z), \\ (|v_k|, s_k) &\in \mathcal{C}_k, & k = 1, \dots, n, \\ \arg(v_1) &= 0. \end{aligned}$$

Using the definition of \mathcal{C}_k for $k = j, \dots, n$, the system above is equivalent to the following system in the variables $v_1, \dots, v_n, s_1, \dots, s_n, t_{j+1}, \dots, t_n$:

$$\begin{aligned} (\mathbf{v}, \mathbf{s}) &\in \text{PFE}(T, Z), \\ |v_k| &= v_k(t_k), \quad s_k = \sigma_k(t_k), \quad t_k \in [0, 1], & k = j + 1, \dots, n, \\ s_j &= \hat{s}_j, & \text{if } j \neq 1, \\ V_j^{\min} &\leq |v_j| \leq V_j^{\max}, & \text{if } j \neq 1, \\ (|v_k|, s_k) &\in \mathcal{C}_k, & k = 1, \dots, j - 1, \\ \arg(v_1) &= 0. \end{aligned}$$

Applying Lemma 2 to the constraint $(\mathbf{v}, \mathbf{s}) \in \text{PFE}(T, Z)$, and utilizing the identities $|v_k| = v_k(t_k)$, $s_k = \sigma_k(t_k)$, $s_j = \hat{s}_j$, we obtain equivalence to the following system in the variables $v_1, \dots, v_j, s_1, \dots, s_{j-1}, s'_j, t_{j+1}, \dots, t_n, \tilde{s}_{j+1}, \dots, \tilde{s}_n$:

$$\begin{aligned} (\mathbf{v}', \mathbf{s}') &\in \text{PFE}(T', Z'), & \mathbf{v}' &= (v_1, \dots, v_j)^T, \quad \mathbf{s}' = (s_1, \dots, s_{j-1}, s'_j)^T, \\ |v_j| &= \left| v_k(t_k) - \frac{(\sigma_k(t_k))^* z_{kj}}{v_k(t_k)} \right|, & k &= j + 1, \dots, n, \\ s'_j &= \hat{s}_j + \sum_{k=j+1}^n \tilde{s}_k, & & \text{if } j \neq 1, \end{aligned}$$

$$\begin{aligned}
 v_k &= v_j + \frac{\tilde{s}_k^* z_{kj}}{v_j^*}, & k &= j + 1, \dots, n, \\
 \tilde{s}_k &= \sigma_k(t_k) - z_{kj} \frac{|\sigma_k(t_k)|^2}{(v_k(t_k))^2}, & k &= j + 1, \dots, n, \\
 t_k &\in [0, 1], & k &= j + 1, \dots, n, \\
 V_j^{min} &\leq |v_j| \leq V_j^{max}, & & \text{if } j \neq 1, \\
 (|v_k|, s_k) &\in \mathcal{C}_k, & k &= 1, \dots, j - 1, \\
 \arg(v_1) &= 0.
 \end{aligned}$$

Substituting the definition of $\tilde{v}_k, \tilde{\sigma}_k$, and observing that $\tilde{s}_k = \tilde{\sigma}_k(t_k)$, we conclude that the above system is equivalent to the following system in the variables $v_1, \dots, v_j, s_1, \dots, s_{j-1}, s'_j, t_{j+1}, \dots, t_n$:

$$\begin{aligned}
 (\mathbf{v}', \mathbf{s}') &\in \text{PFE}(T', Z'), & \mathbf{v}' &= (v_1, \dots, v_j)^T, \mathbf{s}' = (s_1, \dots, s_{j-1}, s'_j)^T, \\
 |v_j| &= \tilde{v}_k(t_k), & k &= j + 1, \dots, n, \\
 s'_j &= \hat{s}_j + \sum_{k=j+1}^n \tilde{\sigma}_k(t_k), & & \text{if } j \neq 1, \\
 v_k &= v_j + \frac{(\tilde{\sigma}_k(t_k))^* z_{kj}}{v_j^*}, & k &= j + 1, \dots, n, \\
 t_k &\in [0, 1], & k &= j + 1, \dots, n, \\
 V_j^{min} &\leq |v_j| \leq V_j^{max}, & & \text{if } j \neq 1, \\
 (|v_k|, s_k) &\in \mathcal{C}_k, & k &= 1, \dots, j - 1, \\
 \arg(v_1) &= 0.
 \end{aligned}$$

Using the assumed invertibility of \tilde{v}_k , we can conclude that $|v_j| = \tilde{v}_k(t_k)$ if and only if $|v_j| \in \text{image}(\tilde{v}_k)$ and $t_k = \tilde{v}_k^{-1}(|v_j|)$, and obtain equivalence to the system in the variables $v_1, \dots, v_j, s_1, \dots, s_{j-1}, s'_j$,

$$\begin{aligned}
 (\mathbf{v}', \mathbf{s}') &\in \text{PFE}(T', Z'), & \mathbf{v}' &= (v_1, \dots, v_j)^T, \mathbf{s}' = (s_1, \dots, s_{j-1}, s'_j)^T, \\
 s'_j &= \hat{s}_j + \sum_{k=j+1}^n \tilde{\sigma}_k(\tilde{v}_k^{-1}(|v_j|)), & & \text{if } j \neq 1, \\
 v_k &= v_j + \frac{(\tilde{\sigma}_k(\tilde{v}_k^{-1}(|v_j|)))^* z_{kj}}{v_j^*}, & k &= j + 1, \dots, n, \\
 |v_j| &\in \text{image}(\tilde{v}_k), & k &= j + 1, \dots, n, \quad (*) \\
 V_{min} &\leq |v_j| \leq V_{max}, & & \text{if } j \neq 1, \quad (**) \\
 (|v_k|, s_k) &\in \mathcal{C}_k, & k &= 1, \dots, j - 1, \\
 \arg(v_1) &= 0.
 \end{aligned}$$

Since \tilde{v}_k is continuous on the domain $[0, 1]$, the set $\text{image}(\tilde{v}_k)$ is a bounded closed interval. Recalling the definition of U_j , we conclude that if $U_j = \emptyset$, then $\mathcal{F}_v(P) = \emptyset$. If $U_j \neq \emptyset$, then it is a closed bounded interval. Denoting $U_j = [v_j^{min}, v_j^{max}]$, the constraints (*) and

(**) above can be replaced with the inequality $v_j^{min} \leq |v_j| \leq v_j^{max}$. Thus, by letting

$$v_j : [0, 1] \rightarrow \mathbb{R} \quad \text{s.t.} \quad v_j(t) = v_j^{max} \cdot t + v_j^{min} \cdot (1 - t),$$

the system can be equivalently written as

$$\begin{aligned} (\mathbf{v}', \mathbf{s}') &\in \text{PFE}(T', Z'), & \mathbf{v}' &= (v_1, \dots, v_j)^T, \quad \mathbf{s}' = (s_1, \dots, s_{j-1}, s'_j)^T, \\ s'_j &= \hat{s}_j + \sum_{k=j+1}^n \tilde{\sigma}_k(\tilde{v}_k^{-1}(v_j(t))), & & \text{if } j \neq 1, \\ v_k &= v_j + \frac{(\tilde{\sigma}_k(\tilde{v}_k^{-1}(|v_j|)))^* z_{kj}}{v_j^*}, & & k = j + 1, \dots, n, \\ |v_j| &= v_j(t), \\ (|v_k|, s_k) &\in \mathcal{C}_k, & & k = 1, \dots, j - 1 \\ t &\in [0, 1], \\ \arg(v_1) &= 0. \end{aligned}$$

Note that when $j = 1$, the constraint $(\mathbf{v}', \mathbf{s}') \in \text{PFE}(T', Z')$ is equivalent to $s'_1 = 0$. Thus, by defining σ_j as in (9), letting

$$\mathcal{C}'_j = \text{image}(v_j, \sigma_j),$$

reordering, and recalling the definition of ϕ_k , the system can be rewritten as

$$\begin{aligned} (\mathbf{v}', \mathbf{s}') &\in \text{PFE}(T', Z'), & \mathbf{v}' &= (v_1, \dots, v_j)^T, \quad \mathbf{s}' = (s_1, \dots, s_{j-1}, s'_j)^T, \\ (|v_k|, s_k) &\in \mathcal{C}_k, & & k = 1, \dots, j - 1, \\ (|v_j|, s'_j) &\in \mathcal{C}'_j, \\ \arg(v_1) &= 0, \\ v_k &= v_j + z_{kj} \left(\frac{\phi_k(|v_j|)}{v_j} \right)^*, & & k = j + 1, \dots, n \end{aligned}$$

The system above completes the proof, since the first four relations are the same as $\mathbf{v}' \in \mathcal{F}_v(P')$ and the fourth relation is the same as item 2 in the premise of the theorem. □

6 Numerical experiments

We evaluated our implementation of the tree reduction/expansion method for solving the OPF problem on several cases to demonstrate the following features of our method:

- **Accuracy:** show that a moderate value of the approximation density d is enough for the pairs (\mathbf{v}, \mathbf{s}) produced by our method to be feasible.
- **Reliability:** we always find a global optimum for feasible problems, in contrast to MAT-POWER [20], which to the best of our knowledge is considered state of the art.
- **The number of parallel CRNs in practice:** verify that the number of CRNs on which the tree reduction method operates concurrently remains small.

We performed our numerical experiments on existing networks that were slightly modified to satisfy all the assumptions required from restricted radial networks. Specifically, we used

the IEEE radial distribution networks with 13, 37, and 123 buses available at [19], a network with 47 buses used in [9], and a network with 69 buses used in [4]. We made our MATLAB implementation of the tree reduction/expansion method as an open source project on GitHub¹, together with the networks we used for our experiments.

6.1 Accuracy

In order to say that a pair (\mathbf{v}, \mathbf{s}) produced by our method is “practically feasible” for a restricted radial network P , we need to define some measures of deviation from feasibility. The definition of these measures uses the following ingredients: the set of PQ-constrained nodes PQ , the set of PV-constrained nodes PV , the distance function

$$d(x; \alpha, \beta) = \begin{cases} x - \beta & x > \beta \\ \alpha - x & x < \alpha \\ 0 & \text{otherwise,} \end{cases}$$

which measures the distance of x from the interval $[\alpha, \beta]$. With the ingredients above, we define the following measures for deviation from the constraints of PQ constrained nodes:

$$E_{PQ,v}(\mathbf{v}, \mathbf{s}) = \max_{j \in PQ} d(|v_j|, V_j^{min}, V_j^{max}),$$

$$E_{PQ,s}(\mathbf{v}, \mathbf{s}) = \max_{j \in PQ} |s_j - \hat{s}_j|,$$

and the following measures for PV constrained nodes:

$$E_{PV,v}(\mathbf{v}, \mathbf{s}) = \max_{j \in PV} |\hat{v}_j - |v_j||$$

$$E_{PV,p}(\mathbf{v}, \mathbf{s}) = \max_{j \in PV} |\hat{p}_j - \text{re}(s_j)|$$

$$E_{PV,q}(\mathbf{v}, \mathbf{s}) = \max_{j \in PV} d(\text{im}(s_j), Q_j^{min}, Q_j^{max})$$

Clearly, for a pair $(\mathbf{v}, \mathbf{s}) \in \mathcal{F}_{vs}(P)$ all of the above measures are zero. Note that we do not measure the deviation from the constraints imposed by the power flow equations, since, by definition, our method computes $\mathbf{s} = \mathcal{Q}_Z(\mathbf{v})$.

For every network, we chose a set of approximation densities $d \in [2^3, 2^{12}]$ as inputs to the tree reduction method (Algorithm 1), and for each such density produced $m = 1000$ samples of the feasible set using the tree-expansion method (Algorithm 2), and took the maximum of each error measure over all the samples. In all our experiments, the measures $E_{PQ,v}$ and $E_{PV,q}$ were always zero. The results for the other measures, which appear in Fig. 8, show that for $d \geq 2^{10}$ we obtain solutions which are practically feasible. In addition, very accurate results are obtained even for d as small as 2^5 .

6.2 Reliability

We compared our method to MATPOWER on networks which were randomly generated from our existing networks by perturbing every PQ constraint $C_j = [V_j^{min}, V_j^{max}] \times \{\hat{p}_j + i\hat{q}_j\}$ into

$$[V_j^{min}, V_j^{max}] \times \{\hat{p}_j \cdot \alpha_j + (\hat{q}_j \cdot \beta_j)i\},$$

¹ https://github.com/alexshft/trem_opf_solver.

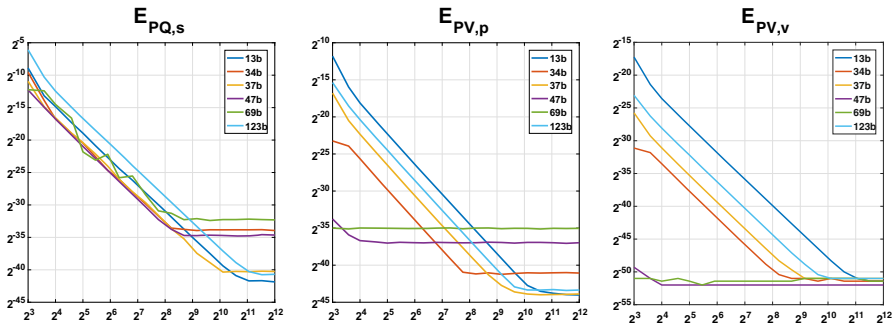


Fig. 8 Constraint violation measures for different networks

Table 1 The % of random networks, generated from each original network, for which our method found a solution while MATPOWER did not

13-Bus	34-Bus	37-Bus	47-Bus	69-Bus	123-Bus
6.46%	5.22%	2.56%	2.78%	2.34%	39.64%

where α_j, β_j are uniformly distributed random variables in $[0, 2]$. That is, we introduced uniform multiplicative noise to the prescribed power in the PQ constraints in our networks, while leaving the other constraints unchanged.

From each network, we generated 5000 random networks and solved the OPF problem on each random network using both MATPOWER and the tree reduction/expansion method, with the following *stability* objective function:

$$f(\mathbf{v}, \mathbf{s}) = \sum_{j \in PQ} \left| |v_j| - \frac{1}{2}(V_j^{min} + V_j^{max}) \right|$$

Note that MATPOWER does not naturally support the function above, but its extension mechanism lets us specify this function quite easily. For each network we gathered statistics about the results of running both solvers on the randomly perturbed networks.

For each original network, we measured the percentage of randomly generated networks for which MATPOWER produced a feasible solution while our algorithm claimed that the network is infeasible. However, there were many networks for which MATPOWER could not find a solution since it did not converge, while our method did. Table 1 shows the percentage, out of the 5000 random networks we generated from each original networks, for which MATPOWER could not find a solution. As an example, a specific 13-bus network is shown in Fig. 9.

Finally, comparing the objective function value for the randomly generated networks for which both algorithms reported a solution resulted in no significant difference. This leads us to the conclusion that for this set of experiments, if MATPOWER finds a solution, it is probably a global solution.

The results reported above provide empirical evidence for our claim that in contrast to MATPOWER, our method finds the global solution, up to discretization, if one exists.

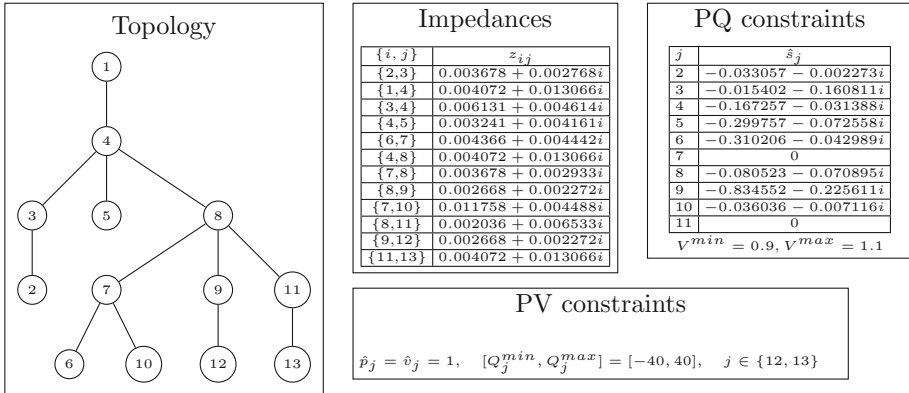


Fig. 9 A 13-bus network for which MATPOWER does not converge given the OPF problem with the stability objective function

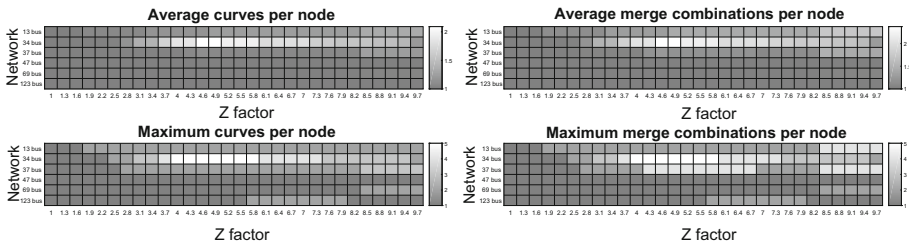


Fig. 10 The number of curves and curve combination considered when the impedances are increased by various factors

6.3 CRN set size and computational time

One way to violate the invertibility assumption is by increasing the network’s impedances. Intuitively, looking at equation (4), when the impedance z_{kj} is small, the value of v_k dominates the expression. Since in our algorithm v_k are linear functions, the resulting \tilde{v}_k is ‘almost linear’, and thus invertible.

Based on the intuition above, we multiplied all the impedances in our test networks by various factors between 1 and 10. For each execution, we measured the number of curves associated with each node, and the number of curve combinations considered when a reduction is performed. The results are summarized in Fig. 10. Indeed, the representation size and the computational time we pay for eliminating the invertibility assumption remains small.

Note that when the impedances increase too much, the network becomes infeasible, and the tree reduction method stops long before reaching the root of the tree. Thus, when the factor is too large, the number of curves and considered combinations drops.

References

1. Bienstock, D.: Progress on solving power flow problems. *Optima* **93**, 1–7 (2003)
2. Bose, S., Gayme, D.F., Low, S., Chandy, K.M.: Optimal power flow over tree networks. In: 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1342–1348 (2011)

3. Cespedes, R.G.: New method for the analysis of distribution networks. *IEEE Trans. Power Deliv.* **5**(1), 391–396 (1990)
4. Chakravorty, M., Das, D.: Voltage stability analysis of radial distribution networks. *Int. J. Electr. Power Energy Syst.* **23**(2), 129–135 (2001)
5. Chang, G.W., Chu, S.Y., Wang, H.L.: An improved backward/forward sweep load flow algorithm for radial distribution systems. *IEEE Trans. Power Syst.* **22**(2), 882–884 (2007)
6. Das, D., Kothari, D.P., Kalam, A.: Simple and efficient method for load flow solution of radial distribution networks. *Int. J. Electr. Power Energy Syst.* **17**(5), 335–346 (1995)
7. de Boor, Carl: A Practical Guide to Splines, volume 27 of Applied Mathematical Sciences. Springer-Verlag, New York (2001)
8. Eminoglu, U., Hocaoglu, M.H.: Distribution systems forward/backward sweep-based power flow algorithms: a review and comparison study. *Electr. Power Compon. Syst.* **37**(1), 91–110 (2008)
9. Farivar, M., Clarke, C.R., Low, S.H., Chandy, K.M.: Inverter var control for distribution systems with renewables. In: 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 457–462. (2011)
10. Farivar, M., Low, S.H.: Branch flow model: relaxations and convexification—part i. *IEEE Trans. Power Syst.* **28**(3), 2554–2564 (2013)
11. Frank, Stephen, Steponavice, Ingrida, Rebennack, Steffen: Optimal power flow: a bibliographic survey i. *Energy Syst.* **3**(3), 221–258 (2012)
12. Gan, L., Li, N., Topcu, U., Low, S.H.: Exact convex relaxation of optimal power flow in radial networks. *IEEE Trans. Autom. Control* **60**(1), 72–87 (2015)
13. Haque, M.H.: Efficient load flow method for distribution systems with radial or mesh configuration. *IEE Proc. Gen. Transm. Distrib.* **143**(1), 33–38 (1996)
14. Kocuk, B., Dey, S.S., Sun, X.A.: Inexactness of sdp relaxation and valid inequalities for optimal power flow. *IEEE Trans. Power Syst.* **31**(1), 642–651 (2016)
15. Lavaei, J., Low, S.H.: Zero duality gap in optimal power flow problem. *IEEE Trans. Power Syst.* **27**(1), 92–107 (2012)
16. Low, S.H.: Convex relaxation of optimal power flow—part i: formulations and equivalence. *IEEE Trans. Control Netw. Syst.* **1**(1), 15–27 (2014)
17. Low, S.H.: Convex relaxation of optimal power flow—part ii: exactness. *IEEE Trans. Control Netw. Syst.* **1**(2), 177–189 (2014)
18. MathWorks. Cubic spline interpolation—matlab csapi. <http://www.mathworks.com/help/curvefit/csapi.html>. Accessed 10 April 2017
19. IEEE Power and Energy Society. Distribution test feeders. <https://ewh.ieee.org/soc/pes/dsacom/testfeeders/>, 2000. Accessed 10 April 2017
20. Zimmerman, R.D., Murillo-Sanchez, C.E., Thomas, R.J.: Matpower: steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Trans. Power Syst.* **26**(1), 12–19 (2011)