Rank-1 Positive Semidefinite Matrix-Based Nonlinear Programming Formulation for AC OPF

Minyue Ma[1]* and Lingling Fan[2]† ‡

[1]*Department of Electrical Engineering, University of South Florida, Tampa FL USA 33620. Tel.:1(813)361-6659*
[2]*Department of Electrical Engineering, University of South Florida, Tampa FL USA 33620.Tel.:1(813)974-2031*

Abstract
Semidefinite programming (SDP) relaxation offers a tight relaxation to nonconvex alternating current optimal power flow (AC OPF) problems. When the solution obtained from SDP relaxation of AC OPF is a rank-1 positive semidefinite (PSD) matrix, this solution is exact to the original problem. Research efforts have been devoted to find a rank-1 PSD matrix. In this paper, a nonlinear programming formulation with the PSD matrix as the decision variable is proposed. The rank-1 PSD matrix constraint is equivalent to all 2×2 minors of the PSD matrix being zero. The main challenge of the proposed formulation is the large number of the quadratic equality constraints. For a system of N buses, there are $C_N^2 C_N^2$ minor related constraints (For a 10-node system, this number is 2025). Graph decomposition based approach is then implemented in this research to decompose a power grid into radial lines and three-node cycles. Enforcing the related sub-matrices PSD and rank-1 guarantees a full PSD rank-1 matrix. Case

*E-mail: minyuema@mail.usf.edu
†Corresponding author.
‡E-mail: linglingfan@usf.edu

study results demonstrate that the proposed formulation can provide the similar quality results with the original AC OPF formulation.

Keywords: Nonlinear programming, Convex relaxation, Exactness conditions, AC optimal power flow.

1 Introduction

Alternating current optimal power flow (AC OPF) is a classic optimization problem in power systems[1]. The objective is to minimize generation cost and/or power loss. Constraints are related to power grid physical characteristics (e.g., power flow equations), component limits (e.g., generator capacity limits, transmission line limits) as well as network operation limits (e.g., voltage limits). Depending on the practical requirements, some extra constraints may be included, such as security constraint[2], or stability constraint[3;4].

As the power flow equations are nonlinear, AC OPF is non-convex. Traditionally, nonlinear optimization solving methods, e.g., Newton-type method[5] and interior point method[6], have been applied to solve the problem[7]. These methods essentially find a local optimal solution in the feasible region that satisfies the first-order optimality condition[8]. Examples presented in[8] indicate that local optima could occur due to disconnected feasible regions, loop flow, an excess of real or reactive power, or large difference in voltage angles across lines.

Global optimum means guaranteed least cost. In recent years, applying convex relaxation techniques to solve AC OPF problem and find global minimum has been carried out and a tutorial can be found in[9;10]. Relaxation problems find the lower bound of the solution to AC OPF. The two major relaxation techniques are semidefinite programming (SDP) relaxation, and second-order cone programming (SOCP) relaxation. SDP relaxation was first applied to solve AC OPF in[11]. SOCP relaxation was proposed in[12] for radial networks. In radial networks, SOCP relaxation and SDP relaxation are equivalent[9].

Though it has been studied that SDP relaxation can give global optimum for many IEEE test systems while the solutions are feasible to the original AC OPF problems (termed as "SDP exact") in[13], in some other cases, SDP relaxation leads to inexact solutions for the original problem[8;14;15]. Thus, research efforts have been devoted to achieve SDP exactness, e.g.,[16;17].

The exactness conditions for SDP and SOCP relaxations are presented in[9]. Some researches have been conducted to achieve exactness for convex relaxation through exploiting the exactness conditions. In[16;17], objective functions are modified to include penalty related to the rank-1 constraint. Reference[18] treats an AC OPF problem as an SDP relaxation problem and a non-convex rank-1 feasible region mapping problem. Alternating direction method of multipliers (ADMM) iterative procedure is then applied. In[19;20], the exactness constraints are reformulated as quadratic minor constraints. The minor constraints are approximated as convex constraints in[19]. A strengthened SOCP relaxation of AC OPF is then solved. In[20], the convex-hull descriptions of the minor constraints are examined and valid inequalities are added for SDP relaxation. Reference[21] proposes a convex iteration algorithm to solve a convex problem with a regularization term related to the maximal eigenvalue of the full PSD matrix. With the regularization term achieving zero, the solution achieves global minimum. In[22], the non-convex OPF branch flow equation is decomposed into SOCP constraint and a non-convex constraint related to the difference of two convex functions. The concave term is then approximated by linear functions and updated in each iteration. A sequential convex optimization method is implemented to carry out the iteration. The aforementioned approaches rely on convex relaxation formulations. In many cases, exact solutions can be found after dealing the exactness condition. However, large gaps are still observed for special cases[17].

In this paper, we rely on nonlinear programming formulation with a PSD matrix as the decision variable. We reformulate the rank-1 constraint as a set of quadratic minor constraints. The idea of minor constraints has been mentioned in[19] and[20]. The research in[19;20] obtains convex constraints to be used to tighten the respective convex relaxation formulations. Different from the aforementioned research, in this work, we will directly deal with all 2×2 minors and come up with a nonlinear programming formulation.

Therefore, we aim to use the same decision variables of SOCP or SDP relaxation, but to solve a nonlinear optimization problem with exactness constraints imposed. With the solution from SOCP or SDP relaxation as the starting point, nonlinear programming solvers may find a feasible solution.

Our contribution is two-fold. First, we formulate a nonlinear programming problem of AC OPF based on a new set of decision variables instead of voltage phasors. The new set of decision variables align with those in SDP/SOCP relaxation. In our formulation, rank-1 constraints are replaced by a set of quadratic equality constraints representing all 2×2 minors equal to zeros. The challenge of the formulation is that the number of those minors are very large. For a N-bus power grid, there are a total $C_N^2 C_N^2$ minors. Thus, our second contribution is to employ graph decomposition technique to significantly reduce computational burden. We first decompose a power network into lines and 3-node cycles. Instead of considering all minors, only those minors related to lines and 3-node cycles are considered. As a result, an alternative AC OPF formulation and the corresponding solver are the final outcome.

This solver is tested on special cases with known large gaps between the lower bound obtained from SDP relaxation and the upper bound obtained from MATPOWER using flat start. The solutions from the proposed formulation are the same as those from MATPOWER.

The rest of the paper is organized as follows. In Section 2, we introduced the SOCP and SDP relaxation formulations of the AC OPF. The exactness conditions are presented and reformulated as 2×2 minors equal to zero. In Section 3, we describe the cycle identification technique used for graph decomposition and the proposed formulation of the problem. Case study is presented in Section 4. We conclude this paper in Section 5.

2 SOCP/SDP relaxation of AC OPF and exactness conditions

2.1 AC OPF Problem

First we describe the original formulation of AC OPF. Considering a power network, we denote the buses as $i \in \mathcal{N}$, the transmission line as $(i, j) \in \mathcal{L}$ and the generators as $i \in \mathcal{G}$. The admittance matrix is defined as Y where $Y = G + jB$, G and B are the conductance matrix and the susceptance matrix, respectively. The classic AC OPF problem is formulated as follows.

$$\min_{U,\theta,P_g,Q_g} \quad \sum_{k\in\mathcal{G}} C_{2k}P_{gk}^2 + C_{1k}P_{gk} + C_{0k} \tag{1a}$$

$$P_i^g - P_i^d = \sum_{j\in Adj_i}^{n} U_iU_j(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}), i\in\mathcal{N} \tag{1b}$$

$$Q_i^g - Q_i^d = \sum_{j\in Adj_i} U_iU_j(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}), i\in\mathcal{N} \tag{1c}$$

$$|S_{ij}(U,\theta)| \le S_{ij}^{\max}, \quad (i,j)\in\mathcal{L} \tag{1d}$$

$$P_{gi}^{\min} \le P_{gi} \le P_{gi}^{\max}, \quad i\in\mathcal{G} \tag{1e}$$

$$Q_{gi}^{\min} \le Q_{gi} \le Q_{gi}^{\max}, \quad i\in\mathcal{G} \tag{1f}$$

$$U_i^{\min} \le U_i \le U_i^{\max}, \quad i\in\mathcal{N} \tag{1g}$$

where C_{2k}, C_{1k} and C_{0k} are the coefficients of the quadratic cost function for the generator k, P_i^g, Q_i^g are the total generated active and reactive powers from the generators connected at Bus i, P_i^d, Q_i^d are total demanded active and reactive powers at Bus i, Adj_i is the set of the buses that have direct connection with Bus i, $U \in \mathbb{R}^{|\mathcal{N}|}$ and $\theta \in \mathbb{R}^{|\mathcal{N}|}$ are the voltage magnitude vector and angle vector, respectively, S_{ij} is the complex power flow on the transmission line from Bus i to Bus j. $|.|$ notates the cardinality of a set. The decision variables are $\{P_g, Q_g, U, \theta\}$.

2.2 SOCP and SDP relaxation formulations

Define a Hermitian matrix X, and

$$X = \bar{U}\bar{U}^H, \tag{2}$$

where $\bar{U}$ is the voltage phasor vector, $\bar{U}_i = U_i\angle\theta_i$, and $(.)^H$ notates conjugate transpose. Note that the rank of X is 1.

Each element of X is:

$$X_{ij} = \bar{U}_i\bar{U}_j^* = U_iU_j\cos(\theta_i - \theta_j) + jU_iU_j\sin(\theta_i - \theta_j) \tag{3}$$

Notate the real and imaginary parts of X_{ij} as R_{ij} and $-I_{ij}$.

$$R_{ij} = \text{Re}(X_{ij}) = U_i U_j \cos(\theta_i - \theta_j) \tag{4a}$$

$$I_{ij} = \text{Im}(X_{ij}) = -U_i U_j \sin(\theta_i - \theta_j) \tag{4b}$$

The following relationship should be satisfied.

$$R_{ij}^2 + I_{ij}^2 = R_{ii} R_{jj}, \text{ or } \quad |X_{ij}| = \sqrt{X_{ii} X_{jj}} \tag{5}$$

If (5) is relaxed into an inequality constraint[12], the resulting optimization problem in (6) based on R and I is the SOCP relaxation of (1).

$$\min_{P_g, Q_g, R, I} \quad \sum_{k \in \mathcal{G}} C_{2k} P_{gk}^2 + C_{1k} P_{gk} + C_{0k} \tag{6a}$$

$$P_i^g - P_i^d = \sum_{j \in Adj_i} (G_{ij} R_{ij} - B_{ij} I_{ij}), \quad i \in \mathcal{N} \tag{6b}$$

$$Q_i^g - Q_i^d = \sum_{j \in Adj_i} (-G_{ij} I_{ij} - B_{ij} R_{ij}), \quad i \in \mathcal{N} \tag{6c}$$

$$|S_{ij}(R_{ii}, R_{ij}, I_{ij})| \leq S_{ij}^{\max}, \quad (i,j) \in \mathcal{L} \tag{6d}$$

$$P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max}, \quad i \in \mathcal{G} \tag{6e}$$

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max}, \quad i \in \mathcal{G} \tag{6f}$$

$$(U_i^{\min})^2 \leq R_{ii} \leq (U_i^{\max})^2, \quad i \in \mathcal{N} \tag{6g}$$

$$R_{ij}^2 + I_{ij}^2 \leq R_{ii} R_{jj}, \quad (i,j) \in \mathcal{L} \tag{6h}$$

For SDP relaxation, X is treated as a decision variable with its rank-1 constraint relaxed[11]. SDP relaxation is presented in (7).

$$\min_{P_g, Q_g, R, I, X} \sum_{k \in \mathcal{G}} C_{2k} P_{gk}^2 + C_{1k} P_{gk} + C_{0k} \tag{7a}$$

$$\text{s.t.Constraints:(6b)} \sim \text{(6g)}$$

$$X_{ij} = R_{ij} - j I_{ij}, \quad (i,j) \in \mathcal{L} \tag{7b}$$

$$X \succeq 0 \tag{7c}$$

where $(.) \succeq 0$ denotes $(.)$ is a PSD matrix.

In the above formulations, all U and θ in the variable set are replaced by R_{ij} and I_{ij}. For a relaxation formulation, if its solution is feasible to the original AC OPF problem, then the solution is exact. The exact conditions of SDP and SOCP have been thoroughly discussed in[9] and are presented as follows.

The exactness condition for SDP is the rank-1 constraint shown in (8).

$$X \succeq 0, \quad \text{rank}(X) = 1 \tag{8}$$

For SOCP, the exactness conditions are:

$$R_{ij}^2 + I_{ij}^2 = R_{ii}R_{jj}, \text{or} \begin{vmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{vmatrix} = 0, \text{ for } (i,j) \in \mathcal{L} \tag{9a}$$

$$\sum_{(i,j)\in c} \angle X_{ij} = 0, \quad c \in \Psi \tag{9b}$$

where Ψ is the set of cycles in the power network.

Note that that the two exactness conditions (8) and (9) are exchangeable.

2.3 2×2 Minor-based Rank-1 Constraints

To implement exactness constraints, we convert the constraints in (8) to minor constraints[19]. The reformulation is based on Proposition 3.1 in[19]: a PSD matrix X is rank 1 if and only if all its 2×2 minors are zeros and the diagonal elements of X are nonnegative. Note a $m \times m$ minor of the matrix $X \in \mathbb{C}^{n\times n}$ is defined as the submatrix of X by deleting $n - m$ rows and $n - m$ columns.

For example, suppose that for an Hermitian X, rows except i, l and columns except j, k are eliminated. The resulting minor constraint is as follows

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

$$\Rightarrow \quad X_{ij}X_{lk} - X_{ik}X_{lj} = 0$$

Separating the real and imaginary parts leads to:

$$\Rightarrow \quad \begin{cases} R_{ij}R_{lk} - I_{ij}I_{lk} - R_{lj}R_{ik} + I_{lj}I_{ik} = 0 \\ I_{ij}R_{lk} - R_{ij}I_{lk} - I_{lj}R_{ik} - R_{lj}I_{ik} = 0 \end{cases} \tag{10}$$

Thus we may find all 2×2 minors of the SDP relaxation's decision variable X. The challenge is that the number of minors is large. The total number of minor constraints in terms of X is $C_n^2 C_n^2$. Among them, C_n^2 are the number of the principal minors as shown in (9a). These minors can be expressed in R and I and there are total C_n^2 constraints in the real domain. The rest minors are non-principal minors and each can be separated into two constraints in the real domain. Considering the Hermitian matrix's feature ($X^T = X^*$), we will have $\frac{C_n^2 C_n^2 - C_n^2}{2}$ constraints to represent the non-principal 2×2 minors in the complex domain, or $C_n^2 C_n^2 - C_n^2$ non-principal minor constraints in the real domain. Note that the principal minor constraints include the exactness constraints for every lines. Therefore, the total number of constraints in real-domain related to all minors is $C_n^2 C_n^2$. For a 10-node system, this number is 2025.

Instead of dealing with the full matrix X, we now examine the SOCP exactness conditions. The first condition (9a) is related to each line. For a branch connecting Bus i and Bus j, the exactness condition is to have the principal minor related to i and j be zero. The next condition (9b) is the cycle constraint which should be enforced for every cycles.

In the following, we will show the cycle constraint can be replaced by non-principal minor constraints of each embedded 3-node cycle. There are three steps to lead to the conclusion. The *first* step is to show an n-node cycle can be decomposed into $(n-2)$ 3-node cycles, the *second* step is to show that the exactness conditions in (9a) and (9b) for a 3-node cycle indeed guarantees a PSD rank-1 matrix. Hence the conditions can be replaced by a set of 2×2 minor constraints. The *third* step is to show these minor constraints can be expressed as 9 quadratic equality constraints in the real domain.

The first step With $(n-3)$ virtual lines, any chordless cycle of n nodes can be decomposed into $(n-2)$ 3-node cycles. The cycle constraint of the chordless cycle will be replaced by $(n-2)$ cycle constraints related to those 3-node cycles.

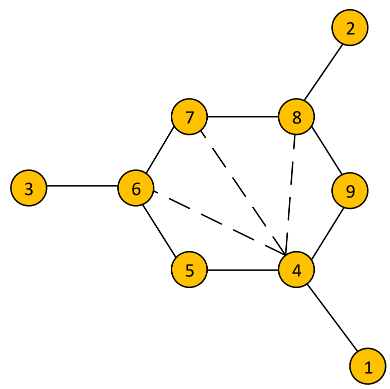

FIGURE 1: One chordless cycle become 3-node cycles with virtual lines.

Fig. 1 shows one example power network for the decompose strategy. Three virtual lines are added in the chordless cycle $\{4 \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4\}$. There are now four 3-node cycles presented: $\{4, 5, 6\}$, $\{4, 6, 7\}$, $\{4, 7, 8\}$, and $\{4, 8, 9\}$. In power grids, adding virtual lines is similar to claim that any two nodes without direct line connection can be viewed as connected through a line with infinite impedance. The numbers of virtual lines added for a n-node cycle is $(n-3)$ and the number of the resulting 3-node cycles is $(n-2)$.

Obviously, if all 3-node cycles satisfy the cycle constraint (9b), the original cycle condition can also be satisfied.

The second step Notate a 3×3 Hermitian matrix corresponding to a 3-node cycle as X_c. We will show that the exactness conditions for SOCP relaxation (9) related to a 3-node cycle is indeed equivalent to the corresponding matrix X_c being PSD and rank-1.

$$X_c = \begin{bmatrix} X_{c11} & X_{c12} & X_{c13} \\ X_{c21} & X_{c22} & X_{c23} \\ X_{c31} & X_{c32} & X_{c33} \end{bmatrix}$$

Its exactness condition is shown as follows.

$$X_{cii} \geq 0,\ i = 1, 2, 3 \tag{11a}$$
$$|X_{cij}| = \sqrt{X_{cii}X_{cjj}}, \quad (i, j) \in \{(1, 2), (2, 3), (3, 1)\} \tag{11b}$$
$$\angle X_{c12} + \angle X_{c23} + \angle X_{c31} = 0 \tag{11c}$$

(11b) indicates that all 2×2 principal minors equal to 0:

$$\begin{aligned} X_{c11}X_{c22} &= |X_{c12}|^2 = |X_{c21}|^2 \\ X_{c22}X_{c33} &= |X_{c23}|^2 = |X_{c32}|^2 \\ X_{c11}X_{c33} &= |X_{c13}|^2 = |X_{c31}|^2 \end{aligned} \tag{12}$$

Consider a non-principal minor M in X_c such as:

$$\begin{aligned} M = \det \begin{bmatrix} X_{c21} & X_{c23} \\ X_{c31} & X_{c33} \end{bmatrix} &= X_{c21}X_{c33} - X_{c23}X_{c31} \\ &= |X_{c21}||X_{c33}|\angle X_{c21} - |X_{c23}||X_{c31}|(\angle X_{c23} + \angle X_{c23}) \end{aligned}$$

According (12) and (11c), we can derive:

$$\begin{aligned} &\begin{cases} |X_{c21}||X_{c33}| = |X_{c23}||X_{c31}| \\ \angle X_{c21} = \angle X_{c23} + \angle X_{c23} \end{cases} \\ \Rightarrow \quad & |X_{c21}||X_{c33}|\angle X_{c21} = |X_{c23}||X_{c31}|(\angle X_{c23} + \angle X_{c23}) \\ \Rightarrow \quad & M = 0 \end{aligned}$$

Through the similar processes, we can find all non-principal minors are zeros. Since all 2×2 minors are 0, the rank of X_c is less than 2. And the diagonal components of X_c are all greater than 0. Hence X_c is PSD and rank-1.

The third step Basing on the previous step and the Proposition 3.1[19], for each 3-node cycle, the exactness condition or X_c is PSD and rank-1 can be replaced by a set of quadratic minor constraints. The minor constraints of X_c can be expressed as:

$$R_{ij}^2 + I_{ij}^2 = R_{ii}R_{jj}, \quad (i,j) \in \{(1,2),(2,3),(3,1)\} \tag{13a}$$
$$R_{12}R_{23} - I_{12}I_{23} - R_{22}R_{13} = 0, \tag{13b}$$
$$I_{12}R_{23} + R_{12}I_{23} - R_{22}I_{13} = 0, \tag{13c}$$
$$R_{23}R_{13} + I_{23}I_{13} - R_{33}R_{12} = 0, \tag{13d}$$
$$I_{23}R_{13} - R_{23}I_{13} + R_{33}I_{12} = 0, \tag{13e}$$
$$R_{13}R_{12} + I_{13}I_{12} - R_{11}R_{23} = 0, \tag{13f}$$
$$I_{13}R_{12} - R_{13}I_{12} - R_{11}I_{23} = 0. \tag{13g}$$

As a summary, for a power grid that has been decomposed into lines (the set is notated as $\mathcal{L}$), virtual lines (notated as $\mathcal{L}_{\mathcal{V}}$), and 3-node cycles (notated as Ψ), there will be $|\mathcal{L}| + |\mathcal{L}_{\mathcal{V}}|$ constraints related to the principal minors, and $6 \times |\Psi|$ constraints related to non-principal minors. Take the example shown in Fig. 1, the system has a total 9 lines with one 6-node chordless cycle. Three virtual lines are added $4-6$, $4-7$, $4-8$ to decompose the chordless cycle into 4 3-node cycles. The system's exactness constraints consist of 12 constraints related to the 12 principal minors corresponding to 12 lines (9 lines and 3 virtual lines) and 6×4 related to the non-principal minors. Total, there are 36 quadratic equality constraints.

3 Rank-1 PSD Matrix-Based Nonlinear Programming Formulation

In Section II, the exactness conditions have been converted to quadratic equality constraints. A nonlinear programming problem can now be formulated with those constraints. The next step is to identify cycles in a power network and decompose any chordless cycle with size greater than 3 into 3-node cycles. With all virtual lines and 3-node cycles identified, the nonlinear programming formulation can be derived.

3.1 Cycle Basis Identification

A cycle basis of a graph is the set of cycles with each cycle having only one of its edges common with the spanning tree of the graph. We use the cycle identification algorithm in[23] to identify the cycles. A MATLAB based toolbox[24] is applied to find a cycle basis. The algorithm first searches a minimal spanning tree of the network and then adds the rest of the lines back one by one. Each added line will be considered as a token to identify one cycle. For example, consider the five buses network in Fig. 2, the set of the lines is $\mathcal{L} = \{(1,2),(2,3),(3,4),(4,5),(5,1),(4,1)\}$. To identify its cycles, first the cycle identification algorithm will start from the minimal spanning tree in Fig. 3. This spanning tree contains lines $(1,2),(2,3),(1,4),(1,5)$. Next, the remaining lines are added back one by one. After adding $(3,4)$, we obtain a cycle $c_a = \{1-2-3-4-1\}$. After adding $(4,5)$, we get a cycle $c_b = \{1-4-5-1\}$.

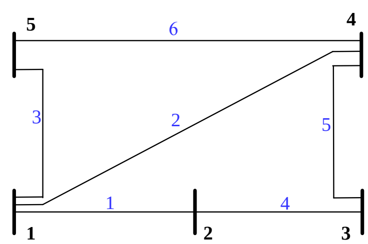

FIGURE 2: Five-bus test case with two cycles. Cycle a: nodes $\{1,2,3,4\}$, lines $\{(1,2),(2,3),(3,4),(4,1)\}$; Cycle b: nodes $\{1,4,5\}$, lines $\{(1,4),(4,5),(5,1)\}$.

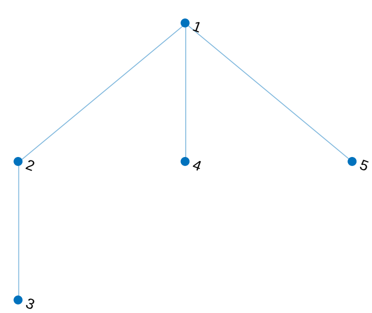

FIGURE 3: Spanning tree of the five-bus test system.

3.2 Nonlinear Programming Problem Formulation

Through the cycle identification algorithm and the graph decomposition process, we obtain a set of virtual lines and 3-node cycles. The set of all lines including virtual lines are notated as $\mathcal{L}_{ch} = \mathcal{L}_{\mathcal{V}} \cup \mathcal{L}$. The proposed rank-1 nonlinear programming formulation is as follows.

$$\min \quad \sum_{k\in\mathcal{G}} C_{2k}P_{gk}^2 + C_{1k}P_{gk} + C_{0k} \tag{14a}$$

$$P_i^g - P_i^d = \sum_{j\in Adj_i} (G_{ij}R_{ij} - B_{ij}I_{ij}), \quad i \in \mathcal{N} \tag{14b}$$

$$Q_i^g - Q_i^d = \sum_{j\in Adj_i} (-G_{ij}I_{ij} - B_{ij}R_{ij}), \quad i \in \mathcal{N} \tag{14c}$$

$$|S_{ij}(R_{ii}, R_{ij}, I_{ij})| \le S_{ij}^{\max}, \quad (i,j) \in \mathcal{L} \tag{14d}$$

$$P_{gi}^{\min} \le P_{gi} \le P_{gi}^{\max}, \quad i \in \mathcal{G} \tag{14e}$$

$$Q_{gi}^{\min} \le Q_{gi} \le Q_{gi}^{\max}, \quad i \in \mathcal{G} \tag{14f}$$

$$(U_i^{\min})^2 \le R_{ii} \le (U_i^{\max})^2, \quad i \in \mathcal{N} \tag{14g}$$

$$R_{ij}^2 + I_{ij}^2 = R_{ii}R_{jj}, \quad (i,j) \in \mathcal{L}_{ch} \tag{14h}$$

For each 3 nodes cycle $c = \{i,j,k\} \in \Psi$

$$R_{ij}R_{jk} - I_{ij}I_{jk} - R_{jj}R_{ik} = 0, \tag{14i}$$

$$I_{ij}R_{jk} + R_{ij}I_{jk} - R_{jj}I_{ik} = 0, \tag{14j}$$

$$R_{jk}R_{ik} + I_{jk}I_{ik} - R_{kk}R_{ij} = 0, \tag{14k}$$

$$I_{jk}R_{ik} - R_{jk}I_{ik} + R_{kk}I_{ij} = 0, \tag{14l}$$

$$R_{ik}R_{ij} + I_{ik}I_{ij} - R_{ii}R_{jk} = 0, \tag{14m}$$

$$I_{ik}R_{ij} - R_{ik}I_{ij} - R_{ii}I_{jk} = 0. \tag{14n}$$

This problem is a quadratically constrained quadratic program (QCQP) problem with quadratic equality constraints (14h) to (14n). The problem can be solved by nonlinear programming solvers, e.g., IPOPT.

Compared with MATPOWER, the decision variables in (14) are no longer voltage phasors. Instead, R and I can be initialized using the solution of the PSD matrix X from a SDP relaxation AC OPF solver developed in[25].

3.3 Voltage Recover Technique

The solution from (14) is a set of R_{ii}, R_{ij}, and I_{ij}, or partial information of a PSD matrix X. Voltage vectors can be recovered from the decision variables R and I. R_{ii} will be found for every bus. R_{ij} and I_{ij} are related to all lines and virtual lines. we implement a voltage recovery method[9] through visiting the spanning tree of the network.

First, we define the phase angle of voltage phasor at the reference node as $0°$. We identify the spanning tree of the power network, and define path from the reference node to the node i as $\mathcal{P}_i$. Then the voltage phasor at any node i can be recovered through the following equations.

$$\begin{aligned} U_i &= \sqrt{R_{ii}} \\ \theta_i &= -\sum_{(j,k)\in\mathcal{P}_i} \angle(R_{jk} - jI_{jk}) \end{aligned}$$

4 Case studies

In this section, we present the case study results. Our numerical experiments were performed on an Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.30 GHZ (2 processors) computer. All solvers are implemented on Matlab 2017a. The proposed non-linear programming AC OPF formulation was implemented in MATLAB using Yalmip and select `IPOPT` as the main solver. The configurations of the `IPOPT` are: convergence tolerance is 1×10^{-7}; maximum number of iteration is 10000; update strategy for barrier parameter is “adaptive”; Hessian information is “limit-memory”; and the other configurations are default.

To obtain the initial point for the nonlinear solver, we solve OPF problems first through a CVX[26] based sparse SOCP/SDP relaxation solver[25]. The objective value of the SOCP/SDP solver and the maximal rank of the corresponding PSD submatrices will be given for each test case. MATPOWER with `IPOPT` as the solver is used for comparison.

4.1 Test setup

In results tables, the column "Rank" list the maximal rank of all submatrices for each case. For each submatrix, rank is counted by considering only the eigenvalues that are greater than the 0.001% of the maximal eigenvalue; N_vlines is the number of the virtual lines; N_lines is the number of the original lines; N_cycle3 is the number of the 3-node cycles based on the decomposition results. Time is the total CPU seconds in IPOPT to solve the problem. The Gap means the relaxation gap which is calculated trough the following equation:

$$Gap = \frac{UB - Obj}{UB} \times 100\%$$

where UB is the Upper bound of the objective value which is calculated by MATPOWER; Obj is the objective value which is computed through the correspond method.

4.2 Results

We tested the proposed formulation on two sets of cases. The first set of cases are chosen from the NICTA test archive[27]. Gaps of these cases are small or zeros. The second set of cases are special cases which are modified to have large gaps. These cases are selected from the test achieves which are provided by[14] and[8]. The related results are listed in the Table 1 and 2 respectively.

In Table 1, the results show that for all tested standard cases, our formulation provides the same objective values as those from MATPOWER. In Table 2, we listed test results on modified cases. From the table, we can see that for cases with large gaps, the proposed formulation is capable to obtain the same objective value as MATPOWER . These results indicate that the proposed formulation can provide similar quality results to the original ACOPF formulation that is implemented in MATPOWER. Moreover, according the number of 3-node cycles, we can see the improvement on the number of the added equality constraints. For example, in "nesta_case57_ieee", N_vlines$=55$, N_lines$=78$, N_cycle3$=77$, which means the number of the added equality constraints are $77 \times 6 + 78 + 55 = 595$. This number is much less than $C_{57}^2 C_{57}^2 = 2547216$ which is the number of the added equality constraints without the decomposition.

For the cases from "case9Tree" to "case57Tree" in table 2, as they are radial networks, we do not add any virtual line on them. On the computation time, we can see the proposed method has good performance for the small size cases. For the medium size cases (nesta_case89_pegase, nesta_case118_ieee, case57Tree), the computational cost is relatively high, but still acceptable.

5 Conclusion

In this paper, we proposed a nonlinear programming formulation for AC OPF. This formulation is based on decision variables that align with SOCP/SDP relaxation. The proposed formulation exploits power network sparsity feature and employs a small set of minor constrains related to all 3-node cycles as equality constraints to enforce rank-1 constraint. Case study results demonstrate the correctness of this formulation.

6 All Symbols

Index and set:

$\mathcal{N}$	Set of buses.
$\mathcal{L}$	Set of transmission lines.
$\mathcal{L}_{\mathcal{V}}$	Set of virtual lines.
$\mathcal{L}_{ch}$	Set of all lines which is equal to $\mathcal{L}_{\mathcal{V}} \cup \mathcal{L}$.
$\mathcal{G}$	Set of generators.
Ψ	Set of cycles.
Adj_i	Set of the buses that have direct connection with Bus i.
$\mathcal{P}_i$	Path from reference node to the node i.

Parameters:

Y	Admittance matrix of the system.
$G\backslash G_{ij}$	Conductance matrix of the system\element of G.
$B\backslash B_{ij}$	Susceptance matrix of the system\element of B.
C_{2k}, C_{1k}, C_{0k}	Coefficients of the quadratic cost function for the generator k.
P_i^d, Q_i^d	Total active and reactive powers demand Bus i.
$P_{gi}^{min}, P_{gi}^{max}$	Minimum and maximum active power of genertor i.
$Q_{gi}^{min}, Q_{gi}^{max}$	Minimum and maximum reactive power of genertor i.
U_i^{min}, U_i^{max}	Minimum and maximum bus voltages.
S_{ij}^{max}	Complex power limit of transmission line (i, j).

Variables:

P_i^g, Q_i^g	Total active and reactive powers from the generators connected at Bus i.
U	Voltage magnitude vector.
θ	Voltage angle vector.
S_{ij}	The complex power flow on the transmission line from Bus i to j
$\bar{U}$	The voltage phasor vector.
$X\backslash X_{ij}$	A Hermitian matrix which is equal to $\bar{U}\bar{U}^H$ \ Element of X.
R_{ij}, I_{ij}	Real and imaginary parts of X_{ij}.
M	Non-principle minor of X.

References

1. Dommel Hermann W, Tinney William F. Optimal power flow solutions *IEEE Transactions on power apparatus and systems.* 1968:1866–1876.

2. Monticelli A, Pereira MVF, Granville S. Security-constrained optimal power flow with post-contingency corrective rescheduling *IEEE Transactions on Power Systems.* 1987;2:175–180.

3. Milano Federico, Cañizares Claudio A, Conejo Antonio J. Sensitivity-based security-constrained OPF market clearing model *IEEE Transactions on power systems.* 2005;20:2051–2060.

4. Zárate-Miñano Rafael, Milano Federico, Conejo Antonio J. An OPF methodology to ensure small-signal stability *IEEE Transactions on Power Systems.* 2011;26:1050–1061.

5. Sun David I, Ashley Bruce, Brewer Brian, Hughes Art, Tinney William F. Optimal power flow by Newton approach *IEEE Transactions on Power Apparatus and systems.* 1984:2864–2880.

6. Torres Geraldo Leite, Quintana Victor Hugo. An interior-point method for nonlinear optimal power flow using voltage rectangular coordinates *IEEE transactions on Power Systems.* 1998;13:1211–1218.

7. Castillo Anya, O'Neill Richard P. Survey of approaches to solving the ACOPF (OPF Paper 4) *US Federal Energy Regulatory Commission, Tech. Rep.* 2013.

8. Bukhsh Waqquas A, Grothey Andreas, McKinnon Ken IM, Trodden Paul A. Local solutions of the optimal power flow problem *IEEE Transactions on Power Systems.* 2013;28:4780–4788.

9. Low Steven H. Convex relaxation of optimal power flow—Part I: Formulations and equivalence *IEEE Transactions on Control of Network Systems.* 2014;1:15–27.

10. Low Steven H. Convex relaxation of optimal power flow—Part II: Exactness *IEEE Transactions on Control of Network Systems.* 2014;1:177–189.

11. Bai Xiaoqing, Wei Hua, Fujisawa Katsuki, Wang Yong. Semidefinite programming for optimal power flow problems *International Journal of Electrical Power & Energy Systems.* 2008;30:383–392.

12. Jabr Rabih A. Radial distribution load flow using conic programming *IEEE transactions on power systems.* 2006;21:1458–1459.

13. Lavaei Javad, Low Steven H. Zero duality gap in optimal power flow problem *IEEE Transactions on Power Systems.* 2012;27:92–107.

14. Kocuk Burak, Dey Santanu S, Sun Xu Andy. Inexactness of SDP relaxation and valid inequalities for optimal power flow *IEEE Transactions on Power Systems.* 2016;31:642–651.

15. Lesieutre Bernard C, Molzahn Daniel K, Borden Alex R, DeMarco Christopher L. Examining the limits of the application of semidefinite programming to power flow problems in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*:1492–1499IEEE 2011.

16. Madani Ramtin, Ashraphijuo Morteza, Lavaei Javad. Promises of conic relaxation for contingency-constrained optimal power flow problem *IEEE Transactions on Power Systems.* 2016;31:1297–1307.

17. Molzahn Daniel K, Josz Cédric, Hiskens Ian A, Panciatici Patrick. A laplacian-based approach for finding near globally optimal solutions to opf problems *IEEE Transactions on Power Systems.* 2017;32:305–315.

18. You Seungil, Peng Qiuyu. A non-convex alternating direction method of multipliers heuristic for optimal power flow in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*:788–793IEEE 2014.

19. Kocuk Burak, Dey Santanu S, Sun X Andy. Matrix minor reformulation and SOCP-based spatial branch-and-cut method for the AC optimal power flow problem *arXiv preprint arXiv:1703.03050.* 2017.

20. Chen Chen, Atamtürk Alper, Oren Shmuel S. A spatial branch-and-cut method for nonconvex QCQP with bounded complex variables *Mathematical Programming.* 2017;165:549–577.

21. Wang Wei, Yu Nanpeng. Chordal Conversion based Convex Iteration Algorithm for Three-phase Optimal Power Flow Problems *IEEE Transactions on Power Systems.* 2017.

22. Wei Wei, Wang Jianhui, Li Na, Mei Shengwei. Optimal Power Flow of Radial Networks and its Variations: A Sequential Convex Optimization Approach *IEEE Transactions on Smart Grid.* 2017;8:2974–2987.

23. Mehlhorn Kurt, Michail Dimitrios. Implementing minimum cycle basis algorithms *Journal of Experimental Algorithmics (JEA).* 2007;11:2–5.

24. Iglin Sergii. grTheory - Graph Theory Toolbox https://www.mathworks.com/matlabcentral/fileexchange/4266-grtheory-graph-theory-toolbox?focused=5177310&tab=function 2011.

25. Ma M., Fan L., Miao Z., Zeng B., Ghassempour H.. A Sparse Convex AC OPF Solver and Convex Iteration Implementation Based on 3-Node Cycles *under review, Electric Power Systems Research.* 2019.

26. Grant Michael, Boyd Stephen, Ye Yinyu. CVX: Matlab software for disciplined convex programming 2008.

27. Coffrin C, Gordon D, Scott P. NESTA, the NICTA energy system test case archive 2014.

List of Tables

TABLE 1
Test case results

	SDP Relaxation			Matpower	Nonlinear Rank 1 Method					
Case	Obj	Rank	%Gap	Obj	Obj	%Diff	Time	N_vlines	N_lines	N_cycle3
nesta_case3_lmbd	5789.914	2	0.04	5812.643	5812.643	0.00	0.08	0	3	1
nesta_case5_pjm	16635.76	2	5.22	17551.891	17551.889	0.00	0.13	1	6	3
nesta_case9_wscc	5296.685	2	0.00	5296.686	5296.686	0.00	0.13	3	9	4
nesta_case57_ieee	1143.280	2	0.00	1143.283	1143.283	0.00	2.43	55	78	77
nesta_case89_pegase	5820.217	2	0.00	5820.387	5820.387	0.00	52.84	55	206	173
nesta_case118_ieee	3689.495	3	0.01	3692.891	3692.891	0.00	50.35	73	179	135

TABLE 2
Special test case results

	SDP Relaxation			Matpower	Nonlinear Rank 1 Method					
Case	Obj	Rank	%Gap	Obj	Obj	%Diff	Time	N_vlines	N_lines	N_cycle3
WB5	946.530	2	0.01	946.584	946.584	0.00	0.08	1	6	3
case9mod	2753.041	2	10.84	3087.842	3087.842	0.00	0.15	3	9	4
case39mod1	10804.055	2	3.72	11221.003	11221.003	0.00	1.81	21	46	29
case39mod2	940.341	2	0.15	941.738	941.738	0.00	1.85	21	46	29
case9Tree	5335.701	2	52.70	11279.476	11279.476	0.00	0.09	0	8	0
case14Tree	11861.899	2	0.59	11932.252	11932.252	0.00	0.16	0	13	0
case30Tree	4244.549	2	11.47	4794.313	4794.314	0.00	0.31	0	29	0
case39Tree	44868.452	2	0.37	45037.039	45037.042	0.00	0.49	0	38	0
case57Tree	10458.099	2	13.58	12100.849	12100.856	0.00	25.81	0	56	0

Legends of figures:

FIGURE 1: One chordless cycle become 3-node cycles with virtual lines.

FIGURE 2: Five-bus test case with two cycles. Cycle a: nodes $\{1, 2, 3, 4\}$, lines $\{(1, 2), (2, 3), (3, 4), (4, 1)\}$; Cycle b: nodes $\{1, 4, 5\}$, lines $\{(1, 4), (4, 5), (5, 1)\}$.

FIGURE 3: Spanning tree of the five-bus test system.