Matrix Minor Reformulation and SOCP-based Spatial Branch-and-Cut Method for the AC Optimal Power Flow Problem

Burak Kocuk, Santanu S. Dey, X. Andy Sun

March 8, 2017

Abstract

Alternating current optimal power flow (AC OPF) is one of the most fundamental optimization problems in electrical power systems. It can be formulated as a semidefinite program (SDP) with rank constraints. Solving AC OPF, that is, obtaining near optimal primal solutions as well as high quality dual bounds for this non-convex program, presents a major computational challenge to today's power industry for the real-time operation of large-scale power grids. In this paper, we propose a new technique for reformulation of the rank constraints using both principal and non-principal 2-by-2 minors of the involved Hermitian matrix variable and characterize all such minors into three types. We show the equivalence of these minor constraints to the physical constraints of voltage angle differences summing to zero over three- and four-cycles in the power network. We study second-order conic programming (SOCP) relaxations of this minor reformulation and propose strong cutting planes, convex envelopes, and bound tightening techniques to strengthen the resulting SOCP relaxations. We then propose an SOCP-based spatial branch-and-cut method to obtain the global optimum of AC OPF. Extensive computational experiments show that the proposed algorithm significantly outperforms the state-of-the-art SDP-based OPF solver and on a simple personal computer is able to obtain on average a 0.71% optimality gap in no more than 720 seconds for the most challenging power system instances in the literature.

1 Introduction

1.1 Approaches to solving alternating current optimal power flow problem

Alternating current optimal power flow (AC OPF) is one of the most fundamental optimization problems in electrical power system operation. Since its introduction in 1962 [8], developing efficient algorithms for solving AC OPF has remained an active field of research. The main challenge in solving AC OPF in order to obtain a near-optimal primal solution and a high quality dual bound lies in the non-convex nature of the AC OPF model, the very large scale of the resulting optimization problem for real-world power grids, and the very short amount of solution time required by the real-time dispatch of the generation and demand resources in the power grid.

Literature on the AC OPF problem can be roughly categorized into three groups: local methods, relaxation approaches, and global optimization methods (e.g. see recent surveys in [38, 39, 15, 16, 7]). In the local methods, the aim is to find a local optimal solution or a stationary point of the OPF problem. Typically, these methods are based on the Newton-Raphson algorithm or the interior point methods (IPM) [51, 49, 24, 50, 53], and are effective in locating good feasible solutions only under normal operating conditions (e.g. when the power system is not stressed by high level of load and/or tight transmission constraints). However, since the OPF problem is non-convex, such local methods may get stuck at local optimal or stationary solutions [6, 28]. Also, local methods are sensitive to the initial point and may fail to produce solutions when a warm starting point is not available [28].

Relaxation approaches for solving AC OPF are mainly based on the semidefinite programming (SDP) relaxation techniques [2, 1, 29]. It recently become a popular approach as researchers found that the SDP relaxations of the standard IEEE test instances up to 300 buses all obtained globally optimal solutions [29]. However, it was soon realized that the requirements to guarantee exactness of the SDP relaxation are typically quite restrictive (e.g. resistive networks with no reactive loads [29], networks with a large number of virtual phase shifters [44], radial networks with no generation lower bounds [52, 5, 4]). Furthermore, when the relaxation is not exact, it is a challenge to recover a feasible primal solution. However, since the dual bounds obtained by SDP relaxations are typically strong, this approach proves to be a valuable tool for evaluating the quality of the primal feasible solutions obtained by other methods, e.g. a local method. Last but not the least, solving SDP is not very scalable with the existing algorithms. Methods exploiting the sparsity pattern of the underlying graph have been proposed for general polynomial optimization problems in in [17] and [41], and later applied to the OPF problem in [23, 36, 31, 37, 32]. Despite the considerable efforts, solving large-scale AC OPF with SDP relaxation is still a computational challenge for real-world power systems.

The scalability issue of SDP-based relaxations motivates researchers (including the present paper) to pursue simpler convex relaxations, such as second-order cone programming (SOCP) and linear programming (LP) relaxations. Basic SOCP relaxation is first applied to the OPF problem in [25]. Various strengthening techniques using valid inequalities, convex envelopes, and cutting planes are proposed to strengthen the SOCP relaxation, and the resulting strong SOCP relaxations prove to be not dominated by SDP relaxations and can be solved much faster [28, 27, 20, 12]. LP approximation and relaxation techniques are also applied to the OPF problem in [13, 3]. Global methods try to combine the above methods. A particular work in this area is [43], where a local solver (IPOPT) is used to find feasible primal solutions and a Lagrangian relaxation algorithm is used to obtain dual bounds in a spatial branch-and-bound framework. However, the proposed approach in [43] is only tested on standard IEEE instances, which turn out to be simple instances and can all be solvable in the root node. A recent work [10] also proposed a branch-and-cut algorithm for solving complex quadratic constrained quadratic programs with applications to AC OPF.

1.2 Our contributions

In this paper, we aim to solve the AC OPF problem to global optimum. We first propose a new systematic approach to dealing with the rank constraint of a complex Hermitian matrix in a nonconvex QCQP in lifted space. In particular, we use the fact that the enforcing the rank of a Hermitian matrix being equal to one is equivalent to imposing the condition that *all* the 2×2 minors of the matrix are zero. These minor constraints can be classified into three types. Each type can be further reformulated as a set of quadratic or bilinear constraints involving a small set of variables. This reformulation applies to any non-convex quadratic problem with Hermitian matrices and it lays the foundation for a systematic approach to relaxing and/or convexifying the rank constraints. Interestingly, these three types of minors also have clear interpretations. In particular, they correspond to edges, 3-cycles (i.e. cycles of three nodes), and 4-cycles (i.e. cycles of 4 nodes) of the underlying network. In the context of AC OPF, these cycle constraints have a clear physical meaning, e.g. voltage angle differences of edges in a cycle sum to zero. In [9, 10, 19], only *principal* 2×2 minors, corresponding to Type 1 minors in our language, are considered to improve the SDP relaxation. By considering the other two types of non-principal minors, we are able to obtain stronger relaxations than existing ones.

Based on this minor reformulation, we propose convexification techniques to exploit the special structures of cycles in a graph, and design and implement a complete spatial branch-

and-cut algorithm based on second-order conic programming (SOCP) for obtaining global or near global optimal solutions of AC OPF. Extensive computational experiments show that the proposed algorithm significantly outperforms the state-of-the-art SDP-based OPF solver and on a simple personal computer is able to obtain on average a 0.71% optimality gap in no more than 720 seconds for the most challenging power system instances in the literature.

The rest of the paper is organized as follows: In Section 2, we formally define the AC OPF problem. In Section 3, we give the new reformulation of a complex SDP problem with rank constraint as a real quadratic optimization problem with minor constraints. Then, we propose several outer-approximation schemes to incorporate the minor conditions in a convex relaxation of the AC OPF problem. Section 4 presents our SOCP based spatial branch-and-cut algorithm with a particular emphasis on the root node relaxation. Our extensive computational experiments on challenging NESTA instances [11] are summarized in Section 5. Finally, Section 6 concludes the paper with some further remarks and possible future research directions.

2 AC Optimal Power Flow Problem

In this section, we present the mathematical optimization formulation of the AC OPF problem using the so-called rectangular formulation. Consider a power network $\mathcal{N} = (\mathcal{B}, \mathcal{L})$, where $\mathcal{B}$ denotes the set of nodes (or buses), and $\mathcal{L}$ denotes the set of edges (or transmission lines). Electric power generators are attached to a subset of buses, which is denoted as $\mathcal{G} \subseteq \mathcal{B}$. We assume that there is electric demand (or load) at every bus, some of which might potentially be zero. The aim of the AC OPF problem is to determine generation levels of generators to satisfy demand with the minimum total generation cost in such a way that various physical constraints and operational constraints are satisfied.

Let $Y \in \mathbb{C}^{|\mathcal{B}|\times|\mathcal{B}|}$ denote the nodal admittance matrix, which has components $Y_{ij} = G_{ij} + \mathrm{i}B_{ij}$ for each line $(i,j) \in \mathcal{L}$, and $G_{ii} = g_{ii} - \sum_{j\neq i} G_{ij}$, $B_{ii} = b_{ii} - \sum_{j\neq i} B_{ij}$, where g_{ii} and b_{ii} are respectively the shunt conductance and susceptance at bus $i \in \mathcal{B}$. Here, $\mathrm{i} = \sqrt{-1}$. Let p_i^d and q_i^d be the real and reactive load at bus i given as part of the data. We also define decision variables p_i^g and q_i^g to represent the real and reactive power output of the generator at bus i. Another set of decision variables is the complex voltage (also called voltage phasor) V_i at bus i, which can be expressed in the rectangular form as $V_i = e_i + \mathrm{i}f_i$. It is sometimes convenient to represent V_i in the polar form as $V_i = |V_i|(\cos\theta_i + \mathrm{i}\sin\theta_i)$, where $|V_i|^2 = e_i^2 + f_i^2$ is the magnitude and θ_i is the angle of the complex voltage.

Finally, the AC OPF problem in the rectangular formulation is given below [8].

$$
\begin{aligned}
&\min \sum_{i \in \mathcal{G}} C_i(p_i^g) && && \text{(1a)}\\
&\text{s.t. } p_i^g - p_i^d = G_{ii}(e_i^2 + f_i^2) + \sum_{j \in \delta(i)} [G_{ij}(e_i e_j + f_i f_j) - B_{ij}(e_i f_j - e_j f_i)] && i \in \mathcal{B} && \text{(1b)}\\
&\quad q_i^g - q_i^d = -B_{ii}(e_i^2 + f_i^2) + \sum_{j \in \delta(i)} [-B_{ij}(e_i e_j + f_i f_j) - G_{ij}(e_i f_j - e_j f_i)] && i \in \mathcal{B} && \text{(1c)}\\
&\quad \underline{V}_i^2 \le e_i^2 + f_i^2 \le \overline{V}_i^2 && i \in \mathcal{B} && \text{(1d)}\\
&\quad p_i^{\min} \le p_i^g \le p_i^{\max} && i \in \mathcal{B} && \text{(1e)}\\
&\quad q_i^{\min} \le q_i^g \le q_i^{\max} && i \in \mathcal{B}. && \text{(1f)}\\
&\quad [-G_{ij}(e_i^2 + f_i^2) + G_{ij}(e_i e_j + f_i f_j) - B_{ij}(e_i f_j - e_j f_i)]^2 && && \\
&\quad + [B_{ij}(e_i^2 + f_i^2) - B_{ij}(e_i e_j + f_i f_j) - G_{ij}(e_i f_j - e_j f_i)]^2 \le (S_{ij}^{\max})^2 && (i,j) \in \mathcal{L} && \text{(1g)}
\end{aligned}
$$

In the OPF formulation, the objective function is separable over the generators and $C_i(p_i^g)$ is typically linear or convex quadratic in the real power output p_i^g. Constraints (1b) and (1c) correspond to the Kirchoff's Current Law, i.e. the conservation of active and reactive power flows at each bus, respectively, where $\delta(i)$ denotes the set of neighbors of bus i. Constraint (1d) limits voltage magnitude at each bus, which is usually normalized against a unit voltage level and is expressed in per unit (p.u.). Therefore, $\underline{V}_i$ and $\overline{V}_i$ are both close to 1 p.u. at each bus i. This strict requirement is crucial to maintain the stability across the power system. Constraints (1e) and (1f), respectively, restrict the active and reactive power output of each generator to their physical capability. Here, we set $p_i^{\min} = p_i^{\max} = q_i^{\min} = q_i^{\max} = 0$ for non-generator bus $i \in \mathcal{B} \setminus \mathcal{G}$. Finally, constraint (1g) puts an upper bound on the total electric power on each transmission line (i, j).

3 Minor-based Reformulation of AC OPF Problem

3.1 Standard rank formulation

The AC OPF problem given in (1) is a polynomial optimization problem involving quadratic and quartic polynomials, and it can be posed as a quadratically constrained quadratic program (QCQP) by rewriting the constraint (1g) using additional variables. For nonconvex QCQPs, a standard strategy is to lift the variable into a higher dimensional space [42]. This can be accomplished by defining a matrix variable to replace the quadratic terms in the original variables with linear ones in the lifted matrix variable. In this reformulation, the only nonconvex constraint is a requirement that the rank of the matrix variable is one, which we call the rank-one constraint. The procedure to obtain this lifted formulation for the OPF problem is given below.

First, let us define a Hermitian matrix $X \in \mathbb{C}^{|\mathcal{B}|\times|\mathcal{B}|}$, i.e., $X = X^*$, where X^* is the conjugate transpose of X. Consider the following set of constraints:

$$p_i^g - p_i^d = G_{ii}X_{ii} + \sum_{j\in\delta(i)} [G_{ij}\Re(X_{ij}) + B_{ij}\Im(X_{ij})] \qquad \forall i \in \mathcal{B} \tag{2a}$$

$$q_i^g - q_i^d = -B_{ii}X_{ii} + \sum_{j\in\delta(i)} [-B_{ij}\Re(X_{ij}) + G_{ij}\Im(X_{ij})] \qquad \forall i \in \mathcal{B} \tag{2b}$$

$$\underline{V}_i^2 \le X_{ii} \le \overline{V}_i^2 \qquad \forall i \in \mathcal{B} \tag{2c}$$

$$[-G_{ij}X_{ii} + G_{ij}\Re(X_{ij}) + B_{ij}\Im(X_{ij})]^2 + [B_{ij}X_{ii} - B_{ij}\Re(X_{ij}) + G_{ij}\Im(X_{ij})]^2 \le (S_{ij}^{\max})^2 \ \forall (i,j) \in \mathcal{L} \tag{2d}$$

$$X \text{ is Hermitian} \tag{2e}$$

$$X \succeq 0 \tag{2f}$$

$$\text{rank}(X) = 1, \tag{2g}$$

where $\Re(x)$ and $\Im(x)$ are the real and imaginary parts of the complex number x, respectively. Let V denote the vector of voltage phasors with the i-th entry $V_i = e_i + \mathrm{i} f_i$ for each bus $i \in \mathcal{B}$. Then, the AC OPF problem in (1) can be equivalently rewritten in the following *rank formulation*:

$$z_{\text{rank}} := \min \left\{ \sum_{i\in\mathcal{G}} C_i(p_i^g) : (2), (1e) - (1f) \right\}. \tag{3}$$

Let z_{ACOPF} denote the optimal value of the AC OPF problem in the rectangular formulation (1). Then, clearly $z_{\text{ACOPF}} = z_{\text{rank}}$. The nonconvexity in the rank formulation (3) is captured by the rank constraint (2g). In order to deal with this challenging constraint, in the following we propose a new reformulation of AC OPF, based on conditions on the minors of matrix X.

3.2 New approach: minor-based reformulation

In this section, we propose a new approach to reformulating the rank-one constraints. First, we recall the definition of an $m \times m$ minor of a matrix [22]:

Definition 3.1. *Let X be an $n \times n$ complex matrix. Then, an $m \times m$ minor of the matrix X for $1 \le m \le n$ is the determinant of an $m \times m$ submatrix of X obtained by deleting $n - m$ rows and columns from X.*

The following Proposition 3.1 provides the key characterization of the rank-one constraint in terms of 2×2 minors.

Proposition 3.1. *A nonzero, Hermitian matrix X is positive semidefinite and has rank one, i.e. $X \succeq 0$ and rank$(X) = 1$, if and only if all the 2×2 minors of X are zero and the diagonal elements of X are nonnegative.*

Proof. ($\Rightarrow$) Assume $X \succeq 0$ and $\operatorname{rank}(X) = 1$. Then, there exists a vector $x \in \mathbb{C}^n$ such that $X = xx^*$. Let us consider a 2×2 submatrix of X of the form

$$X_{(i,j),(k,l)} = \begin{bmatrix} X_{ik} & X_{il} \\ X_{jk} & X_{jl} \end{bmatrix}.$$

Note that we have $X_{ik} = x_i x_k^*$, $X_{il} = x_i x_l^*$, $X_{jk} = x_j x_k^*$, and $X_{jl} = x_j x_l^*$. Hence, $\det X_{(i,j),(k,l)} = 0$, implying that all the 2×2 minors of X are zero. Also, for any diagonal element of the matrix X, we have $X_{ii} = x_i x_i^* = |x_i|^2 \geq 0$. Therefore, we conclude that the diagonal elements of X are nonnegative.
($\Leftarrow$) Recall that $\operatorname{rank}(X)$ is equal to the size of the largest invertible submatrix of X [22]. Since all the 2×2 minors of X are zero, none of the 2×2 submatrices of X are invertible. This implies that $\operatorname{rank}(X) = 1$ as X is a nonzero matrix. Finally, since a rank-one, Hermitian matrix with nonnegative diagonal elements can be written in the form $X = xx^*$ for some vector $x \in \mathbb{C}^n$, we conclude that $X \succeq 0$. □

Based on this property of matrix minors, we can reformulate the AC OPF problem (1) as

$$z_{\text{minor}} = \min \left\{ \sum_{i \in \mathcal{G}} C_i(p_i^g) : \text{(2a)} - \text{(2f)}, \text{(1e)} - \text{(1f)},\ \text{all } 2 \times 2 \text{ minors of } X \text{ are zero} \right\}. \quad (4)$$

We call (4) the minor-based reformulation, which is an exact reformulation of (1). Therefore, we have $z_{\text{ACOPF}} = z_{\text{rank}} = z_{\text{minor}}$.

3.3 Discussion on relaxations

The rank (3) and the minor-based (4) reformulations lead to systematic ways to construct relaxations of the AC OPF problem (1). For example, the standard SDP relaxation for the OPF problem can be obtained by relaxing the rank constraint (2g), or equivalently, all the minor constraints in (4) as

$$z_{\text{SDP}} = \min \left\{ \sum_{i \in \mathcal{G}} C_i(p_i^g) : \text{(2a)} - \text{(2f)}, \text{(1e)} - \text{(1f)} \right\}. \quad (5)$$

By construction, $z_{\text{SDP}} \leq z_{\text{ACOPF}}$. The standard SDP relaxation has been shown to be exact for the standard IEEE test cases up to 300 buses [29, 33]. However, it is also shown by recent study that, for challenging test cases such as the instances from NESTA archive [11], the duality gap of the standard SDP relaxation (5) can be quite large [12]. Another challenge of (5) is the computational difficulty of solving large-scale SDP problems. Although sparsity exploitation methods help reduce the solution times significantly [37], for truly large-scale power systems, it is necessary to seek computationally less demanding methods.

Our approach is to avoid the SDP constraint (2f), which is computationally expensive to handle, and focus on providing efficient ways to incorporate convex relaxations of the minor constraints in the form of *linear* or *second-order conic* representable constraints. In this approach, some of the proposed constraints will be a convex relaxation of the 2×2 minor constraints, discussed in Sections 3.4 and 3.5; and some will be a weaker version of the semidefiniteness requirement on X, discussed in Sections 3.4 and 3.6. The computational cost of the proposed approach can be significantly lower than the standard SDP relaxation. Moreover, the convex relaxation of the minor constraints may provide additional strengthen that is not captured by the standard SDP relaxation. In this way, the proposed approach may be both faster and stronger than the SDP relaxation, which is indeed verified by extensive experiments on the challenging NESTA instances.

3.4 Analysis of 2×2 Minors

In this section, we analyze all the 2×2 minors of the matrix X in detail and characterize them into three different types. Note that for notational purposes, we will define $c_{ij} := \Re(X_{ij})$ and $s_{ij} := -\Im(X_{ij})$ for $i, j \in \mathcal{B}$ in the sequel.

(i) Type 1: Edge Minor. Let i and j be distinct elements of the set $\mathcal{B}$. Then, we have

$$\begin{vmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{vmatrix} = 0, \tag{6}$$

which is equivalent to

$$0 = X_{ii}X_{jj} - X_{ij}X_{ji} = c_{ii}c_{jj} - (c_{ij} - \mathrm{i}s_{ij})(c_{ij} + \mathrm{i}s_{ij}) = c_{ii}c_{jj} - (c_{ij}^2 + s_{ij}^2). \tag{7}$$

Note that this relation defines the boundary of the rotated SOCP cone in $\mathbb{R}^4$.

(ii) Type 2: 3-Cycle Minor. Let i, j and k be distinct elements of the set $\mathcal{B}$, assuming

$|\mathcal{B}| \geq 3$. Then, consider the following minor

$$\begin{vmatrix} X_{ii} & X_{ij} \\ X_{ki} & X_{kj} \end{vmatrix} = 0, \tag{8}$$

which is equivalent to

$$\begin{aligned} 0 =& X_{ii}X_{kj} - X_{ij}X_{ki} = c_{ii}(c_{kj} - \mathrm{i}s_{kj}) - (c_{ij} - \mathrm{i}s_{ij})(c_{ki} - \mathrm{i}s_{ki}) \\ =& (c_{ii}c_{kj} - c_{ij}c_{ki} + s_{ij}s_{ki}) - \mathrm{i}(c_{ii}s_{kj} - s_{ij}c_{ki} - c_{ij}s_{ki}). \end{aligned} \tag{9}$$

Note that this relation defines two bilinear equations in $\mathbb{R}^7$.

(iii) Type 3: 4-Cycle Minor. Let i, j, k and l be distinct elements of the set $\mathcal{B}$, assuming $|\mathcal{B}| \geq 4$. Then, consider the following minor

$$\begin{vmatrix} X_{ij} & X_{ik} \\ X_{lj} & X_{lk} \end{vmatrix} = 0, \tag{10}$$

which is equivalent to

$$\begin{aligned} 0 =& X_{ij}X_{lk} - X_{ij}X_{lj} = (c_{ij} - \mathrm{i}s_{ij})(c_{lk} - \mathrm{i}s_{lk}) - (c_{ij} - \mathrm{i}s_{ij})(c_{lj} - \mathrm{i}s_{lj}) \\ =& (c_{ij}c_{lk} - s_{ij}s_{lk} - c_{lj}c_{ik} + s_{lj}s_{ik}) - \mathrm{i}(s_{ij}c_{lk} - c_{ij}s_{lk} - s_{lj}c_{ik} - c_{lj}s_{ik}). \end{aligned} \tag{11}$$

Note that this relation defines two bilinear equations in $\mathbb{R}^8$.

We analyze these minors and their relaxations in detail below. At this point, we would like to contrast our approach with the existing literature. In [9, 10, 19], only *principal* 2×2 minors, corresponding to Type 1 minors in our language, are considered to improve the SDP relaxation. By considering Type 2 and 3 minors, we can potentially obtain stronger relaxations.

3.4.1 Type 1: Edge Minors

Type 1 minors are principal minors of the matrix X. The straightforward convex relaxation of (7) leads to an SOCP relaxation of the standard SDP relaxation of AC OPF and has been extensively used in the literature [25, 26, 9, 10, 19]. Our goal is to go beyond the simple SOCP relaxation and to obtain the convex hull description of the set defined by a Type 1 minor in (6) over the hypercube $\mathcal{D}_{ij} := [\underline{c}_{ii}, \overline{c}_{ii}] \times [\underline{c}_{jj}, \overline{c}_{jj}] \times [\underline{c}_{ij}, \overline{c}_{ij}] \times [\underline{s}_{ij}, \overline{s}_{ij}]$, where (i, j) is an edge in the power network. Specifically, we define the following nonconvex set.

$$\mathcal{K}^{=}_{ij} := \left\{(c_{ii}, c_{jj}, c_{ij}, s_{ij}) \in \mathcal{D}_{ij} : c_{ij}^2 + s_{ij}^2 = c_{ii}c_{jj}\right\}. \tag{12}$$

3.4.1.1 Convex hull description of $\mathcal{K}^{=}_{ij}$.

It follows from [47, Theorem 1] that

$$\mathrm{conv}(\mathcal{K}^{=}_{ij}) = \mathcal{K}^{\leq}_{ij} \cap \mathrm{conv}(\mathcal{K}^{\geq}_{ij}), \tag{13}$$

where $\mathcal{K}^{\leq}_{ij}$ and $\mathcal{K}^{\geq}_{ij}$ are defined accordingly. First of all, $\mathcal{K}^{\leq}_{ij}$ is a convex set (in fact, second-order cone representable as it is the rotated cone in $\mathbb{R}^4$). Therefore, in order to construct $\mathrm{conv}(\mathcal{K}^{=}_{ij})$, it suffices to find $\mathrm{conv}(\mathcal{K}^{\geq}_{ij})$, which is the intersection of a polytope and the complement of a convex set. Here, we use the following result:

Theorem 3.1. *(Theorem 1 in [21]) Let $P \subset \mathbb{R}^n$ be a nonempty polytope and G be a proper subset of $\mathbb{R}^n$ such that $\mathbb{R}^n \setminus G$ is convex. Then, conv$(P \cap G)$ is a polytope.*

The proof of Theorem 3.1 is constructive. Let E_l, $l = 1, \dots, L$ be all the one-dimensional faces of the polytope P, i.e. edges of P. Then, we have

$$\mathrm{conv}(P \cap G) = \mathrm{conv}\left(\bigcup_{l=1}^{L} \mathrm{conv}(E_l \cap G)\right). \tag{14}$$

Hence, $\mathrm{conv}(P \cap G)$ is precisely the convex hull of the extreme points of $\mathrm{conv}(E_l \cap G)$, $l = 1, \dots, L$.

We can apply Theorem 3.1 to our case, where $P = \mathcal{D}_{ij}$ and $G = \{(c_{ii}, c_{jj}, c_{ij}, s_{ij}) \in \mathbb{R}^4 : c_{ij}^2 + s_{ij}^2 \geq c_{ii}c_{jj}\}$. The hypercube $\mathcal{D}_{ij}$ has 32 one-dimensional faces. The extreme points of $\mathrm{conv}\left(\mathcal{K}^{\geq}_{ij}\right)$ can be easily obtain by fixing three of the variables to one of their bounds. The exact procedures to obtain extreme points are given in Algorithms 3, 4, and 5 (see Appendix B).

After we have computed the extreme points of one-dimensional faces, say z^k_{ij}, $k = 1, \dots, K$, where $z_{ij} := (c_{ii}, c_{jj}, c_{ij}, s_{ij})$ and $K \leq 32$, we can give the convex hull description of $\mathcal{K}^{\geq}_{ij}$ as follows:

$$\mathrm{conv}\left(\mathcal{K}^{\geq}_{ij}\right) = \mathrm{conv}\left(\{z^k_{ij}\}_{k=1}^{K}\right) = \left\{z_{ij} : \exists \lambda,\ z_{ij} = \sum_{k=1}^{K} \lambda_k z^k_{ij}, \sum_{k=1}^{K} \lambda_k = 1, \lambda \geq 0\right\}. \tag{15}$$

The above discussion can be summarized as the following result:

Proposition 3.2. *Let z_{ij}^k be computed using Algorithms 3-5. Then,*

$$\mathit{conv}(\mathcal{K}_{ij}^{=}) = \left\{ z_{ij} \in \mathcal{D}_{ij} : \exists \lambda : c_{ij}^2 + s_{ij}^2 \leq c_{ii} c_{jj},\ z_{ij} = \sum_{k=1}^{K} \lambda_k z_{ij}^k,\ \sum_{k=1}^{K} \lambda_k = 1, \lambda \geq 0 \right\}. \tag{16}$$

3.4.1.2 An Outer Approximation to conv($\mathcal{K}_{ij}^{=}$).

Note that Proposition 3.2 describes the convex hull of conv($\mathcal{K}_{ij}^{=}$) in an extended space of (z, λ), and the dimension of the λ variables, K, can be as large as 32. Directly incorporating the full convex hull description (16) for each edge in the power network could lead to a very large formulation. Instead, we propose an outer approximation to conv($\mathcal{K}_{ij}^{=}$) in the space of the z variable using only four linear inequalities. Our extensive experiments show that this approximation is quite accurate and computationally efficient.

We again focus on conv($\mathcal{K}_{ij}^{\geq}$) and rewrite the "reverse-cone" constraint as follows:

$$f(c_{ij}, s_{ij}) := \sqrt{c_{ij}^2 + s_{ij}^2} \geq \sqrt{c_{ii} c_{jj}} =: g(c_{ii}, c_{jj}). \tag{17}$$

Note that f is a convex function and g is a concave function. If we overestimate the former and underestimate the latter by hyperplanes, the inequality still holds. The following propositions formalize this idea:

Proposition 3.3. *The affine functions $g_m(c_{ii}, c_{jj}) := \nu_{ij}^m + \eta_{ij}^m c_{ii} + \delta_{ij}^m c_{jj}$, $m = 1, 2$, underestimate $\sqrt{c_{ii} c_{jj}}$ over the box $[\underline{c}_{ii}, \overline{c}_{ii}] \times [\underline{c}_{jj}, \overline{c}_{jj}]$, where*
$\eta_{ij}^1 = \frac{\sqrt{\underline{c}_{ii}}}{\sqrt{\underline{c}_{jj}} + \sqrt{\overline{c}_{jj}}}$, $\delta_{ij}^1 = \frac{\sqrt{\underline{c}_{jj}}}{\sqrt{\underline{c}_{ii}} + \sqrt{\overline{c}_{ii}}}$, $\nu_{ij}^1 = \sqrt{\underline{c}_{ii} \underline{c}_{jj}} - \eta_{ij}^1 \underline{c}_{ii} - \delta_{ij}^1 \underline{c}_{jj}$ *and*
$\eta_{ij}^2 = \frac{\sqrt{\overline{c}_{ii}}}{\sqrt{\underline{c}_{jj}} + \sqrt{\overline{c}_{jj}}}$, $\delta_{ij}^2 = \frac{\sqrt{\overline{c}_{jj}}}{\sqrt{\underline{c}_{ii}} + \sqrt{\overline{c}_{ii}}}$, $\nu_{ij}^2 = \sqrt{\overline{c}_{ii} \overline{c}_{jj}} - \eta_{ij}^2 \overline{c}_{ii} - \delta_{ij}^2 \overline{c}_{jj}$.

Proposition 3.4. *If $\sqrt{\overline{c}_{ij}^2 + \overline{s}_{ij}^2} + \sqrt{\underline{c}_{ij}^2 + \underline{s}_{ij}^2} - \sqrt{\overline{c}_{ij}^2 + \underline{s}_{ij}^2} - \sqrt{\underline{c}_{ij}^2 + \overline{s}_{ij}^2} < 0$, then the affine functions $f_n(c_{ij}, s_{ij}) := \nu_{ij}^n + \eta_{ij}^n c_{ij} + \delta_{ij}^l s_{ij}$, $n = 3, 4$, overestimate $\sqrt{c_{ij}^2 + s_{ij}^2}$ over the box $[\underline{c}_{ij}, \overline{c}_{ij}] \times [\underline{s}_{ij}, \overline{s}_{ij}]$, where*
$\eta_{ij}^3 = \frac{\sqrt{\overline{c}_{ij}^2 + \underline{s}_{ij}^2} - \sqrt{\underline{c}_{ij}^2 + \underline{s}_{ij}^2}}{\overline{c}_{ij} - \underline{c}_{ij}}$, $\delta_{ij}^3 = \frac{\sqrt{\underline{c}_{ij}^2 + \overline{s}_{ij}^2} - \sqrt{\underline{c}_{ij}^2 + \underline{s}_{ij}^2}}{\overline{s}_{ij} - \underline{s}_{ij}}$, $\nu_{ij}^3 = \sqrt{\underline{c}_{ij}^2 + \underline{s}_{ij}^2} - \eta_{ij}^3 \underline{c}_{ij} - \delta_{ij}^3 \underline{s}_{ij}$ *and*
$\eta_{ij}^4 = \frac{\sqrt{\overline{c}_{ij}^2 + \overline{s}_{ij}^2} - \sqrt{\underline{c}_{ij}^2 + \overline{s}_{ij}^2}}{\overline{c}_{ij} - \underline{c}_{ij}}$, $\delta_{ij}^4 = \frac{\sqrt{\overline{c}_{ij}^2 + \overline{s}_{ij}^2} - \sqrt{\overline{c}_{ij}^2 + \underline{s}_{ij}^2}}{\overline{s}_{ij} - \underline{s}_{ij}}$, $\nu_{ij}^4 = \sqrt{\overline{c}_{ij}^2 + \overline{s}_{ij}^2} - \eta_{ij}^4 \overline{c}_{ij} - \delta_{ij}^4 \overline{s}_{ij}$.

Proposition 3.5. *If $\sqrt{\overline{c}_{ij}^2 + \overline{s}_{ij}^2} + \sqrt{\underline{c}_{ij}^2 + \underline{s}_{ij}^2} - \sqrt{\overline{c}_{ij}^2 + \underline{s}_{ij}^2} - \sqrt{\underline{c}_{ij}^2 + \overline{s}_{ij}^2} > 0$, then the affine functions $f_n(c_{ij}, s_{ij}) := \nu_{ij}^n + \eta_{ij}^n c_{ij} + \delta_{ij}^n s_{ij}$, $n = 3, 4$, overestimate $\sqrt{c_{ij}^2 + s_{ij}^2}$ over the box $[\underline{c}_{ij}, \overline{c}_{ij}] \times [\underline{s}_{ij}, \overline{s}_{ij}]$, where*
$\eta_{ij}^3 = \frac{\sqrt{\overline{c}_{ij}^2 + \overline{s}_{ij}^2} - \sqrt{\underline{c}_{ij}^2 + \overline{s}_{ij}^2}}{\overline{c}_{ij} - \underline{c}_{ij}}$, $\delta_{ij}^3 = \frac{\sqrt{\underline{c}_{ij}^2 + \overline{s}_{ij}^2} - \sqrt{\underline{c}_{ij}^2 + \underline{s}_{ij}^2}}{\overline{s}_{ij} - \underline{s}_{ij}}$, $\nu_{ij}^3 = \sqrt{\underline{c}_{ij}^2 + \underline{s}_{ij}^2} - \eta_{ij}^3 \underline{c}_{ij} - \delta_{ij}^3 \underline{s}_{ij}$ *and*
$\eta_{ij}^4 = \frac{\sqrt{\overline{c}_{ij}^2 + \underline{s}_{ij}^2} - \sqrt{\underline{c}_{ij}^2 + \underline{s}_{ij}^2}}{\overline{c}_{ij} - \underline{c}_{ij}}$, $\delta_{ij}^4 = \frac{\sqrt{\overline{c}_{ij}^2 + \overline{s}_{ij}^2} - \sqrt{\overline{c}_{ij}^2 + \underline{s}_{ij}^2}}{\overline{s}_{ij} - \underline{s}_{ij}}$, $\nu_{ij}^4 = \sqrt{\overline{c}_{ij}^2 + \overline{s}_{ij}^2} - \eta_{ij}^4 \overline{c}_{ij} - \delta_{ij}^4 \overline{s}_{ij}$.

Proposition 3.6. *Let ν_{ij}^m, η_{ij}^m and δ_{ij}^m, $m = 1, 2$ and ν_{ij}^n, η_{ij}^n and δ_{ij}^n, $n = 3, 4$ be calculated using Propositions 3.3-3.5. Then, the edge cuts (EC) defined as*

$$EC_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}): \ \nu_{ij}^n + \eta_{ij}^n c_{ij} + \delta_{ij}^n s_{ij} \geq \nu_{ij}^m + \eta_{ij}^m c_{ii} + \delta_{ij}^m c_{jj}, \ m = 1, 2 \textit{ and } n = 3, 4 \tag{18}$$

are valid for conv($\mathcal{K}_{ij}^{=}$).

A different analysis related to Type 1 minor condition is carried out in [10] by considering the following set:

$$\begin{aligned} \mathcal{K}'_{ij} = \{(c_{ii}, c_{jj}, c_{ij}, s_{ij}) : c_{ij}^2 + s_{ij}^2 = c_{ii}c_{jj}, (c_{ii}, c_{jj}) \in [\underline{c}_{ii}, \overline{c}_{ii}] \times [\underline{c}_{jj}, \overline{c}_{jj}], \\ c_{ij} \tan \underline{\theta}_{ij} \leq s_{ij} \leq c_{ij} \tan \overline{\theta}_{ij}\}. \end{aligned} \tag{19}$$

It turns out that conv($\mathcal{K}'_{ij}$) is second-order cone representable with two non-trivial linear inequalities. In our experiments, we observe that the addition of these valid linear inequalities does not produce any extra gap closure in our approach, and hence, they are not used in our relaxation scheme.

3.4.2 Types 2 and 3: 3- and 4-Cycle Minors

The real and imaginary parts of Type 2 and Type 3 minors in (8) and (10) can be written generically as $\sum_{i=1}^N a_i x_i y_i = 0$ for some $a \in \mathbb{R}^N$ with $a_i \neq 0$. Let $\underline{x}, \overline{x}, \underline{y}, \overline{y}$ be N-vectors with the property that $\underline{x} < \overline{x}$ and $\underline{y} < \overline{y}$. We are interested in finding the convex hull of the following set:

$$\mathcal{S}_a = \left\{(x, y) \in \mathbb{R}^N \times \mathbb{R}^N : \sum_{i=1}^N a_i x_i y_i = 0, \ \underline{x} \leq x \leq \overline{x}, \ \underline{y} \leq y \leq \overline{y}\right\}. \tag{20}$$

We have the following theorem, whose proof is given in Appendix C:

Theorem 3.2. *conv($\mathcal{S}_a$) is second-order cone representable.*

In the proof of Theorem 3.2, we construct the convex hull of conv(S_a) in an extended space with the number of disjunctions exponential in N, where some of the disjunctions may contain second-order conic constraints. Including the exact description of conv(S_a) as part of the relaxation might be quite costly. In the following, we propose tight outer approximation of conv(S_a).

3.4.2.1 An Outer Approximation to conv(S_a). We propose a linear outer approximation to conv(S_a) using McCormick envelopes [34] as follows:

$$\begin{aligned}\mathcal{S}_a^M = \{(x, y) \in \mathbb{R}^N \times \mathbb{R}^N : \exists w \in \mathbb{R}^N : &\sum_{i=1}^{N} a_i w_i = 0, \\ \max\{\underline{y}x + \underline{x}y - \underline{x}\underline{y}, \overline{y}x + \overline{x}y - \overline{x}\overline{y}\} \leq w &\leq \min\{\underline{y}x + \overline{x}y - \overline{x}\underline{y}, \overline{y}x + \underline{x}y - \underline{x}\overline{y}\}\}.\end{aligned} \tag{21}$$

Our extensive tests show that $\mathcal{S}_a^M$ tightly approximates conv($\mathcal{S}_a$).

3.4.2.2 A Cycle Based Relaxation Although we have linear outer approximations of Type 2 and 3 minors, their total number is cubic and quartic in the number of buses. Therefore, including relaxations for every minor can be quite expensive. Instead, we focus on a set of cycles in the graph and include a subset of minors for each cycle. This corresponds to triangulating a given cycle into 3- and 4-cycles. This idea is similar to the one used in our previous paper [27] although the construction in that paper is entirely different.

We do not propose to include all such minors in our relaxation scheme either. Rather, we construct a cycle relaxation and use it as a basis to generate cutting planes. More precisely, let C be a given cycle in the power network and X^C denote the principal submatrix of X, which corresponds to the rows and columns of the nodes in the cycle C. Next, we choose a subset of Type 2 and 3 minors from X^C and define the following set

$$\mathcal{Q}_C = \{(c, s) : \exists(\tilde{c}, \tilde{s}) : q_k(c, s, \tilde{c}, \tilde{s}) = 0, \ k \in \mathcal{K}_C\}, \tag{22}$$

where q_k's are bilinear equations corresponding to minor constraints (9) and (11) index by $\mathcal{K}_C$. Here, we denote the variables associated with original edge (i, j) in the power network as c_{ij}, s_{ij}, and denote $\tilde{c}_{ij}, \tilde{s}_{ij}$ for an artificial edge added to triangulate the cycle C. See Figure 1 for an illustration on how a 7-cycle can be triangulated.

Finally, we write the McCormick relaxation for the nonconvex set $\mathcal{Q}_C$, parametrized by the variable bounds $\underline{c}, \overline{c}, \underline{s}, \overline{s}$, for a given cycle C, compactly, as follows:

$$\mathcal{M}_C(\underline{c}, \overline{c}, \underline{s}, \overline{s}) = \left\{(c, s) : \exists(\tilde{c}, \tilde{s}, w) : A \begin{bmatrix} c \\ s \end{bmatrix} + \tilde{A} \begin{bmatrix} \tilde{c} \\ \tilde{s} \end{bmatrix} + Bw \leq c, \ Ew = 0\right\}. \tag{23}$$

Here, w is a vector of new variables defined to linearize the bilinear terms in the minor constraints. Inequality constraints contain the McCormick envelopes of the bilinear terms and bounds on the variables, while equality constraints include the linearized minor equality constraints.

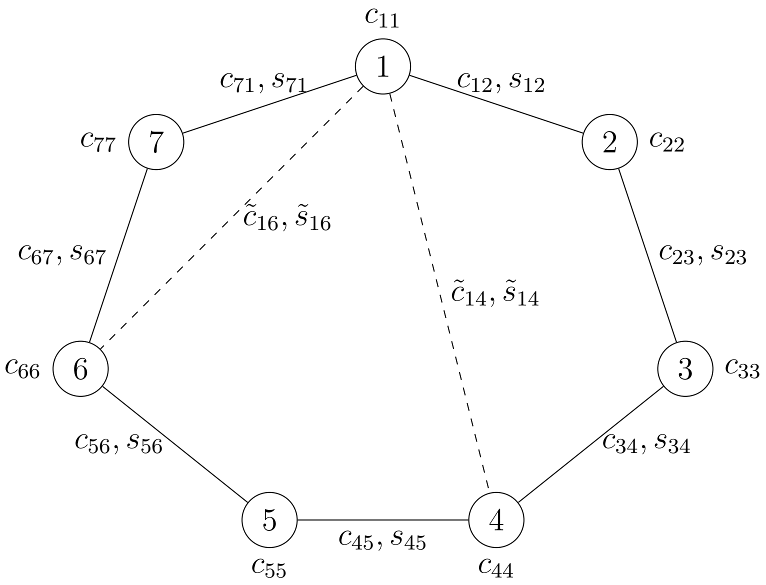

Figure 1: Triangulation of a 7-cycle. Here, we consider three minors corresponding to the "subcycles" $\{1, 2, 3, 4\}$, $\{1, 4, 5, 6\}$ and $\{1, 6, 7\}$. Variables corresponding to each node and edge are also shown.

3.4.2.3 Discretization Our preliminary analysis has shown that $\mathcal{M}_C$ does not approximate $\text{conv}(\mathcal{Q}_C)$ accurately in most cases. However, it is well-known that McCormick relaxation of a nonconvex set converges to the convex hull of the set as the variable ranges shrink. Motivated by this fact, we propose a discretization technique to improve the accuracy of the set $\mathcal{M}_C$. We note that this idea has been applied to other nonconvex problems in the literature (e.g. pooling problem [35, 14, 18]).

To begin with, let $[\underline{c}_d, \overline{c}_d] \times [\underline{s}_d, \overline{s}_d]$, for $d \in D$, be a partition of the initial box $[\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}]$. Then, the following relations trivially hold:

$$\mathcal{M}_C^D := \text{conv}\left(\bigcup_{d \in D} \mathcal{M}_C(\underline{c}_d, \overline{c}_d, \underline{s}_d, \overline{s}_d)\right) \subseteq \mathcal{M}_C(\underline{c}, \overline{c}, \underline{s}, \overline{s}) \quad \text{and} \quad \text{conv}(\mathcal{Q}_D) \subseteq \mathcal{M}_C^D. \tag{24}$$

Note that since $\bigcup_{d \in D} \mathcal{M}_C(\underline{c}_d, \overline{c}_d, \underline{s}_d, \overline{s}_d)$ is a finite union of polyhedral representable sets, $\mathcal{M}_C^D$ is also polyhedral representable.

In our implementation, we use the following construction. First, we decide a set of variables and bisect their variable ranges to obtain the collection D. Then, we construct the set $\mathcal{M}_C^D$. Finally, we solve the separation problem presented in Section 3.7 to obtain cutting planes valid for $\mathcal{M}_C^D$.

After initial calibration, we decided to choose the collection D as follows. We first choose a reference node to start triangulation. Then, for each subcycle of a cycle, we pick the edges which are neither the first nor the last line in a subcycle and apply bisection to the corresponding c_{ij} and s_{ij} variables. For instance, in Figure 1, variables $c_{23}, s_{23}, c_{34}, s_{34}, c_{45}, s_{45}, c_{56}, s_{56}, c_{67}, s_{67}$ are bisected.

3.5 An Alternative to Type 2 and 3 Minor Conditions: Arctangent Constraints

In this section, we propose another equivalent characterization of the rank-one, or equivalently 2×2 minors requirement explained above. Our alternative condition is based on the following relationship between the phase angles θ and c, s variables, which correspond to the real and imaginary parts of the complex matrix variable X using the $\operatorname{atan2}(y,x)$[1] function:

$$\theta_j - \theta_i = \operatorname{atan2}(s_{ij}, c_{ij}) \quad i, j \in \mathcal{B} \tag{25}$$

3.5.1 Equivalence

First, we claim that Type 1 minor constraint (6) together with arctangent constraint (25) implies Type 2 minor (8) equations.

Proposition 3.7.

$$\left\{(c,s) : \begin{vmatrix} X_{ii} & X_{ij} \\ X_{ki} & X_{kj} \end{vmatrix} = 0\right\} \supseteq \left\{(c,s) : \exists\theta : \begin{vmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{vmatrix} = \begin{vmatrix} X_{kk} & X_{ki} \\ X_{ik} & X_{ii} \end{vmatrix} = \begin{vmatrix} X_{kk} & X_{kj} \\ X_{jk} & X_{jj} \end{vmatrix} = 0,\right.$$
$$\left.\theta_j - \theta_i = \operatorname{atan2}(s_{ij}, c_{ij}), \theta_i - \theta_k = \operatorname{atan2}(s_{ki}, c_{ki}), \theta_j - \theta_k = \operatorname{atan2}(s_{kj}, c_{kj})\right\}.$$

Proof. From (25), we have that $c_{ij} = \sqrt{c_{ii}c_{jj}}\cos(\theta_j - \theta_i)$ and $s_{ij} = \sqrt{c_{ii}c_{jj}}\sin(\theta_j - \theta_i)$. Then,

$$\begin{aligned} c_{ii}c_{kj} - c_{ij}c_{ki} + s_{ij}s_{ki} =& c_{ii}\sqrt{c_{kk}c_{jj}}\cos(\theta_j - \theta_k) - \sqrt{c_{ii}c_{jj}}\cos(\theta_j - \theta_i)\sqrt{c_{kk}c_{ii}}\cos(\theta_i - \theta_k) \\ & + \sqrt{c_{ii}c_{jj}}\sin(\theta_j - \theta_i)\sqrt{c_{kk}c_{ii}}\sin(\theta_i - \theta_k) \\ =& c_{ii}\sqrt{c_{jj}c_{kk}}[\cos(\theta_j - \theta_k) - \cos(\theta_j - \theta_i)\cos(\theta_i - \theta_k) + \sin(\theta_j - \theta_i)\sin(\theta_i - \theta_k)] \\ =& 0, \end{aligned}$$

[1] $$\operatorname{atan2}(y,x) = \begin{cases} \arctan\frac{y}{x} & x > 0 \\ \arctan\frac{y}{x} + \pi & y \geq 0, x < 0 \\ \arctan\frac{y}{x} - \pi & y < 0, x < 0 \\ +\frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

and

$$\begin{aligned}c_{ii}s_{kj} - s_{ij}c_{ki} - c_{ij}s_{ki} =& c_{ii}\sqrt{c_{kk}c_{jj}}\sin(\theta_j - \theta_k) - \sqrt{c_{ii}c_{jj}}\sin(\theta_j - \theta_i)\sqrt{c_{kk}c_{ii}}\cos(\theta_i - \theta_k) \\ &- \sqrt{c_{ii}c_{jj}}\cos(\theta_j - \theta_i)\sqrt{c_{kk}c_{ii}}\sin(\theta_i - \theta_k) \\ =& c_{ii}\sqrt{c_{jj}c_{kk}}[\sin(\theta_j - \theta_k) - \sin(\theta_j - \theta_i)\cos(\theta_i - \theta_k) - \cos(\theta_j - \theta_i)\sin(\theta_i - \theta_k)] \\ =& 0,\end{aligned}$$

which imply Type 2 minor (8). □

A similar proposition about Type 3 minors is also true, that is, Type 1 minor constraint (6) together with arctangent constraint (25) implies Type 3 minor (10) equations.

Proposition 3.8.

$$\left\{(c,s) : \begin{vmatrix} X_{ij} & X_{ik} \\ X_{lj} & X_{lk} \end{vmatrix} = 0\right\} \supseteq \left\{(c,s) : \exists\theta : \begin{vmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{vmatrix} = \begin{vmatrix} X_{ii} & X_{ik} \\ X_{ki} & X_{kk} \end{vmatrix} = \begin{vmatrix} X_{jj} & X_{jl} \\ X_{lj} & X_{ll} \end{vmatrix} = 0, \begin{vmatrix} X_{kk} & X_{kl} \\ X_{lk} & X_{ll} \end{vmatrix} = 0,\right.$$
$$\left.\theta_j - \theta_i = \operatorname{atan2}(s_{ij}, c_{ij}), \theta_k - \theta_i = \operatorname{atan2}(s_{ik}, c_{ik}), \theta_j - \theta_l = \operatorname{atan2}(s_{lj}, c_{lj}), \theta_l - \theta_l = \operatorname{atan2}(s_{lk}, c_{lk})\right\}.$$

We omitted the proof of Proposition 3.8 due to its similarity to the proof of Proposition 3.7.

3.5.2 Convexification

In Section 3.4.2, we analyzed Type 2 and 3 minors and proposed a method to obtain a linear outer-approximation. Now, we propose another linearization method using the arctangent restriction (25). To start with, let us define the following nonconvex set

$$\mathcal{AT} := \left\{(c,s,\theta) \in \mathbb{R}^3 : \theta = \arctan\left(\frac{s}{c}\right), (c,s,\theta) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}] \times [\underline{\theta}, \overline{\theta}]\right\}, \tag{26}$$

where we denote $\theta = \theta_j - \theta_i$ and drop (i,j) indices for brevity. We also assume $\underline{c} > 0$. Let us denote the four corners of the box in (c,s,θ) space as follows:

$$\begin{aligned}\zeta^1 &= (\underline{c}, \overline{s}, \arctan(\overline{s}/\underline{c})), \quad \zeta^2 = (\overline{c}, \overline{s}, \arctan(\overline{s}/\overline{c})), \\ \zeta^3 &= (\overline{c}, \underline{s}, \arctan(\underline{s}/\overline{c})), \quad \zeta^4 = (\underline{c}, \underline{s}, \arctan(\underline{s}/\underline{c})).\end{aligned} \tag{27}$$

Two inequalities that approximate the upper envelope of $\mathcal{AT}$ are described below.

Proposition 3.9. *Let $\theta = \gamma_1 + \alpha_1 c + \beta_1 s$ and $\theta = \gamma_2 + \alpha_2 c + \beta_2 s$ be the planes passing through points $\{\zeta^1, \zeta^2, \zeta^3\}$, and $\{\zeta^1, \zeta^3, \zeta^4\}$, respectively. Then, two valid inequalities for $\mathcal{AT}$*

can be obtained as

$$\bar{\gamma}_m + \alpha_m c + \beta_m s \geq \arctan\left(\frac{s}{c}\right) \tag{28}$$

for all $(c, s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}]$ *with* $\bar{\gamma}_m = \gamma_m + \Delta\gamma_m$*, where*

$$\Delta\gamma_m = \max\left\{\arctan\left(\frac{s}{c}\right) - (\gamma_m + \alpha_m c + \beta_m s) : (c, s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}],\ c\tan\underline{\theta} \leq s \leq c\tan\overline{\theta}\right\}, \tag{29}$$

for $m = 1, 2$*.*

Note that by the construction of (29), it is evident that $\bar{\gamma}_m + \alpha_m c + \beta_m s$ dominates the $\arctan(s/c)$ over the box. The nonconvex optimization problem (29) can be solved by enumerating all possible Karush-Kuhn-Tucker (KKT) points. See Appendix A for details. These inequalities are improvements over the similar ones in [27] since the bounds on θ variables are also taken into consideration in the calculation of the offset value $\Delta\gamma_m$.

Two inequalities that approximate the lower envelope of $\mathcal{AT}$ are described below.

Proposition 3.10. *Let* $\theta = \gamma_3 + \alpha_3 c + \beta_3 s$ *and* $\theta = \gamma_4 + \alpha_4 c + \beta_4 s$ *be the planes passing through points* $\{\zeta^1, \zeta^2, \zeta^4\}$*, and* $\{\zeta^2, \zeta^3, \zeta^4\}$*, respectively. Then, two valid inequalities for* $\mathcal{AT}$ *are defined as*

$$\bar{\gamma}_n + \alpha_n c + \beta_n s \leq \arctan\left(\frac{s}{c}\right) \tag{30}$$

for all $(c, s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}]$ *with* $\bar{\gamma}_n = \gamma_n - \Delta\gamma_n$*, where*

$$\Delta\gamma_n = \max\left\{(\gamma_n + \alpha_n c + \beta_n s) - \arctan\left(\frac{s}{c}\right) : (c, s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}],\ c\tan\underline{\theta} \leq s \leq c\tan\overline{\theta}\right\}, \tag{31}$$

for $n = 3, 4$*.*

In summary, the four linear inequalities that approximate $\text{conv}(\mathcal{AT}_{ij})$ are given as

$$\text{AT}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) :\ \bar{\gamma}^m_{ij} + \alpha^m_{ij} c_{ij} + \beta^m_{ij} s_{ij} \leq \theta_j - \theta_i \leq \bar{\gamma}^n_{ij} + \alpha^n_{ij} c_{ij} + \beta^n_{ij} s_{ij},\ m = 1, 2 \text{ and } n = 3, 4 \tag{32}$$

for some line (i, j).

3.6 General Principal Submatrices

Up until this point, we have mainly discussed how to incorporate the minor constraints into our relaxation scheme. In this section, we propose an approach to include a relaxed version of the positive semidefiniteness constraint by considering general principal minors of the matrix variable X. Our approach is a relaxation of the positive semidefiniteness requirement since $X \succeq 0$ implies that *all* principal submatrices should be positive semidefinite while we only include a few hyperplanes which outer-approximate $X^s \succeq 0$ for some principal submatrix X^s. From a different view point, the approach in this section can be seen as a simultaneous convexification of several, appropriately chosen Type 2 and 3 minor conditions (9) and (11).

Let $\mathcal{B}'$ be a subset of buses. Let $x \in \mathbb{R}^{2|\mathcal{B}'|}$ be a vector of bus voltages defined as $x = [e^T; f^T]^T$ such that $x_i = e_i$ for $i \in \mathcal{B}'$ and $x_{i'} = f_i$ for $i' = i + |\mathcal{B}'|$. Observe that the following linear relationship between c, s and W holds,

$$\begin{aligned}
c_{ij} &= e_ie_j + f_if_j = W_{ij} + W_{i'j'} && i,j \in \mathcal{B}' && \text{(33a)}\\
s_{ij} &= e_if_j - e_jf_i = W_{ij'} - W_{ji'} && i,j \in \mathcal{B}' && \text{(33b)}\\
c_{ii} &= e_i^2 + f_i^2 = W_{ii} + W_{i'i'}, && i \in \mathcal{B}' && \text{(33c)}\\
W &= xx^T. &&&& \text{(33d)}
\end{aligned}$$

Here, we used real matrices instead of complex matrices for convenience. It is proven in [48] that using real matrices is equivalent to complex matrices.

Clearly, the set defined by (33) is nonconvex. A straightforward SDP relaxation can be presented as follows:

$$\begin{aligned}
\mathcal{S}_{\mathcal{B}'} := \{(c,s) \in \mathbb{R}^{2|\mathcal{B}'|} : \exists W \in \mathbb{S}_+^{2|\mathcal{B}'|},\ c_{ij} &= W_{ij} + W_{i'j'} \quad i,j \in \mathcal{B}' \\
s_{ij} &= W_{ij'} - W_{ji'} \quad i,j \in \mathcal{B}' \\
c_{ii} &= W_{ii} + W_{i'i'} \quad i \in \mathcal{B}' \ \}.
\end{aligned} \tag{34}$$

Note that this relaxation is a further relaxation of the SDP relaxation since only one principal submatrix is considered here. Although the subset $\mathcal{B}'$ can be general, previous experience [27] shows that it makes sense to use a subset of buses that correspond to a cycle. In particular, we define the following set $\mathcal{S}_C$ for a cycle C

$$\begin{aligned}
\mathcal{S}_C := \{(c,s) \in \mathbb{R}^{2|C|} : \exists W \in \mathbb{S}_+^{2|C|},\ c_{ij} &= W_{ij} + W_{i'j'} \quad (i,j) \in C \\
s_{ij} &= W_{ij'} - W_{ji'} \quad (i,j) \in C \\
c_{ii} &= W_{ii} + W_{i'i'} \quad i \in \{k : (k,l) \in C\} \ \},
\end{aligned} \tag{35}$$

and then, use the following procedure in Section 3.7 to obtain cutting planes for this set.

3.7 Separation Problems for LP and SDP Based Cycle Relaxations

In Sections 3.4.2 and 3.6, we presented LP and SDP based approximations for the cycles in our problem. In this section, we propose to utilize these relaxations in a cutting plane framework.

Suppose that we are given a solution (c^*, s^*) and we want to determine whether this point belongs to either LP or SDP relaxation given in (23) or (35). In order to cover both cases, let us focus on a generic setting where we have a conic representable set $S = \{x : \exists u, Ax + Bu \succeq_K b\}$ and a point x^* that we want to separate. Here, we assume that the cone K is either the nonnegative orthant or the cone of positive semidefinite matrices with appropriate dimension. Suppose we want to determine if the given point x^* is in S, or find a separating hyperplane $\alpha^T x \geq \beta$ such that $\alpha^T x^* < \beta$ otherwise. This problem can be formulated as

$$\max_{\alpha,\beta} \left\{\beta - \alpha^T x^* : \alpha^T x \geq \beta \; \forall x \in S, \; \|\alpha\|_\infty \leq 1\right\}.$$

Clearly, the optimal value of this problem without the norm constraint is unbounded if $x^* \notin S$. In order to find a cutting plane in the proposed form, we dualize the constraint and normalize the α vector to be in the unit ℓ_∞ ball. The resulting problem of separating a given point x^* from a conic representable set S is given as follows:

$$\text{SEP}(S, x^*) : Z^* := \max_{\alpha,\beta,\mu}\{\beta - \alpha^T x^* : b^T\mu \geq \beta, A^T\mu = \alpha, B^T\mu = 0, \mu \in K, -e \leq \alpha \leq e\}. \quad (36)$$

Here, we have two cases: If $Z^* \leq 0$, then $x^* \in S$, otherwise, the optimal solution (α^*, β^*) from (36) gives a separating hyperplane of the form $\alpha^T x \geq \beta$, which then can be used as a cutting plane.

3.8 Bound Tightening

So far, one of the standing assumptions for the construction of McCormick relaxations and arctangent outer-approximation was the availability of lower and upper bounds on c and s variables. Clearly, tighter variable bounds will lead to better relaxations. In this section, we explain how good bounds can be obtained by first solving small size bounding SOCPs and then, improving these bounds further by incorporating some dual information.

3.8.1 Optimization-Based Bound Tightening

Assuming some angle bounds for a line $(i, j) \in \mathcal{L}$ as

$$\underline{\theta}_{ij} \leq \theta_i - \theta_j \leq \overline{\theta}_{ij}, \quad (37)$$

we can obtain a rough first estimate for the bounds on c and s as follows:

$$\underline{c}_{ij} = \underline{V}_i \underline{V}_j \cos(\underline{\theta}_{ij}), \qquad \overline{c}_{ij} = \overline{V}_i \overline{V}_j, \tag{38a}$$

$$\underline{s}_{ij} = \overline{V}_i \overline{V}_j \sin(\underline{\theta}_{ij}), \qquad \overline{s}_{ij} = \overline{V}_i \overline{V}_j \sin(\overline{\theta}_{ij}). \tag{38b}$$

We claim that these bounds can be further tightened by solving SOCP bounding problems. Let us consider a line $(k, l) \in \mathcal{L}$ and fix a "closeness" parameter r. We first define the following sets: $\mathcal{B}_{kl}(r)$, the set of buses which can be reached from either k or l in at most r steps, and $\mathcal{L}_{kl}(r)$, the set of lines incident to at least one bus in $\mathcal{B}_{kl}(r)$. Consider the following second-order cone representable set,

$$p_i^g - p_i^d = G_{ii} c_{ii} + \sum_{j \in \delta(i)} [G_{ij} c_{ij} - B_{ij} s_{ij}] \qquad i \in \mathcal{B}_{kl}(r), \tag{39a}$$

$$q_i^g - q_i^d = -B_{ii} c_{ii} + \sum_{j \in \delta(i)} [-B_{ij} c_{ij} - G_{ij} s_{ij}] \qquad i \in \mathcal{B}_{kl}(r), \tag{39b}$$

$$\underline{V}_i^2 \leq c_{ii} \leq \overline{V}_i^2 \qquad i \in \mathcal{B}_{kl}(r+1), \tag{39c}$$

$$p_i^{\min} \leq p_i^g \leq p_i^{\max} \qquad i \in \mathcal{B}_{kl}(r), \tag{39d}$$

$$q_i^{\min} \leq q_i^g \leq q_i^{\max} \qquad i \in \mathcal{B}_{kl}(r), \tag{39e}$$

$$c_{ij} = c_{ji}, \quad s_{ij} = -s_{ji} \qquad (i,j) \in \mathcal{L}_{kl}(r), \tag{39f}$$

$$[-G_{ij} c_{ii} + G_{ij} c_{ij} - B_{ij} s_{ij}]^2 + [B_{ij} c_{ii} - B_{ij} c_{ij} - G_{ij} s_{ij}]^2 \leq (S_{ij}^{\max})^2 \qquad (i,j) \in \mathcal{L}_{kl}(r), \tag{39g}$$

$$c_{ij}^2 + s_{ij}^2 \leq c_{ii} c_{jj} \qquad (i,j) \in \mathcal{L}_{kl}(r), \tag{39h}$$

$$\mathrm{EC}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) \qquad (i,j) \in \mathcal{L}_{kl}(r), \tag{39i}$$

$$\mathrm{AT}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) \qquad (i,j) \in \mathcal{L}_{kl}(r), \tag{39j}$$

$$\underline{\theta}_{ij} \leq \theta_i - \theta_j \leq \overline{\theta}_{ij} \qquad (i,j) \in \mathcal{L}_{kl}(r), \tag{39k}$$

$$\underline{c}_{ij} \leq c_{ij} \leq \overline{c}_{ij}, \quad \underline{s}_{ij} \leq s_{ij} \leq \overline{s}_{ij} \qquad (i,j) \in \mathcal{L}_{kl}(r), \tag{39l}$$

where EC and AT are defined as in (18) and (32), respectively. This set is an improved version of the one proposed in [27], which does not contain edge cut inequalities (39i) for Type 1 minors or arctangent envelopes (39j) .

Let us now define the following problems:

$$\begin{aligned}
&\underline{P}_{kl}^c(\underline{c}, \overline{c}, \underline{s}, \overline{s}, r) : \ \underline{c}_{kl}^* = \min\{c_{kl} : (39)\}, \\
&\overline{P}_{kl}^c(\underline{c}, \overline{c}, \underline{s}, \overline{s}, r) : \ \overline{c}_{kl}^* = \max\{c_{kl} : (39)\}, \\
&\underline{P}_{kl}^s(\underline{c}, \overline{c}, \underline{s}, \overline{s}, r) : \ \underline{s}_{kl}^* = \min\{s_{kl} : (39)\}, \\
&\overline{P}_{kl}^s(\underline{c}, \overline{c}, \underline{s}, \overline{s}, r) : \ \overline{s}_{kl}^* = \max\{s_{kl} : (39)\}.
\end{aligned} \tag{40}$$

For improved numerical stability, we update a variable bound only if it is improved by at least 10^{-3}. As an implementation note, since these problems are independent of each other for different edges, they can be solved in parallel. According to our experiments, this synchronous parallelization saves a significant amount of computational time.

For artificial edges, it is not possible to use the above procedure as they do not appear in the flow balance constraints. However, we can utilize the bounds already computed for the original edges to obtain some bounds for the variables defined for the artificial edges by adopting the procedure proposed in [27].

3.8.2 Dual-Based Bound Tightening

Let us suppose that the problems (40) have been solved and we have updated the variable bounds on c and s variables to $\underline{c}^*$, $\overline{c}^*$, $\underline{s}^*$ and $\overline{s}^*$. Note that while solving these problems, the existing variable bounds are used, that is, the bounds are not updated. A simple way to incorporate the change in one problem to another is to use the dual variables. Related methods have been used by LP based global solvers [45]. Let us now formally explain how this can be accomplished.

Consider the problem $\underline{P}^c_{kl}$. Let $\underline{\pi}^c_{kl}(\underline{c}_{ij})$, $\underline{\pi}^c_{kl}(\overline{c}_{ij})$, $\underline{\pi}^c_{kl}(\underline{s}_{ij})$ and $\underline{\pi}^c_{kl}(\overline{s}_{ij})$ be the optimal dual variables corresponding to the constraints $c_{ij} \geq \underline{c}_{ij}$, $c_{ij} \leq \overline{c}_{ij}$, $s_{ij} \geq \underline{s}_{ij}$ and $s_{ij} \leq \overline{s}_{ij}$, respectively. First, we calculate the contribution of the constraints other than the bounds on the dual objective as

$$\Pi^c_{kl} = \underline{c}^*_{kl} - \sum_{(i,j)\in\in\mathcal{L}_{kl}(r)} \left(\underline{c}_{ij}\underline{\pi}^c_{kl}(\underline{c}_{ij}) + \overline{c}_{ij}\underline{\pi}^c_{kl}(\overline{c}_{ij}) + \underline{s}_{ij}\underline{\pi}^c_{kl}(\underline{s}_{ij}) + \overline{s}_{ij}\underline{\pi}^c_{kl}(\overline{s}_{ij}) \right). \tag{41}$$

Now, since the bounds on c_{ij} and s_{ij} variables are improved via their own bounding problems $\underline{P}^c_{ij}$, $\overline{P}^c_{ij}$, $\underline{P}^s_{ij}$ and $\overline{P}^s_{ij}$, the lower bound on c_{kl} can be updated as follows:

$$\underline{c}^*_{kl} = \max\left\{ \underline{c}^*_{kl},\ \Pi^c_{kl} + \sum_{(i,j)\in\in\mathcal{L}_{kl}(r)} \left(\underline{c}^*_{ij}\underline{\pi}^c_{kl}(\underline{c}_{ij}) + \overline{c}^*_{ij}\underline{\pi}^c_{kl}(\overline{c}_{ij}) + \underline{s}^*_{ij}\underline{\pi}^c_{kl}(\underline{s}_{ij}) + \overline{s}^*_{ij}\underline{\pi}^c_{kl}(\overline{s}_{ij}) \right) \right\}. \tag{42}$$

Similarly, using the dual variables from $\overline{P}^c_{kl}$, $\underline{P}^s_{kl}$ and $\overline{P}^s_{kl}$, we may try to tighten $\overline{c}^*_{kl}$, $\underline{s}^*_{kl}$ and $\overline{s}^*_{kl}$ further.

4 SOCP Based Spatial Branch-and-Cut Method

In Section 3, we proposed several convexification techniques for the minor (or equivalently, rank) constrained OPF problem including convex and concave envelopes and cutting planes.

In this section, we will show how they can be used in a relaxation scheme. In Section 4.1, we develop an algorithm which can be used stand-alone as a cutting plane approach to find dual bounds for the OPF problem. It can also be treated as the root node relaxation of the SOCP based spatial branch-and-cut algorithm proposed in Section 4.2. Finally, Section 4.3 presents the implementation details of the spatial branch-and-cut algorithm.

4.1 Root Node Relaxation

The computational experiments will be based on an SOCP relaxation of the OPF problem. Let $\text{SOCP}(\underline{c}, \overline{c}, \underline{s}, \overline{s}, \mathcal{H})$ denote this relaxation constructed by using the variable bounds $\underline{c}$, $\overline{c}$, $\underline{s}$ and $\overline{s}$, and a set of cutting planes of the form $\alpha_h^T \begin{bmatrix} c \\ s \end{bmatrix} \geq \beta_h$ from an index set $h \in \mathcal{H}$ obtained by solving separation problems. The full model is defined as follows:

$$
\begin{aligned}
&\text{SOCP}(\underline{c}, \overline{c}, \underline{s}, \overline{s}, \mathcal{H}) : && \\
&\min \ \sum_{i \in \mathcal{G}} C_i(p_i^g) && && \text{(43a)}\\
&\text{s.t. } p_i^g - p_i^d = G_{ii} c_{ii} + \sum_{j \in \delta(i)} [G_{ij} c_{ij} - B_{ij} s_{ij}] && i \in \mathcal{B}, && \text{(43b)}\\
&\qquad q_i^g - q_i^d = -B_{ii} c_{ii} + \sum_{j \in \delta(i)} [-B_{ij} c_{ij} - G_{ij} s_{ij}] && i \in \mathcal{B}, && \text{(43c)}\\
&\qquad \underline{V}_i^2 \leq c_{ii} \leq \overline{V}_i^2 && i \in \mathcal{B}, && \text{(43d)}\\
&\qquad c_{ij} = c_{ji}, \quad s_{ij} = -s_{ji} && (i,j) \in \mathcal{L} && \text{(43e)}\\
&\qquad [-G_{ij} c_{ii} + G_{ij} c_{ij} - B_{ij} s_{ij}]^2 + [B_{ij} c_{ii} - B_{ij} c_{ij} - G_{ij} s_{ij}]^2 \leq (S_{ij}^{\max})^2 && (i,j) \in \mathcal{L}, && \text{(43f)}\\
&\qquad c_{ij}^2 + s_{ij}^2 \leq c_{ii} c_{jj} && (i,j) \in \mathcal{L}, && \text{(43g)}\\
&\qquad \text{EC}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) && (i,j) \in \mathcal{L}, && \text{(43h)}\\
&\qquad \text{AT}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) && (i,j) \in \mathcal{L}, && \text{(43i)}\\
&\qquad \underline{\theta}_{ij} \leq \theta_i - \theta_j \leq \overline{\theta}_{ij} && (i,j) \in \mathcal{L}, && \text{(43j)}\\
&\qquad \underline{c}_{ij} \leq c_{ij} \leq \overline{c}_{ij}, \quad \underline{s}_{ij} \leq s_{ij} \leq \overline{s}_{ij} && (i,j) \in \mathcal{L}, && \text{(43k)}\\
&\qquad \alpha_h^T \begin{bmatrix} c \\ s \end{bmatrix} \geq \beta_h && h \in \mathcal{H}, && \text{(43l)}\\
&\qquad \text{(1e)} - \text{(1f)}. && &&
\end{aligned}
$$

Our approach heavily depends on tightening the variable bounds and enriching the set of cutting planes so that SOCP relaxation gets tightened. The main steps of this root node relaxation algorithm is summarized in Algorithm 1.

Algorithm 1 Root node relaxation.

```
$LB = -\infty$, $UB = \infty$, $\mathcal{C} = \emptyset$, $\mathcal{H} = \emptyset$, $t = 0$
Use a local solver to find a feasible solution and update $UB$.
Obtain a cycle basis $C_b$ and set $\mathcal{C} = C_b$.
Solve bound tightening problems $\underline{P}^c_{kl}$, $\overline{P}^c_{kl}$, $\underline{P}^s_{kl}$ and $\overline{P}^s_{kl}$ for all $(k,l) \in \mathcal{L}$ with $r_1$ and apply dual improvement.
while $t < T$ and $LB < (1-\epsilon)UB$ do
    if $t < T_C$ and $|\mathcal{B}| \leq B_{\max}$ then
        Enlarge $\mathcal{C}$ by adding more cycles.
    end if
    Solve bound tightening problems $\underline{P}^c_{kl}$, $\overline{P}^c_{kl}$, $\underline{P}^s_{kl}$ and $\overline{P}^s_{kl}$ for all $(k,l) \in \mathcal{L}$ with $r_2$ and apply dual improvement.
    Solve SOCP$(\underline{c}, \overline{c}, \underline{s}, \overline{s}, \mathcal{H})$ to obtain a solution $(c^*, s^*)$ and update $LB$.
    Solve SEP$(\mathcal{S}_C, c^*, s^*)$ and/or SEP$(\mathcal{M}^D_C, c^*, s^*)$ to obtain a set of cutting planes $H_t$ for all $C \in \mathcal{C}$.
    Update $\mathcal{H} = \mathcal{H} \cup H_t$.
    Set $t = t + 1$.
end while
```

For small ($|\mathcal{B}| \leq B_{\max}$) and challenging instances, we generate new cycles for T_C many rounds using every pair of distinct cycles in the current set $\mathcal{C}$ which share at least one common line.

4.2 Spatial Branch-and-Cut Algorithm

Algorithm 1 is quite successful in proving strong dual bounds for many instances from the NESTA archive as the numerical experiments in Section 5.2.1 show. Nevertheless, the optimality gap may be more than an acceptable threshold for some of the more challenging instances, for which we propose an SOCP based spatial branch-and-cut algorithm. The main steps can be seen in Algorithm 2.

Our approach is built on the following principles:

(i) Branching: In our approach, we decide a transmission line (i,j) and branch on either c_{ij} and s_{ij}. This branching rule allows us to update convex approximations to both EC$_{ij}$ and AT$_{ij}$. We pick the line to be branched on node L of the branch-and-bound tree as follows:

$$line_L = \max_{(i,j)\in\mathcal{L}} \left| \theta_j - \theta_i - \arctan\left(\frac{s_{ij}}{c_{ij}}\right) \right|. \qquad (44)$$

Then, among c_{ij} and s_{ij}, we choose the variable whose smallest distance to the boundary is the largest. In particular, if $\min\{c_{ij} - \underline{c}_{ij}, \overline{c}_{ij} - c_{ij}\} \geq \min\{s_{ij} - \underline{s}_{ij}, \overline{s}_{ij} - s_{ij}\}$, then

c_{ij} is chosen; otherwise, s_{ij} is chosen. Finally, we use bisection-branching to partition the space [46].

(ii) Local bound tightening: Since branching on a variable c_{ij} or s_{ij} reduces the variable range, other variables which correspond to the nearby lines to the branched line can be improved as well. Therefore, we solve the bound tightening problems for such lines in our algorithm.

(iii) Node selection: Since our aim is to reduce the duality gap on the problem, we choose the node with the smallest node relaxation value and carry out the branching.

(iv) Cutting plane generation: We keep on generating cutting planes to separate relaxation solutions. To be computationally efficient, we only solve the separating problems for the cycles at hand which contains the branched line.

Algorithm 2 Spatial branch-and-cut.

```
Let $LB$, $UB$, $\mathcal{C}$ and $\mathcal{H}$ be computed from Algorithm 1.
Set $list = \{root\}$.
while $|list| > 0$ do
    $LB = \min_{l \in list} LB_l$ and $L = \operatorname{argmin}_{l \in list} LB_l$.
    $list = list \setminus \{L\}$.
    if $LB \geq (1-\epsilon)UB$ then
        STOP.
    end if
    Solve bound tightening problems $\underline{P}^c_{kl}$, $\overline{P}^c_{kl}$, $\underline{P}^s_{kl}$ and $\overline{P}^s_{kl}$ for all $(k,l)$ near $line_{parent(L)}$
    with $r_2$ and apply dual improvement.
    Solve SEP($\mathcal{S}_C, c^*, s^*$) and/or SEP($\mathcal{M}^D_C, c^*, s^*$) to obtain a set of cutting planes $H_t$ for
    all $C \in \mathcal{C}$ such that $line_{parent(L)} \in C$.
    Update $\mathcal{H}_L = \mathcal{H}_L \cup H_t$.
    Solve SOCP($\underline{c}, \overline{c}, \underline{s}, \overline{s}, \mathcal{H}$) to obtain a solution $(c^*, s^*)$ and update $LB$.
    Decide on a transmission line $line_L$ to branch on.
    Obtain two children $L_1$ and $L_2$ by updating variable bounds, EC and AT.
    $list = list \cup \{L_1, L_2\}$.
end while
```

4.3 Implementation

In Algorithm 2, SOCP relaxation of each node can be constructed from scratch given the following four pieces of information:

(i) variable bounds,

(ii) its parent's relaxation solution,

(iii) transmission line branched on, and

(iv) the valid inequalities of its parent.

Therefore, a direct implementation can be obtained by explicit tree handling as long as the parent inherits this set of information.

This implementation is a reasonable attempt since, unlike LPs, there is no efficient warm-start availability for SOCPs. There are also some disadvantages: For instance, the proposed implementation requires the construction of each problem from scratch and explicit tree handling. Although the data needed to be stored at each node is limited, there may be some issues for large problems.

In this implementation, the overhead is the solution of SOCPs at each node of the branch-and-bound tree. We prefer to use MOSEK in this implementation since it is an efficient conic interior point solver.

Finally, bound tightening and separation problems are parallelized to reduce the total computational time.

5 Computational Experiments

In this section, we present the results of our extensive computational experiments from NESTA 0.3.0 archive [11] with Typical, Congested and Small Angle Operating Conditions. We are particularly interested in this set of instances due to their difficulty level, as explained below. Our main code is written in the C# language with Visual Studio 2010 as the compiler. For comparison purposes, we use OPF Solver [30] to solve the SDP relaxation of the OPF problem. This MATLAB package exploits sparsity of the power networks to efficiently solve large-scale SDP problems [31, 32]. We modified the code slightly to incorporate phase angle difference constraints. For all experiments, we used a 64-bit computer with Intel Core i5 CPU 2.50GHz processor and 16 GB RAM. Time is measured in seconds, unless otherwise stated. Conic interior point solver MOSEK 8 [40] is used to solve LPs, SOCPs and SDPs in our main algorithms. OPF Solver is run with MOSEK and SDPT3.

5.1 Methods

We run our algorithms with different settings as to cutting plane generation procedures:

- McCormick Separation ($\mathsf{SEP(M)}$): We only separate the point from $\mathcal{M}_C^D$ defined in (24).

- SDP Separation ($\mathsf{SEP(S)}$): We only separate the point from $\mathcal{S}_C$ defined in (35).
- SDP + McCormick Separation ($\mathsf{SEP(M,S)}$): We separate the point from $\mathcal{M}_C^D$ and $\mathcal{S}_C$.

We use a fixed cycle basis to generate cutting planes for most instances. For small ($|\mathcal{B}| \leq B_{\max} = 118$) and difficult instances, we enlarge the set of cycles to obtain more cuts. After initial calibration, we decide to set the number of bound tightening rounds T to 5, the number of cycle addition rounds T_C to 1, the initial radius r_1 to 2, the later radius r_2 to 4, and the optimality tolerance ϵ to 10^{-3}. We also employ coefficient rounding for SDP cuts to improve numerical stability.

The OPF Solver code is modified to incorporate the phase angle bounds in NESTA instances by adding the following constraints:

$$\Im(X_{ij}) - \tan\overline{\theta}_{ij}\Re(X_{ij}) \leq 0 \quad \text{and} \quad \Im(X_{ij}) - \tan\underline{\theta}_{ij}\Re(X_{ij}) \geq 0. \tag{45}$$

We run the OPF Solver with two solvers:

- MOSEK
- SDPT3

We also compare the three approaches proposed in this paper to our previous paper [27], which utilizes strong SOCP relaxations. For Typical and Congested Operating Conditions, we use a simplified version of $\mathsf{SEP(S)}$ and for Small Angle Operating Condition, we again use a simpler version of $\mathsf{SEP(M)}$, which proved to be the best setting in that paper.

5.2 Comparison to the SOCP and SDP Relaxation

We compared the relaxation values obtained from our approach to the plain SOCP, strong SOCP, and SDP relaxations in terms of the optimality gap, which is calculated as follows: %gap $= 100 \times \frac{z^{\text{UB}} - z^{\text{LB}}}{z^{\text{UB}}}$. Here, z^{LB} is the optimal objective cost of a relaxation and z^{UB} is the objective cost of a feasible solution obtained by MATPOWER [53].

5.2.1 Root Node Relaxation

In this section, we present the computational results for Typical, Congested, and Small Angle Operating Condition instances in Tables 1-3, respectively. Also, we provide a scatter plot Figure 2, which visualizes the average percentage optimality gap and computational times.

	SOCP		SDP		Best of [27]		SEP(M)		SEP(S)		SEP(M, S)	
case	%gap	time (s)	%gap	time (s)	%gap	time (s)	%gap	time (s)	%gap	time (s)	%gap	time (s)
3lmbd	1.32	0.06	0.39	1.00	0.43	0.14	0.10	1.09	0.09	1.12	0.10	0.95
4gs	0.00	0.05	0.00	0.98	0.00	0.08	0.00	0.11	0.00	0.05	0.00	0.03
5pjm	14.54	0.09	5.22	1.03	6.22	0.17	5.63	1.59	3.68	2.04	2.11	3.26
6ww	0.63	0.02	0.00	1.30	0.00	0.44	0.02	0.51	0.01	0.75	0.01	1.08
9wscc	0.00	0.06	0.00	1.03	0.00	0.08	0.00	0.09	0.00	0.06	0.00	0.09
14ieee	0.11	0.05	0.00	1.36	0.00	0.53	0.03	0.80	0.00	1.31	0.00	2.70
29edin	0.14	0.11	0.00	2.98	0.00	1.81	0.06	8.67	0.01	21.06	0.01	33.99
30as	0.06	0.03	0.00	2.49	0.00	0.90	0.06	0.06	0.06	0.14	0.06	0.11
30fsr	0.39	0.06	0.00	1.93	0.03	0.92	0.10	9.61	0.07	8.58	0.07	14.49
30ieee	15.65	0.03	0.00	1.50	0.00	0.92	0.09	10.55	0.03	10.05	0.03	14.55
39epri	0.05	0.09	0.01	2.21	0.01	0.67	0.05	0.14	0.05	0.09	0.05	0.25
57ieee	0.06	0.06	0.00	3.05	0.00	1.73	0.06	0.11	0.06	0.25	0.06	0.22
118ieee	2.10	0.17	0.07	6.31	0.25	4.67	0.42	115.50	0.14	228.60	0.14	355.50
162ieee	4.19	0.17	1.12	17.93	3.50	9.19	2.14	373.38	1.56	666.08	1.57	948.30
189edin	0.22	0.28	0.07	6.59	0.08	2.04	0.28	48.30	0.10	87.17	0.04	63.15
300ieee	1.19	0.39	0.08	16.36	0.30	9.41	0.23	321.62	0.09	345.69	0.09	520.50
2383wp	1.68	6.37	0.37	840.31	1.56	74.25	1.48	33.59	1.36	46.10	1.16	163.27
2736sp	1.57	8.21	0.00	1265.13	1.42	91.07	1.82	49.62	1.07	90.00	0.87	165.61
2737sop	6.54	4.68	0.00	1228.51	1.57	87.30	5.35	51.83	1.15	185.08	1.15	348.35
2746wop	13.61	3.98	0.00	1329.97	1.50	91.39	2.40	66.19	2.57	140.07	2.57	273.07
2746wp	2.48	6.44	0.00	1383.23	1.54	91.84	2.61	41.28	1.08	61.44	1.04	107.09
Average	3.17	1.50	0.35	291.20	0.88	22.36	1.09	54.03	0.63	90.27	0.53	143.65

Table 1: Root node relaxation results for Typical Operating Condition instances.

	SOCP		SDP		Best of [27]		SEP(M)		SEP(S)		SEP(M, S)	
case	%gap	time (s)	%gap	time (s)	%gap	time (s)	%gap	time (s)	%gap	time (s)	%gap	time (s)
3lmbd	3.30	0.05	1.26	0.93	1.31	0.14	0.81	0.62	0.78	1.68	0.81	1.05
4gs	0.65	0.02	0.00	0.88	0.00	0.09	0.05	0.78	0.03	0.47	0.03	0.55
5pjm	0.45	0.05	0.45*	1.03	0.00	0.20	0.09	0.41	0.05	0.76	0.05	0.81
6ww	13.33	0.03	0.00	1.04	0.00	0.48	0.00	1.18	0.00	1.87	0.00	3.39
9wscc	0.00	0.05	0.00	0.93	0.00	0.11	0.00	0.08	0.00	0.08	0.00	0.06
14ieee	1.35	0.06	0.00	1.06	0.00	0.58	0.14	8.67	0.03	7.16	0.04	13.18
29edin	0.44	0.11	0.44*	2.90	0.03	1.96	0.08	66.27	0.04	93.01	0.04	136.83
30as	4.76	0.06	0.00	1.96	1.72	1.00	0.11	25.06	0.08	37.44	0.09	62.09
30fsr	45.97	0.06	11.06	1.90	40.22	1.00	9.91	27.92	5.13	50.09	5.15	90.56
30ieee	0.99	0.05	0.00	2.32	0.08	0.98	0.18	24.04	0.06	37.84	0.06	60.03
39epri	2.99	0.05	0.00	2.55	0.00	0.73	0.09	9.44	0.01	17.14	0.01	26.33
57ieee	0.21	0.06	0.08	2.69	0.13	1.82	0.20	13.95	0.06	84.38	0.06	125.44
118ieee	44.19	0.14	31.53	7.47	39.09	5.07	14.38	285.36	7.91	517.63	7.83	911.90
162ieee	1.52	0.23	1.00	21.43	1.20	9.88	1.17	617.04	1.03	1393.66	1.03	2007.66
189edin	5.88	0.22	0.05	6.53	3.82	2.28	1.12	212.12	0.89	444.09	0.91	592.86
300ieee	0.85	0.39	0.00	14.65	0.15	9.94	0.22	571.77	0.10	735.95	0.10	1048.07
2383wp	0.89	2.07	0.10	857.87	0.00	58.46	0.99	69.62	0.40	208.76	0.40	592.17
2736sp	2.13	2.73	0.07	1439.24	0.72	66.89	1.57	65.91	1.32	109.79	1.32	308.40
2737sop	1.08	2.93	0.01	1203.03	0.31	63.74	1.10	40.59	0.65	268.75	0.65	636.37
2746wop	0.52	2.70	0.00	1413.02	0.00	67.87	0.51	42.65	0.42	118.05	0.42	296.78
2746wp	0.59	2.96	0.00	1457.02	0.00	74.26	0.64	43.13	0.21	174.71	0.71	153.60
Average	6.29	0.72	2.19	306.69	4.23	17.50	1.59	101.27	0.91	204.92	0.94	336.58

Table 2: Root node relaxation results for Congested Operating Condition instances. *: Numerical difficulties are encountered for the SDP relaxation, the resuts from SOCP relaxation is used instead.

	SOCP		SDP		Best of [27]		SEP(M)		SEP(S)		SEP(M, S)	
case	%gap	time (s)	%gap	time (s)	%gap	time (s)	%gap	time (s)	%gap	time (s)	%gap	time (s)
3lmbd	4.28	0.05	2.06	1.15	1.52	0.10	0.11	1.02	0.28	0.64	0.09	1.29
4gs	4.90	0.02	0.05	1.01	0.03	0.11	0.01	0.20	0.01	0.22	0.01	0.66
5pjm	3.61	0.03	0.00	1.11	0.39	0.14	0.07	0.34	0.08	0.36	0.07	0.94
6ww	0.80	0.02	0.00	1.34	0.02	0.27	0.00	0.51	0.00	0.66	0.00	1.53
9wscc	1.50	0.05	0.00	1.02	0.43	0.20	0.01	0.44	0.01	0.45	0.01	1.14
14ieee	0.07	0.03	0.00	1.39	0.06	0.44	0.06	0.03	0.06	0.06	0.06	0.16
29edin	34.47	0.09	28.44	2.51	21.92	4.79	0.90	63.05	0.80	168.42	0.70	325.68
30as	9.16	0.08	0.47	1.82	2.47	1.26	0.14	11.64	0.09	20.67	0.09	38.85
30fsr	0.62	0.09	0.07	2.19	0.29	1.12	0.13	10.44	0.09	14.01	0.09	26.57
30ieee	5.87	0.08	0.00	2.41	2.04	1.01	0.08	5.30	0.02	13.78	0.02	26.78
39epri	0.11	0.03	0.09	2.25	0.09	1.02	0.03	2.67	0.02	6.01	0.02	11.54
57ieee	0.11	0.08	0.02	2.92	0.10	1.83	0.08	6.07	0.07	19.81	0.07	36.75
118ieee	12.88	0.17	7.55	6.03	7.41	6.20	4.57	122.62	3.35	375.91	3.35	748.42
162ieee	7.06	0.14	3.56	20.66	5.86	14.30	4.12	418.86	3.77	1123.51	3.76	1741.94
189edin	2.36	0.25	1.20	6.70	2.33	5.44	4.04	56.13	1.04	346.65	1.41	315.67
300ieee	1.27	0.39	0.13	15.61	0.71	16.75	0.23	345.04	0.11	773.31	0.10	1226.36
2383wp	5.46	5.26	1.30	850.43	3.67	554.44	3.56	86.25	3.08	142.53	3.08	550.16
2736sp	3.47	6.49	0.69	1415.42	2.02	676.52	2.77	57.89	3.07	113.58	3.81	448.04
2737sop	3.63	7.27	1.00	1298.22	3.55	694.33	6.82	38.16	4.56	71.45	5.37	153.54
2746wop	4.32	7.10	1.20	1448.25	3.94	772.76	6.36	40.00	3.98	67.80	4.56	150.09
2746wp	3.76	6.38	0.43	1327.80	2.73	811.51	3.50	42.57	4.20	129.86	2.74	155.73
Average	5.22	1.62	2.30	305.25	2.93	169.74	1.79	62.34	1.37	161.41	1.40	283.90

Table 3: Root node relaxation results for Small Angle Operating Condition instances.

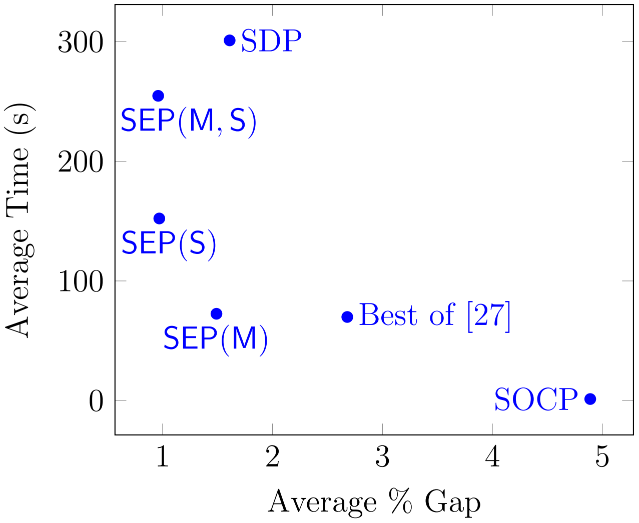

Figure 2: Scatter plot for root node relaxation results.

Tables 1-3 summarize the results of our three methods applied only to the root node relaxation. We see that SEP(M) approach is both the most efficient and the weakest in terms of the optimality gap proven among the three methods whereas SEP(M, S) approach takes the longest computational times but provides the strongest relaxations overall. The accuracy of SEP(S) approach is very close to SEP(M, S) with lower computational cost.

Figure 2 compares our three methods against SDP Relaxation solved using OPF Solver with SDPT3 chosen as the solver. Although MOSEK is much faster with 172.66 seconds on the average than SDPT3 with 301.05 seconds, SDPT3 provides more accurate solutions with 1.61% optimality gap on the average while the optimality gap for MOSEK is 1.98%. Therefore, we will base our comparisons with SDPT3 results. We can easily see that all our methods, SEP(M), SEP(S) and SEP(M, S), dominate OPF Solver. In particular, SEP(S) and SEP(M, S) methods are about two times more accurate than OPF Solver in terms of the average optimality gap proven. The computational costs of SEP(S) and SEP(M, S) are about 50% and 15% less than OPF Solver. We would like to emphasize the success of purely LP and SOCP based method SEP(M) here. Although it provides the weakest relaxation among our three approaches, it is still stronger than a purely SDP based approach in about 25% of the computational time. In general, we should note that our approaches are much faster on large instances, more accurate on hard instances, and comparable to the SDP relaxation on small or easy instances.

We also compare the three approaches proposed in this paper to our previous paper [27]. The approaches in our previous paper are typically faster than the ones proposed in the current paper, however, the optimality gaps are about 2-3 times worse. This shows the significant improvement from our previous work.

5.2.2 Effect of Branching

In this section, we present the detailed branch-and-cut results for the instances which are not solved within the optimality threshold at the root node relaxation in Table 4. We run the branch-and-cut algorithm with a budget of 15 and 30 minutes. The average results are also presented in Table 5.

Overall, the branching reduces the average optimality gaps from 1.49, 0.97 and 0.96 at the root node to 1.21, 0.78 and 0.87 after 15 minutes for the methods $\mathsf{SEP(M)}$, $\mathsf{SEP(S)}$ and $\mathsf{SEP(M,S)}$, respectively. There are quite significant gap closure thanks to branching for 5pjm (typical OC instance), 30fsr (congested OC) and 118ieee (congested and small angle OC) instances. We should point out that it is possible to process more nodes with $\mathsf{SEP(M)}$ method, which helps to reduce the optimality gaps the most among our three methods, however, it is still the weakest relaxation approach. Also, we can see that $\mathsf{SEP(S)}$ method is able to provide lower average optimality gap than $\mathsf{SEP(M,S)}$ at the end since it processes more nodes.

After 30 minutes, the percentage optimality gaps reduce further to 1.12, 0.71 and 0.82 for the methods $\mathsf{SEP(M)}$, $\mathsf{SEP(S)}$ and $\mathsf{SEP(M,S)}$, respectively.

A final comparison of different methods is presented in Figure 3, which can be interpreted as a *cumulative distribution function*. In this figure, we record the fraction of instances solved up to a given percentage optimality gap. Therefore, a method whose corresponding curve is below the others is dominated. By construction, the plain SOCP relaxation approach is dominated since it is the weakest relaxation considered. Strong SOCP relaxations from our previous paper [27] improves the plain SOCP relaxation considerably but it is not very competitive against the SDP relaxation, especially for easier instances. However, the proposed $\mathsf{SEP(S)}$ approach at the root node and after branching is more successful than the SDP relaxation, especially for the more difficult instances.

		SEP(M)				SEP(S)				SEP(M, S)			
OC	Case	15 m	30 m	time (s)	node	15 m	30 m	time (s)	node	15 m	30 m	time (s)	node
TYPICAL (TYP)	3lmbd	0.09	0.09	1.62	3	0.09	0.09	1.12	1	0.09	0.09	0.95	1
	5pjm	0.10	0.10	124.46	270	0.10	0.10	80.23	136	0.10	0.10	108.39	129
	30fsr	0.10	0.10	12.18	3								
	118ieee	0.26	0.24	1801.44	364	0.09	0.09	324.33	13	0.10	0.10	502.47	13
	162ieee	1.96	1.46	1811.55	139	1.56	1.33	1810.54	56	1.46	1.46	1837.44	21
	189edin	0.14	0.10	660.44	179								
	300ieee	0.21	0.19	1807.97	287								
	2383wp	1.33	1.28	1811.84	43	1.16	1.08	1826.53	27	0.92	0.92	1881.99	24
	2736sp	0.93	0.93	1810.29	50	0.87	0.87	1854.32	30	0.76	0.63	1869.73	23
	2737sop	5.35	5.35	1820.06	44	1.15	1.15	1800.99	27	1.15	1.15	1858.50	19
	2746wop	2.40	2.40	1828.43	46	2.57	2.18	1819.52	29	2.57	2.57	1819.18	21
	2746wp	1.61	1.61	1828.00	38	1.04	1.04	1812.67	26	2.54	2.54	1852.72	22
CONGESTED (API)	3lmbd	0.02	0.02	2.09	5	0.02	0.02	3.99	5	0.02	0.02	3.34	5
	14ieee	0.09	0.09	11.86	5								
	30as	0.06	0.06	28.84	3								
	30fsr	4.13	1.79	1801.24	603	0.73	0.35	1803.06	393	1.31	0.83	1802.18	220
	30ieee	0.09	0.09	29.33	5								
	57ieee	0.10	0.10	226.32	57								
	118ieee	13.57	12.14	1803.17	115	7.83	6.17	1809.53	49	7.83	7.83	1834.74	23
	162ieee	1.17	1.16	1831.99	48	1.03	1.03	1821.67	6	1.03	1.03	2007.68	0
	189edin*	0.12	0.12	472.70	42	0.89	0.89	543.14	9	0.12	0.12	663.19	5
	300ieee	0.22	0.18	1810.43	100								
	2383wp	0.98	0.98	1806.07	24	0.20	0.20	1801.00	20	0.40	0.40	1885.56	12
	2736sp	1.57	1.57	1802.19	36	1.26	1.21	1825.24	18	1.32	1.26	1869.59	12
	2737sop	1.10	1.10	1808.90	32	0.65	0.65	1809.25	18	0.65	0.65	1835.00	10
	2746wop	0.51	0.51	1852.43	31	0.42	0.41	1852.74	25	0.42	0.36	1931.52	16
	2746wp	0.64	0.64	1847.24	27	0.20	0.20	1811.35	19	0.71	0.43	1814.47	12
SMALL ANGLE (SAD)	3lmbd	0.03	0.03	1.63	3	0.03	0.03	1.19	3	0.03	0.03	1.29	1
	29edin	0.85	0.84	1800.96	302	0.70	0.70	1819.15	102	0.69	0.67	1837.01	52
	30as	0.08	0.08	14.71	5								
	30fsr	0.08	0.08	18.55	9								
	118ieee	4.02	3.98	1801.68	216	3.07	2.43	1811.48	71	3.35	3.07	1804.74	26
	162ieee	4.12	3.84	1813.33	118	3.76	3.76	1820.50	15				
	189edin*	4.04	4.04	185.41	34	1.14	1.14	1800.41	139	2.70	1.06	1814.79	96
	300ieee	0.21	0.16	1805.98	261	0.10	0.10	797.41	3				
	2383wp	3.56	3.56	1813.15	66	3.03	2.89	1867.73	18	3.08	2.83	1805.81	12
	2736sp	2.77	2.77	1810.32	62	2.98	2.50	1827.66	31	3.81	3.81	1879.16	16
	2737sop	6.82	6.82	1808.89	71	3.57	3.57	1821.21	22	2.81	2.81	1805.93	15
	2746wop	6.36	5.61	1836.42	53	4.56	4.56	1899.28	26	5.41	5.41	1850.18	19
	2746wp	3.50	3.50	1806.90	53	2.74	2.74	1868.46	22	4.49	4.49	1860.37	15

Table 4: Branch-and-cut results for the instances which are not solved at the root node up to the optimality threshold $\epsilon = 10^{-3}$. Percentage optimality gaps are reported after 15 and 30 minutes of branching (root node computation time is included). OC: Operating Condition.

	SEP(M)				SEP(S)				SEP(M, S)			
OC	15 m	30 m	time (s)	node	15 m	30 m	time (s)	node	15 m	30 m	time (s)	node
TYP	0.71	0.68	730.44	70.24	0.43	0.40	562.17	17.00	0.48	0.48	589.65	13.57
API	1.18	0.99	819.67	54.24	0.65	0.55	766.53	27.29	0.68	0.64	815.43	15.52
SAD	1.75	1.70	787.31	60.05	1.24	1.18	829.07	22.00	1.46	1.36	852.02	12.57
ALL	1.21	1.12	779.14	61.51	0.78	0.71	719.26	22.10	0.87	0.82	752.37	13.89

Table 5: Averages of branch-and-cut results for all the instances. Percentage optimality gaps are reported after 15 and 30 minutes of branching (root node computation time is included).

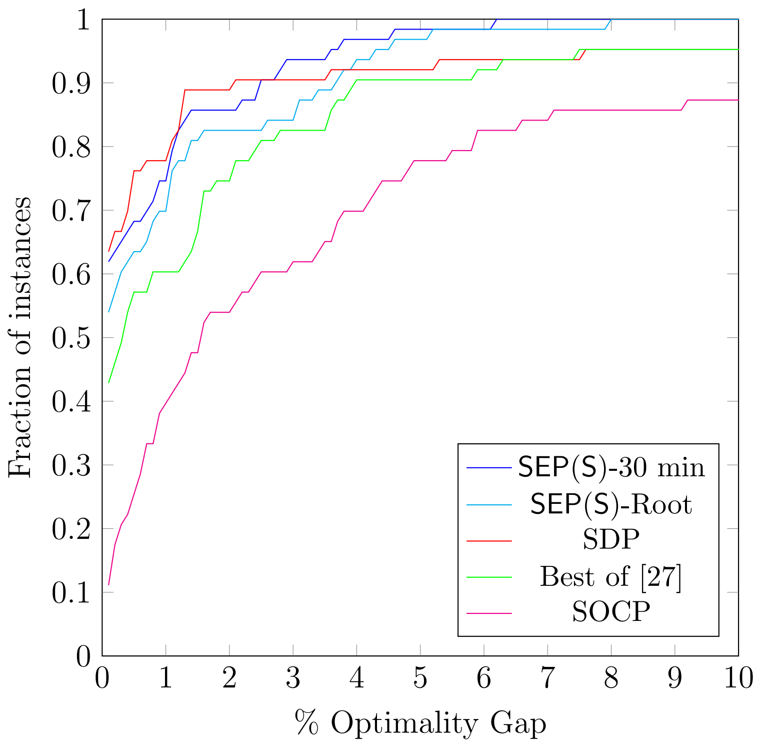

Figure 3: Fraction of instances solved up to a given percentage optimality gap for different methods.

6 Conclusion

In this paper, we proposed new approaches for obtaining globally optimal solutions of the AC OPF Problem. We first reformulated the AC OPF problem as a minor constrained problem, and then proposed several convexification techniques for nonconvexities involving minor constraints using only second-order conic and linear relaxations. We improved the resulting SOCP relaxation via cutting planes and convex envelopes by incorporating bound tightening techniques. We proposed three methods with respect to the cutting plane procedure. Our methods are successful in proving global optimal solutions for many challenging OPF instances from the NESTA archive. Compared to the standard SDP relaxation, our approaches provide about 2 times smaller optimality gaps with only half of the average computation time. For the instances not solved, we propose to use a branch-and-cut scheme where the proposed SOCP relaxation serves as the root node relaxation. The strongest of our SOCP based branch-and-cut algorithms proves 0.71% optimality gap in 720 seconds on the average for the NESTA library.

As a future work, we would like to apply the methodology developed in this chapter to multi-period AC OPF and Optimal Transmission Switching Problems. Another possible line of research is to implement an LP based outer-approximation to SOCP based branch-and-cut method. This would lead to weaker relaxations than a pure SOCP based approach but can incorporate warm-start and it may be possible to process significantly more nodes in the same amount of time.

A KKT Points for Arctangent Envelopes

Let us rewrite the optimization problem in (29) as

$$\Delta\gamma = \max\{f(c,s) : (c,s) \in F\}, \tag{46}$$

where

$$f(c,s) := \arctan\left(\frac{s}{c}\right) - (\gamma + \alpha c + \beta s)$$

and

$$F := \{(c,s) : (c,s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}],\ c\tan\underline{\theta} \leq s \leq \tan\overline{\theta}\}.$$

Without loss of generality, let us assume that the relations $\arctan(\underline{s}/\underline{c}) \leq \underline{\theta}$ and $\overline{\theta} \leq \arctan(\overline{s}/\underline{c})$ hold between the variable bounds (otherwise, at least one of the bounds can be improved). Let us denote the optimal solution to problem (46) as (c^*, s^*). First, we claim that (c^*, s^*) is not in the interior of F. This is due to the fact that the Hessian of f at a point (c,s), which is given as

$$\frac{1}{(c^2+s^2)^2}\begin{bmatrix} 2cs & s^2-c^2 \\ s^2-c^2 & -2cs \end{bmatrix},$$

is an indefinite matrix. Therefore, an interior point of F will not satisfy the second-order necessary conditions of local optimality.

The above argument implies that (c^*, s^*) belongs to the boundary of F, which is the union of the following six line segments:

(i) $[(\underline{c}, \underline{c}\tan\underline{\theta}), (\underline{c}, \underline{c}\tan\overline{\theta})]$

(ii) $[(\underline{c}, \underline{c}\tan\overline{\theta}), (\overline{s}/\tan\overline{\theta}, \overline{s})]$

(iii) $[(\overline{s}/\tan\overline{\theta}, \overline{s}), (\overline{c}, \overline{s})]$

(iv) $[(\overline{c}, \overline{s}), (\overline{c}, \underline{s})]$

(v) $[(\overline{c}, \underline{s}), (\underline{s}/\tan\underline{\theta}, \underline{s})]$

(vi) $[(\underline{s}/\tan\underline{\theta}, \underline{s}), (\underline{c}, \underline{c}\tan\underline{\theta})]$

Note that the line segments (ii) and (vi) cannot contain (c^*, s^*) in their relative interior since the function f is linear along them. Hence, the problem reduces to four 1-dimensional optimization problems, which can be solved easily. The global optimal solution (c^*, s^*) is the one that gives the largest objective value among the KKT points calculated by solving those four 1-dimensional optimization problems.

B Algorithms for Extreme Point Calculations for Edge Cuts

Algorithm 3 Find extreme points of $\mathrm{conv}(\mathcal{K}_{ij}^{\geq})$ when $c_{jj} = \hat{c}_{jj}$, $c_{ij} = \hat{c}_{ij}$ and $s_{ij} = \hat{s}_{ij}$ are fixed to one of their bounds.

```
Compute $\phi = \frac{\hat{c}_{ij}^2 + \hat{s}_{ij}^2}{\hat{c}_{jj}}$ .
Let $E$ be the set of $c_{ii}$ coordinates of the extreme points.
$E = \begin{cases} \emptyset & \text{if } \phi < \underline{c}_{ii} \\ \{\phi, \overline{c}_{ii}\} & \text{if } \underline{c}_{ii} \leq \phi \leq \overline{c}_{ii} \\ \{\underline{c}_{ii}, \overline{c}_{ii}\} & \text{if } \phi > \overline{c}_{ii} \end{cases}$
```

Algorithm 4 Find extreme points of $\mathrm{conv}(\mathcal{K}_{ij}^{\geq})$ when $c_{ii} = \hat{c}_{ii}$, $c_{jj} = \hat{c}_{jj}$ and $s_{ij} = \hat{s}_{ij}$ are fixed to one of their bounds.

```
Compute $\phi = \hat{c}_{ii}\hat{c}_{jj} - \hat{s}_{ij}^2$.
Let $E$ be the set of $c_{ij}$ coordinates of the extreme points.
$E = \begin{cases} \{\underline{c}_{ij}, \overline{c}_{ij}\} & \text{if } \phi < 0 \\ \{\underline{c}_{ij}, \overline{c}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \sqrt{\phi} < \underline{c}_{ij} \\ \{\sqrt{\phi}, \overline{c}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \underline{c}_{ij} \leq \sqrt{\phi} \leq \overline{c}_{ij} \\ \emptyset & \text{if } \phi \geq 0 \text{ and } \sqrt{\phi} > \overline{c}_{ij} \end{cases}$
```

Algorithm 5 Find extreme points of $\text{conv}(\mathcal{K}_{ij}^{\geq})$ when $c_{ii} = \hat{c}_{ii}$, $c_{jj} = \hat{c}_{jj}$ and $c_{ij} = \hat{c}_{ij}$ are fixed to one of their bounds.

```
Compute $\phi = \hat{c}_{ii}\hat{c}_{jj} - \hat{c}_{ij}^2$.
Let $E$ be the set of $s_{ij}$ coordinates of the extreme points.
$E = \begin{cases} \{\underline{s}_{ij}, \overline{s}_{ij}\} & \text{if } \phi < 0 \\ \{\underline{s}_{ij}, \overline{s}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \overline{s}_{ij} < -\sqrt{\phi} \\ \{\underline{s}_{ij}, -\sqrt{\phi}\} & \text{if } \phi \geq 0 \text{ and } \underline{s}_{ij} \leq -\sqrt{\phi} \leq \overline{s}_{ij} \\ \{\underline{s}_{ij}, -\sqrt{\phi}, \sqrt{\phi}, \overline{s}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \underline{s}_{ij} \leq -\sqrt{\phi}, \sqrt{\phi} \geq \overline{s}_{ij} \\ \emptyset & \text{if } \phi \geq 0 \text{ and } -\sqrt{\phi} \leq \underline{s}_{ij}, \overline{s}_{ij} \leq \sqrt{\phi} \\ \{\sqrt{\phi}, \overline{s}_{ij}\} & \text{if } \phi \geq 0 \text{ and } -\sqrt{\phi} \leq \underline{s}_{ij} \leq \sqrt{\phi} \leq \overline{s}_{ij} \\ \{\underline{s}_{ij}, \overline{s}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \underline{s}_{ij} > \sqrt{\phi} \end{cases}$
```

C Proof of Theorem 3.2

Our proof approach is based on identifying the extreme points of $\mathcal{S}_a$. Let us start with a proposition.

Proposition C.1. *Let (x, y) be an extreme point of the set $\mathcal{S}_a$. Then, for a distinct pair of indices i and j*

(i) either x_i or y_j is at one of its bounds.

(ii) either x_i or y_i is at one of its bounds.

(iii) either x_j or y_j is at one of its bounds.

(iv) either x_j or y_i is at one of its bounds.

Proof. We only prove the first statement. The others can be proven using exactly the same reasoning.

Assume for a contradiction that $\underline{x}_i < x_i < \overline{x}_i$ and $\underline{y}_j < y_j < \overline{y}_j$. Consider the following cases:

Case 1: $y_i \neq 0$ and $x_j \neq 0$

Case 1a: $\frac{a_i y_i}{a_j x_j} > 0$

Let $\epsilon = \{x_i - \underline{x}_i, \overline{x}_i - x_i, \frac{a_j x_j}{a_i y_i}(y_i - \underline{y}_i), \frac{a_j x_j}{a_i y_i}(\overline{y}_i - y_i)\}$ and $\delta = \frac{a_i y_i}{a_j x_j}\epsilon$. Note that both ϵ and δ are positive. Now, construct $(x^+, y^-) = (x + \epsilon e_i, y - \delta e_j)$ and

$(x^-, y^+) = (x - \epsilon e_i, y + \delta e_j)$ where e_i is the i-th unit vector. Observe that both (x^+, y^-) and (x^-, y^+) belong to $\mathcal{S}_a$. Moreover, $(x, y) = \frac{1}{2}(x^+, y^-) + \frac{1}{2}(x^-, y^+)$. But, this is a contradiction to (x, y) being an extreme point of $\mathcal{S}_a$.

Case 1b: $\frac{a_i y_i}{a_j x_j} < 0$

Let $\epsilon = \{x_i - \underline{x}_i, \overline{x}_i - x_i, \frac{a_j x_j}{a_i y_i}(\underline{y}_i - y_i), \frac{a_j x_j}{a_i y_i}(y_i - \overline{y}_i)\}$ and $\delta = -\frac{a_i y_i}{a_j x_j}\epsilon$. Note that both ϵ and δ are positive. Now, construct $(x^+, y^+) = (x + \epsilon e_i, y + \delta e_j)$ and $(x^-, y^-) = (x - \epsilon e_i, y - \delta e_j)$. Observe that both (x^+, y^+) and (x^-, y^-) belong to $\mathcal{S}_a$. Moreover, $(x, y) = \frac{1}{2}(x^+, y^+) + \frac{1}{2}(x^-, y^-)$. But, this is a contradiction to (x, y) being an extreme point of $\mathcal{S}_a$.

Case 2: $y_i = 0$

Let $\epsilon = \{x_i - \underline{x}_i, \overline{x}_i - x_i\}$. Note that ϵ is positive. Now, construct $(x^+, y) = (x + \epsilon e_i, y)$ and $(x^-, y) = (x - \epsilon e_i, y)$. Observe that both (x^+, y) and (x^-, y) belong to $\mathcal{S}_a$. Moreover, $(x, y) = \frac{1}{2}(x^+, y) + \frac{1}{2}(x^-, y)$. But, this is a contradiction to (x, y) being an extreme point of $\mathcal{S}_a$.

Case 3: $x_j = 0$

Let $\delta = \{y_i - \underline{y}_i, \overline{y}_i - y_i\}$. Note that δ is positive. Now, construct $(x, y^+) = (x, y + \delta e_j)$ and $(x, y^-) = (x, y - \epsilon e_j)$. Observe that both (x, y^+) and (x, y^-) belong to $\mathcal{S}_a$. Moreover, $(x, y) = \frac{1}{2}(x, y^+) + \frac{1}{2}(x, y^-)$. But, this is a contradiction to (x, y) being an extreme point of $\mathcal{S}_a$.

□

Proposition C.1 implies the following corollary.

Corollary C.1. *Let (x, y) be an extreme point of the set $\mathcal{S}_a$. Then, either x_i and y_i or x_j and y_j are at their bounds for a distinct pair of indices i and j.*

Proof. Let $x_i = \hat{x}_i$ be a shorthand for "either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$". Then, Proposition C.1 implies that

$$\begin{aligned}&(x_i = \hat{x}_i \vee y_j = \hat{y}_j) \wedge (x_i = \hat{x}_i \vee x_j = \hat{x}_j) \wedge (y_i = \hat{y}_i \vee y_j = \hat{y}_j) \wedge (y_i = \hat{y}_i \vee x_j = \hat{x}_j)\\ =&(x_i = \hat{x}_i \wedge y_i = \hat{y}_i) \vee (x_j = \hat{x}_j \wedge y_j = \hat{y}_j),\end{aligned} \tag{47}$$

which is the desired conclusion. □

An immediate consequence of Corollary C.1 is the following characterization of extreme points of $\mathcal{S}_a$:

Corollary C.2. *All the extreme points of $\mathcal{S}_a$ are in one of the following sets:*

- $D_0 = \{(x,y) \in \mathcal{S}_a : (x_i, y_i) = (\hat{x}_i, \hat{y}_i) \ \forall i\}$
- $D_k = \{(x,y) \in \mathcal{S}_a : (x_i, y_i) = (\hat{x}_i, \hat{y}_i) \ i \neq k, x_k y_k = -\frac{1}{a_k} \sum_{i \neq k} a_i \hat{x}_i \hat{y}_i, \ x_k \in [\underline{x}_k, \overline{x}_k], \ y_k \in [\underline{y}_k, \overline{y}_k]\} \quad k = 1, \dots, N$

Note that D_0 is a collection of at most 4^N singletons whereas D_k is a collection of 4^{N-1} sets for each k. The projection of such a set onto (x_k, y_k) is of the following form

$$T_\alpha = \{(x,y) \in \mathbb{R}^2 : xy = \alpha, \quad x \in [\underline{x}, \overline{x}], \ y \in [\underline{y}, \overline{y}]\} \tag{48}$$

for some constant α.

Proposition C.2. *Set conv(T_α) is second-order cone representable for any value of α.*

There are several cases based on parameter values. In the most complicated case, we need $xy \geq \alpha$ (which is conic representable) and McCormick envelopes.

Now, we are ready to prove the main result.

Proof of Theorem 3.2. Since the convex hull of all the disjunctions are second-order cone representable (could be polyhedral or singleton depending on parameter values), conv($\mathcal{S}_a$) is also second-order cone representable. □

References

[1] X. Bai and H. Wei. Semi-definite programming-based method for security-constrained unit commitment with operational and optimal power flow constraints. *IET Generation Transmission & Distribution*, 3(2):182–197, 2009.

[2] X. Bai, H. Wei, K. Fujisawa, and Y. Wang. Semidefinite programming for optimal power flow problems. *Electric Power and Energy Systems*, 30:383–392, 2008.

[3] D. Bienstock and G. Munoz. On linear relaxations of OPF problems. *arXiv preprint arXiv:1411.1120*, 2014.

[4] Subhonmesh Bose, Dennice F Gayme, K Mani Chandy, and Steven H Low. Quadratically constrained quadratic programs on acyclic graphs with application to power flow. *arXiv preprint arXiv:1203.5599*, 2012.

[5] Subhonmesh Bose, Dennice F Gayme, Steven Low, and K Mani Chandy. Optimal power flow over tree networks. In *49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1342–1348, 2011.

[6] Waqquas A. Bukhsh, Andreas Grothey, Ken McKinnon, and Paul Trodden. Local solutions of optimal power flow. *IEEE Transactions on Power Systems*, 28(4):4780–4788, 2013.

[7] M. B. Cain, R. P. O'Neill, and A. Castillo. History of optimal power flow and formulations. http://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-1-history-formulation-testing.pdf, 2012.

[8] J. Carpentier. Contributions to the economic dispatch problem. *Bulletin Society Francaise Electriciens*, 8(3):431–447, 1962.

[9] C. Chen, A. Atamtürk, and S. S. Oren. Bound tightening for the alternating current optimal power flow problem. *IEEE Transactions on Power Systems*, PP(99):1–8, 2015.

[10] C. Chen, A. Atamtürk, and S. S. Oren. A spatial branch-and-cut algorithm for nonconvex QCQP with bounded complex variable. Bcol research report 15.04, University of California, Berkeley, 2015.

[11] C. Coffrin, D. Gordon, and P. Scott. NESTA, The NICTA energy system test case archive. *arXiv preprint arXiv:1411.0359*, 2014.

[12] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck. The QC relaxation: Theoretical and computational results on optimal power flow. *arXiv preprint arXiv:1502.07847*, 2015.

[13] C. Coffrin and P. Van Hentenryck. A linear-programming approximation of AC power flows. *INFORMS Journal on Computing*, 26(4):718–734, 2014.

[14] Santanu S. Dey and Akshay Gupte. Analysis of MILP techniques for the pooling problem. *Operations Research*, 63(2):412–427, 2015.

[15] S. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: A bibliographic survey I – formulations and deterministic methods. *Energy Systems*, 3(3):221–258, 2012.

[16] S. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: A bibliographic survey II – nondeterministic and hybrid methods. *Energy Systems*, 3(3):259–289, 2012.

[17] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion I: General framework. *SIAM Journal on Optimization*, 11(3):647–674, 2001.

[18] Akshay Gupte, Shabbir Ahmed, Santanu S. Dey, and Myun-Seok Cheon. Relaxations and discretizations for the pooling problem. *J. Global Optimization*, 67(3):631–669, 2017.

[19] Hassan Hijazi, Carleton Coffrin, and Pascal Van Hentenryck. Polynomial sdp cuts for optimal power flow. *arXiv preprint arXiv:1510.08107*, 2015.

[20] HL Hijazi, C Coffrin, and P Van Hentenryck. Convex quadratic relaxations of mixed-integer nonlinear programs in power systems. Technical report, NICTA, Canberra, ACT Australia, 2013.

[21] Richard J Hillestad and Stephen E Jacobsen. Linear programs with an additional reverse convex constraint. *Applied Mathematics and Optimization*, 6(1):257–269, 1980.

[22] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2013.

[23] R. A. Jabr. Exploiting sparsity in SDP relaxations of the OPF problem. *IEEE Transactions on Power Systems*, 27(2):1138–1139, 2012.

[24] R. A. Jabr, A. H. Coonick, and B. J. Cory. A primal-dual interior point method for optimal power flow dispatching. *IEEE Transactions on Power Systems*, 17(3):654–662, 2002.

[25] Rabih A. Jabr. Radial distribution load flow using conic programming. *IEEE Transactions on Power Systems*, 21(3):1458–1459, 2006.

[26] Rabih A. Jabr. Optimal power flow using an extended conic quadratic formulation. *IEEE Transactions on Power Systems*, 23(3):1000–1008, 2008.

[27] B. Kocuk, S. S. Dey, and X. A. Sun. Strong SOCP relaxations for the optimal power flow problem. *Operations Research*, 64(6):1176–1196, 2016.

[28] B. Kocuk, S.S. Dey, and X.A. Sun. Inexactness of SDP relaxation and valid inequalities for optimal power flow. *IEEE Transactions on Power Systems*, 31(1):642–651, Jan 2016.

[29] Javad Lavaei and Steven H. Low. Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems*, 27(1):92–107, 2012.

[30] R. Madani, M. Ashraphijuo, and J. Lavaei. OPF Solver Guide, 2014. `http://ieor.berkeley.edu/~lavaei/Software.html`.

[31] R. Madani, M. Ashraphijuo, and J. Lavaei. Promises of conic relaxation for contingency-constrained optimal power flow problem. Allerton, 2014.

[32] R. Madani, S. Sojoudi, and J. Lavaei. Convex relaxation for optimal power flow problem: Mesh networks. *IEEE Transactions on Power Systems*, 30(1):199–211, Jan 2015.

[33] Ramtin Madani, Somayeh Sojoudi, and Javad Lavaei. Convex relaxation for optimal power flow problem: Mesh networks. In *ACSSC*, pages 1375–1382, 2013.

[34] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

[35] Ruth Misener, Jeffrey P. Thompson, and Christodoulos A. Floudas. Apogee: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Computers & Chemical Engineering*, 35(5):876 – 892, 2011.

[36] D. K. Molzahn, J. T. Holzer, B. C. Lesieutre, and C. L DeMarco. Implementation of a large-scale optimal power flow solver based on semidefinite programming. *IEEE Transactions on Power Systems*, 28(4):3987–3998, 2013.

[37] D.K. Molzahn and I.A. Hiskens. Sparsity-exploiting moment-based relaxations of the optimal power flow problem. *To appear in IEEE Transactions on Power Systems*, 2015.

[38] J. A. Momoh, M. E. El-Hawary, and R. Adapa. A review of selected optimal power flow literature to 1993 part I: Nonlinear and quadratic programming approaches. *IEEE Transactions on Power Systems*, 14(1):96–104, 1999.

[39] J. A. Momoh, M. E. El-Hawary, and R. Adapa. A review of selected optimal power flow literature to 1993 part II: Newton, linear programming and interior point methods. *IEEE Transactions on Power Systems*, 14(1):105–111, 1999.

[40] MOSEK. *MOSEK Modeling Manual.* MOSEK ApS, 2013.

[41] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota. Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results. *Mathematical Programming*, 95(2):303–327, 2003.

[42] Y. Nesterov, H. Wolkowicz, and Y. Ye. *Handbook of semidefinite programming*, volume 27 of *Internat. Ser. Oper. Res. Management Sci.*, chapter Semidefinite programming relaxations of nonconvex quadratic optimization, pages 361–419. Kluwer Acad. Publ., 2000.

[43] Dzung T. Phan. Lagrangian duality and branch-and-bound algorithms for optimal power flow. *Operations Research*, 60(2):275–285, 2012.

[44] Somayeh Sojoudi and Javad Lavaei. Physics of power networks makes hard optimization problems easy to solve. In *IEEE Power and Energy Society General Meeting*, pages 1–8, 2012.

[45] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

[46] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.

[47] Mohit Tawarmalani and Jean-Philippe P. Richard. Decomposition techniques in convexification of inequalities. Technical report, 2013. Working paper.

[48] J.A. Taylor. *Convex Optimization of Power Systems*. Cambridge University Press, Cambridge, UK, 2015.

[49] G. L. Torres and V. H. Quintana. An interior-point method for nonlinear optimal power flow using voltage rectangular coordinates. *IEEE Transactions on Power Systems*, 13(4):1211–1218, 1998.

[50] H. Wang, C. E. Murillo-Sánchez, R. D. Zimmerman, and R. J. Thomas. On computational issues of market based optimal power flow. *IEEE Transactions on Power Systems*, 22(3):1185–1193, 2007.

[51] Y. Wu, A. S. Debs, and R. E. Marsten. A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows. *IEEE Transactions on Power Systems*, 9(2):876–883, 1994.

[52] Baosen Zhang and D. Tse. Geometry of feasible injection region of power networks. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 1508–1515, Sept 2011.

[53] R.D. Zimmerman, C.E. Murillo-Sanchez, and R.J. Thomas. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, Feb 2011.