GLOBAL OPTIMIZATION METHODS FOR OPTIMAL POWER FLOW AND TRANSMISSION SWITCHING PROBLEMS IN ELECTRIC POWER SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Burak Kocuk

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2016

GLOBAL OPTIMIZATION METHODS FOR OPTIMAL POWER FLOW AND TRANSMISSION SWITCHING PROBLEMS IN ELECTRIC POWER SYSTEMS

Approved by:

Assoc. Professor Santanu S. Dey and
Assist. Professor X. Andy Sun,
Advisor
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Professor Natashia Boland
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Assist. Professor X. Andy Sun,
Advisor
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Professor Daniel Bienstock
Department of Industrial Engineering
and Operations Research
Columbia University

Professor Shabbir Ahmed
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Date Approved: 30 June 2016

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisors Dr. Santanu S. Dey and Dr. X. Andy Sun for their guidance during my doctoral studies. They always treated me as a colleague rather than a student. I appreciate their diligent and prompt feedback to my research, without which, I would not have had such a productive dissertation work in three years.

I wish to thank the rest of my dissertation committee members, Dr. Shabbir Ahmed, Dr. Daniel Bienstock and Dr. Natashia Boland, for their constructive comments and suggestions, which helped improve the presentation of my thesis. Also, I would like to thank Dr. Arkadi Nemirovski for his support during my early years at Georgia Tech.

I am thankful to all my former and current fellow doctoral students at ISyE for their friendship. I wish to thank Alvaro, Bahar, Can, Çağlar, Diego, Evren, Ezgi, Işıl, İdil, İlke, Jikai, Kevin, Mathias, Melih, Melike, Minkyoung, Murat, Oğuzhan, Süleyman, Tuğçe, and all the members of the Turkish community for their help whenever I needed.

Most importantly, I would like to express my deepest gratitude to my family, especially to my mother Serpil and my sister Tuğba, for their continuous support, love and patience. I owe a lot to these two special ladies, without whom I would never become the person I am now. Last but not least, I would like to thank my girlfriend Beste for her affection, patience and understanding. I met her a year ago but it feels like I have known her forever. She is truly the most wonderful thing happened to me at Georgia Tech.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Power engineering is concerned with the generation, transmission, and distribution of electricity over electric power network. In this thesis, we focus on two operational level optimization problems from power system planning, namely the Optimal Power Flow Problem (OPF) and the Optimal Transmission Switching (OTS) Problem. The former is a nonlinear network problem and the latter is the network design version of the first one. Due to nonlinearity induced by alternating current power flow equations, these two optimization problems, defined precisely in Chapter 1, are nonconvex and require efficient global optimization methods.

In Chapter 2, we consider Alternating Current OPF (AC OPF) problem over radial networks and analyze the approximation outcomes of the semidefinite programming (SDP) relaxation, which is proven to be exact over radial networks under some technical conditions. We design a library of instances that demonstrate positive SDP optimality gaps when these conditions do not hold. Finally, we propose valid inequalities and variable bound tightening techniques that significantly improve the computational performance of a global optimization solver. Our work demonstrates the need of developing efficient global optimization methods for the solution of OPF even in the simple but fundamental case of radial networks.

In Chapter 3, we focus on the solution of AC OPF problem for the general case of meshed networks. This chapter proposes three strong second-order cone programming (SOCP) relaxations for the AC OPF problem by exploiting the underlying network structure. Two of these three relaxations are incomparable to the standard SDP relaxation of OPF. Extensive computational experiments show that these relaxations

have numerous advantages over existing convex relaxations in the literature in terms of both quality of the relaxations and practicability to obtain feasible solutions within a time framework that is compatible with the real-time operations in the current industry practice.

In Chapter 4, we again consider the AC OPF problem with a particular emphasis on solving more challenging instances. We analyze the properties of the minors and submatrices of the matrix variable in a lifted formulation, and obtain a stronger SOCP relaxation than the ones proposed in Chapter 3 by the addition of valid inequalities and improved bound tightening techniques. We also propose an SOCP based spatial branch-and-cut algorithm to solve the most difficult instances. Overall, our methodology provides a computationally tractable approach to obtain strong relaxation bounds for some of the hardest OPF instances from the literature.

In Chapter 5, we consider the so-called Direct Current OTS problem, which incorporates a linear approximation to nonconvex AC power flow equations. Most research on DC OTS has focused on heuristic algorithms for generating quality solutions. However, the mathematical theory of the DC OTS problem is less well-developed. In this chapter, we formally establish that DC OTS is NP-Hard. We characterize the convex hull of a cycle-induced relaxation inspired by Kirchoff's Voltage Law, and this characterization provides strong valid inequalities that can be used in a cutting-plane approach to solve the DC OTS. We give details of a practical implementation, and show promising computational results on standard benchmark instances.

In Chapter 6, we focus on the OTS problem with the full AC power flow model since the commonly-used DC approximation of the power flow model is known to result in inaccurate flow solutions. In this chapter, we propose a new exact formulation for AC OTS and its mixed-integer second-order cone programming (MISOCP) relaxation. We improve this relaxation via several types of strong valid inequalities

inspired by the developments for the AC OPF problem in Chapter 3. We also propose a practical algorithm to obtain high quality feasible solutions for the AC OTS problem. Extensive computational experiments show that the proposed formulation and algorithms lead to significant cost benefits with provably tight bounds.

CHAPTER I

INTRODUCTION

Power engineering is concerned with the generation, transmission, and distribution of electricity over electric power network, which is arguably one of the largest engineering systems in the world, as electricity is used by billions of people in industry, homes, businesses, and transportation continuously. The size of electric utility industry exceeds billions of dollars and its utilization in a cost-effective manner while providing reliable accessibility is extremely important.

Power system planning is a hierarchical decision making environment [94]. Long term planning in power systems includes decisions about generation and transmission capacity expansion in order to satisfy future demand forecasts. These decisions are typically taken over a time horizon of 5-15 years or longer. Medium term planning involves decisions about scheduling of maintenance of key equipments such as generators. These decisions are usually taken annually or semi-annually. Short term decisions are typically taken daily and involve decisions such as which generators to use for the next few days. This fundamental operational problem is known as *Unit Commitment*.

In this thesis, we focus on real-time operations with decisions taken every 5-15 minutes. We consider two fundamental problems: The first problem is known as *Economic Dispatch* or *Optimal Power Flow*. Given the output of Unit Commitment, i.e. the generation on and off schedules, Optimal Power Flow is used to match the supply and demand in real time. This is an especially challenging optimization problem due to nonconvexities introduced by the power flow equations and limited computational time budget.

The second problem is known as *Optimal Transmission Switching*. Recent advances in the modern transmission control and relay technologies have made it possible to change the network topology by switching on and off transmission lines. Quite interestingly, allowing this flexibility may reduce costs and improve reliability of the underlying power network. In Optimal Transmission Switching, the network topology is optimized to obtain the minimum cost dispatch while satisfying the physical laws in Optimal Power Flow. Therefore, this problem contains additional nonconvexities introduced due to discrete decisions based on line switching. In the remainder of this chapter, we formally define these two problems and present the corresponding optimization models.

1.1 Optimal Power Flow

1.1.1 Introduction and Literature Survey

Optimal Power Flow (OPF) was first introduced in the 1960s [21] and still remains to be a fundamental optimization problem in electrical power systems analysis. There are two challenges in the solution of OPF. First, it is an operational level problem solved every few minutes, hence the computational budget is limited. Second, it is a nonconvex optimization problem on a large-scale power network of thousands of buses, generators, and loads. The importance of the problem and the aforementioned difficulties have produced a rich literature, see e.g. [81, 82, 34, 35, 20]. Roughly speaking, we can categorize the previous work into three categories.

The first category of algorithms finds local optimal solutions or stationary points of the OPF problem. The academic literature has focused on improving nonlinear optimization methods such as the interior point methods (IPM) to compute local optimal solutions, see e.g. [108, 99, 49, 106]. A well-known implementation of IPM tailored for the OPF problem is MATPOWER [111]. Although these local methods are effective in solving IEEE test instances, they do not offer any quantification of

the quality of the solution. For instance, in [19], there are several examples which have multiple local optima and it has been shown that local solvers tend to converge to the solution which is closest to the initial guess.

The second category of algorithms attempts to obtain global optimal solutions of OPF by solving convex relaxations. In the recent years, much research interests have been drawn to the this approach. In particular, the second-order cone programming (SOCP) and the semidefinite programming (SDP) relaxations are first applied to the OPF problem in [52], and [6, 5] and [67]. Among these two approaches, the SDP relaxation and its variations have drawn significant attention due to their strength. Since convex conic programs are polynomially solvable, the SDP relaxation offers an effective way for obtaining global optimal solutions to OPF problems whenever the relaxation is exact. Unfortunately, the exactness of the SDP relaxations can be guaranteed only for a restricted class of problems under some assumptions as listed below:

- One of the early works that popularizes this approach is [67]. It is shown that the SDP relaxation is tight for a resistive network with no reactive loads where demand over-satisfaction is allowed, as long as the dual variables are positive. It was conjectured that under normal operating conditions, the SDP relaxation is tight.

- However, [69] gives a very simple counterexample (a 3-bus cycle) with nonzero optimality gap.

- In [91], it is proven that SDP relaxation is exact if load over-satisfaction is allowed and a sufficient number of virtual phase shifters are present.

- An attempt to solve OPF using SDP relaxation is made in [110] for radial networks. In this work, it is proven that under operational constraints on voltage magnitudes, line losses, and line flows, the SDP relaxation is tight if there are

no lower bounds on real and reactive power generation at any bus. Similar results are also obtained in [18, 17] without line limit constraints.

- In [66], it is proven that if voltage magnitudes are fixed, then the convex relaxations are tight under practical angle restrictions for radial networks in the presence of only real power lower bounds. This result extends to the case with variable voltage magnitudes under reasonable assumptions.

A comprehensive survey can be found in [71, 72].

As we observe above, the exactness of the SDP relaxation can only be guaranteed for special classes of OPF instances, often when we disregard some generation lower bounds. Unfortunately, if the SDP relaxation is not tight, the physical meaning of its solution is not easy to recover. A way to further strengthen the SDP relaxation is to solve a hierarchy of moment relaxation problems [65, 86]. This approach is used in [55] to globally solve small-size problems, and is also used in [80] to obtain tighter lower bounds for larger problems of 300-bus systems. However, due to the NP-hardness of the OPF problem [102, 67], in general the order of the Lasserre hierarchy required to obtain a global optimal solution can be arbitrarily large. Furthermore, even the global optimal objective function value is achieved, the solution matrices may not be rank one, which poses another challenge in terms of recovering an optimal voltage solution [66]. This indicates the computational difficulty of the SDP relaxation approach to practically solve real-world sized power networks with more than a thousand buses. For such large-scale OPF problems, a straightforward use of IPM to solve the SDP relaxation becomes prohibitively expensive. Interesting works have been done to exploit the sparsity of power networks as in [48, 79, 74, 80, 75]. The underlying methodology utilizes techniques such as chordal graph extension, tree-width decomposition, and matrix completion, as proposed and developed in [36] and [83].

More recently, there is a growing trend to use computationally less demanding relaxations based on linear programming (LP) and SOCP to solve the OPF problem.

For instance, linear and quadratic envelopes for trigonometric functions in the polar formulation of the OPF problem are constructed in [28, 45, 27]. In [14], LP based outer approximations are proposed which are strengthened by incorporating several types of valid inequalities.

The third category of algorithms attempts to remove the pitfalls of the previous two approaches by endeavoring to obtain globally optimal solutions. One such algorithm based on branch-and-bound method is proposed in [87] for the solution of OPF. Lagrangian relaxation is used to find lower bounds while a local solver (IPOPT) is utilized to obtain upper bounds. Global solution techniques are in their infancy today and much work needs to be done to make them practically efficient.

1.1.2 Formulation

Consider a power network $\mathcal{N} = (\mathcal{B}, \mathcal{L})$, where $\mathcal{B}$ denotes the node set, i.e., the set of buses, and $\mathcal{L}$ denotes the edge set, i.e., the set of transmission lines. A typical power system involves a power network, that is, a set of nodes called *buses* and edges called *transmission lines*. See Figure 1 for an illustration. This small power network contains 14 buses, 20 transmission lines and 5 generators. Generation units (i.e. electric power generators) are connected to a subset of buses, denoted as $\mathcal{G} \subseteq \mathcal{B}$. We assume that there is electric demand, also called load, at every bus. The aim of the optimal power flow problem is to satisfy demand at all buses with the minimum total production costs of generators such that the solution obeys the physical laws (e.g., Ohm's Law and Kirchoff's Law) and other operational restrictions (e.g., transmission line flow limit constraints).

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} (resp. b_{ii}) is the shunt conductance (resp. susceptance) at bus $i \in \mathcal{B}$ and $\mathrm{i} = \sqrt{-1}$. Let p_i^g, q_i^g (resp. p_i^d, q_i^d) be the real and reactive power output of the

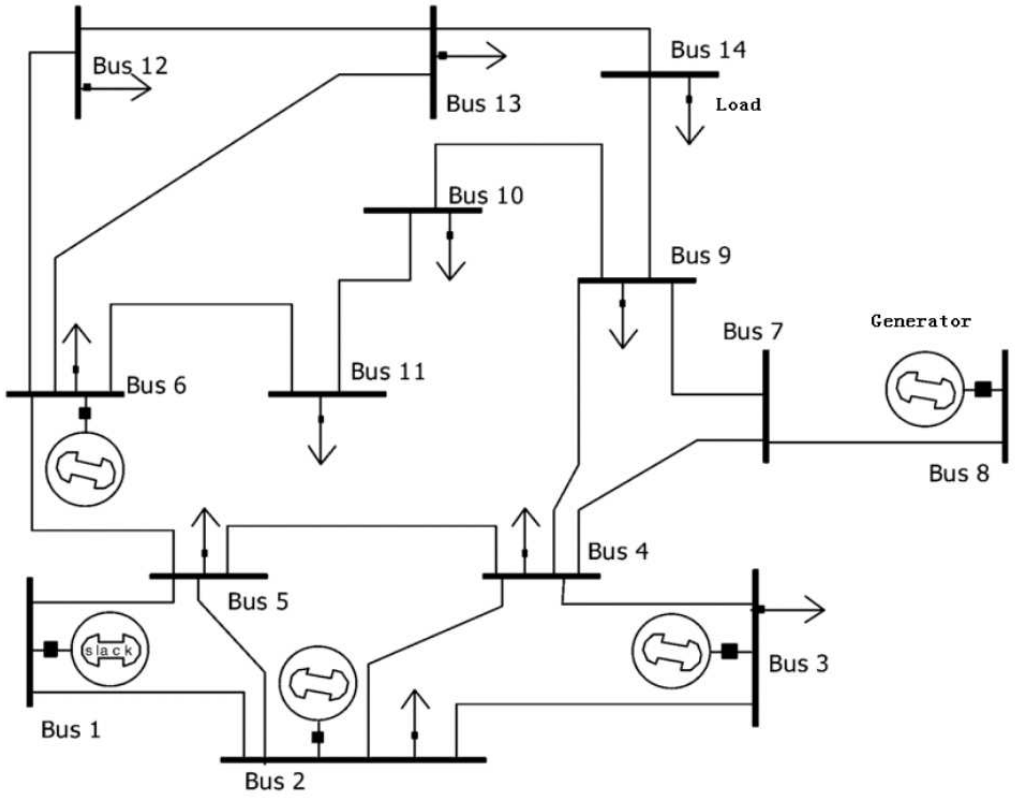


Figure 1: Standard IEEE 14-bus network. Nodes with circles and arrows represent generator and load buses, respectively [24].

generator (resp. load) at bus i. The complex voltage (also called voltage phasor) V_i at bus i can be expressed either in the rectangular form as $V_i = e_i + \mathrm{i} f_i$ or 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 voltage magnitude and θ_i is the angle of the complex voltage. In power system analysis, the voltage magnitude is usually normalized against a unit voltage level and is expressed in per unit (p.u.). For example, if the unit voltage is 100kV, then 110kV is expressed as 1.1 p.u.. In transmission systems, the bus voltage magnitudes are usually restricted to be close to the unit voltage level to maintain system stability. We also define active and reactive power flow along a line as p_{ij} and q_{ij}, respectively. We assume that there is at most one generator at each bus. This assumption is without loss of generality, since the models we present can be extended to the case with multiple generators by replacing the dispatch variable by the sum of different dispatch variables attached to

a particular bus.

With the above notation, the OPF problem is given in the so-called rectangular formulation as follows [94]:

$$\min \ \sum_{i\in\mathcal{G}} C_i(p_i^g) \tag{1a}$$

$$\text{s.t. } p_i^g - p_i^d = \ g_{ii}(e_i^2+f_i^2) + \sum_{j\in\delta(i)} p_{ij} \qquad i\in\mathcal{B} \tag{1b}$$

$$q_i^g - q_i^d = -b_{ii}(e_i^2+f_i^2) + \sum_{j\in\delta(i)} q_{ij} \qquad i\in\mathcal{B} \tag{1c}$$

$$p_{ij} = -G_{ij}(e_i^2+f_i^2) + G_{ij}(e_ie_j+f_if_j) - B_{ij}(e_if_j-e_jf_i) \qquad (i,j)\in\mathcal{L} \tag{1d}$$

$$q_{ij} = \ B_{ij}(e_i^2+f_i^2) - B_{ij}(e_ie_j+f_if_j) - G_{ij}(e_if_j-e_jf_i) \qquad (i,j)\in\mathcal{L} \tag{1e}$$

$$\underline{V}_i^2 \le e_i^2+f_i^2 \le \overline{V}_i^2 \qquad i\in\mathcal{B} \tag{1f}$$

$$p_{ij}^2+q_{ij}^2 \le \overline{S}_{ij}^2 \qquad (i,j)\in\mathcal{L} \tag{1g}$$

$$p_i^{\min} \le p_i^g \le p_i^{\max} \qquad i\in\mathcal{B} \tag{1h}$$

$$q_i^{\min} \le q_i^g \le q_i^{\max} \qquad i\in\mathcal{B}. \tag{1i}$$

Here, $C_i(p_i^g)$ in (1a) represents the production cost of generator i, which typically is either a linear or a convex quadratic nondecreasing function of p_i^g. Constraints (1b)-(1c) enforce flow conservation at each bus i, where $\delta(i)$ is the set of buses adjacent to i. Constraints (1d)-(1e) define real and reactive power across line (i,j), namely p_{ij} and q_{ij}, in terms of the complex voltage at buses i and j while constraint (1g) imposes an upper bound on apparent power over line (i,j). Constraint (1f) limits the upper and lower bounds on the bus voltage magnitudes. Usually $\underline{V}_i$ and $\overline{V}_i$ are close to the unit voltage. Constraints (1h)-(1i) are the upper and lower bounds on generator i's real and reactive power, respectively. Here, we have $p_i^{\min} = p_i^{\max} = q_i^{\min} = q_i^{\max} = 0$ for bus i where there is no generator, i.e. $i \in \mathcal{B}\setminus\mathcal{G}$.

One can equivalently formulate the above OPF problem in polar coordinates by replacing (1f)-(1f) with (2) [94]:

$$p_i^g - p_i^d = \quad g_{ii}|V_i|^2 + \sum_{j\in\delta(i)} p_{ij} \qquad i \in \mathcal{B} \qquad (2a)$$

$$q_i^g - q_i^d = -b_{ii}|V_i|^2 + \sum_{j\in\delta(i)} q_{ij} \qquad i \in \mathcal{B} \qquad (2b)$$

$$\underline{V}_i \le |V_i| \le \overline{V}_i \qquad i \in \mathcal{B} \qquad (2c)$$

$$p_{ij} = -G_{ij}|V_i|^2 + G_{ij}|V_i||V_j|\cos(\theta_i - \theta_j) + B_{ij}|V_i||V_j|\sin(\theta_i - \theta_j) \quad (i,j) \in \mathcal{L} \quad (2d)$$

$$q_{ij} = \quad B_{ij}|V_i|^2 - B_{ij}|V_i||V_j|\cos(\theta_i - \theta_j) + G_{ij}|V_i||V_j|\sin(\theta_i - \theta_j) \quad (i,j) \in \mathcal{L}. \quad (2e)$$

Sometimes, the rectangular formulation is preferred since the Hessian matrices of the constraints are constant and this is an advantage for the interior point methods. On the other hand, when the voltage magnitude is fixed at some buses, the polar formulation may become more advantageous [94].

1.1.3 DC Approximation

AC OPF is a difficult optimization problem due to its inherent nonlinearity and nonconvexity. Due to these challenges, the current practice in the electricity industry is to use the so-called DC OPF approximation [31]. DC OPF is a linearization of AC OPF by exploiting some physical properties of the power flows in typical power systems, such as tight bounds on voltage magnitudes at buses and small voltage angle differences between buses. However, such an approximation completely ignores important aspects of power flow physics, such as the reactive power and voltage magnitude. To partially remedy this drawback, the current practice is to solve DC OPF and then to solve a set of power flow equations with the DC OPF solution to compute feasible reactive powers and voltages. Especially for uncongested systems with small losses, DC OPF offers a reasonable approximation for AC OPF, or at least, provides good starting points for nonlinear solvers.

DC approximation exploits some physical properties of the power flows. In particular, we make the following three assumptions to obtain DC power flow equations:

(i) Resistance of transmission lines is small, i.e., $G \approx 0$.

(ii) Voltage magnitudes are fixed to be around 1 p.u., i.e., $|V| \approx 1$.

(iii) Phase angle difference between two neighboring buses is small, i.e., $\theta_i - \theta_j \approx 0$. This assumption leads to the approximation $\sin(\theta_i - \theta_j) \approx \theta_i - \theta_j$.

Now, we present the formulation for DC OPF using a subset of decision variables in AC OPF with one exception: In order to emphasize the difference between AC and DC versions of the problem, we denote the real power flow across line (i, j) as f_{ij}. In the DC model, power flow on a transmission line is proportional to the difference in phase angles of voltages at the two ends of the line with a constant coefficient B_{ij}. Since DC approximation implies that the lines are lossless, we have $f_{ij} = -f_{ji}$. Each line $(i, j) \in \mathcal{L}$ is given an (arbitrary) orientation, with the convention that power flow in the direction from $i \to j$ is positive, while power that flows along line (i, j) in the direction $j \to i$ is negative. We use the standard notation that $\delta^+(i) := \{j \in \mathcal{B} : (i, j) \in \mathcal{L}\}$ and $\delta^-(i) := \{j \in \mathcal{B} : (j, i) \in \mathcal{L}\}$. Each transmission line $(i, j) \in \mathcal{L}$ has an upper bound $\bar{f}_{ij}$ on the allowed power flow. Finally, DC OPF problem is modeled as follows [94]:

$$\min \sum_{i \in \mathcal{G}} C_i(p_i^g) \tag{3a}$$

$$\text{s.t. } p_i^g - p_i^d = \sum_{j \in \delta^+(i)} f_{ij} - \sum_{j \in \delta^-(i)} f_{ji} \qquad i \in \mathcal{B} \tag{3b}$$

$$f_{ij} = B_{ij}(\theta_i - \theta_j) \qquad (i, j) \in \mathcal{L} \tag{3c}$$

$$-\overline{f}_{ij} \leq f_{ij} \leq \overline{f}_{ij} \qquad (i, j) \in \mathcal{L}, \tag{3d}$$

(1h).

Here, constraint (3b) enforces flow conservation at each bus i. Constraint (3c) defines real power across line (i, j), namely f_{ij} in terms of the phase angles at buses i and j while constraint (3d) imposes an upper bound on power over line (i, j).

1.2 Optimal Transmission Switching

1.2.1 Introduction and Literature Survey

Transmission switching, as an emerging operational scheme, has gained considerable attention in both industry and academia in the recent years [84, 33, 42, 58, 43]. Switching on and off transmission lines, therefore, changing the network topology in the real-time operation, may bring several benefits that the traditional economic dispatch cannot offer, such as reducing the total operational cost [33, 41, 39], mitigating transmission congestion [98], clearing contingencies [57, 63], and improving do-not-exceed limits [64] .

One consequence of the underlying physical laws of electric power flow is a type of "Braess's Paradox" where removing lines from a transmission network may result in *improved* network efficiency. [84] proposes exploiting this well-known attribute of power transmission networks by switching off lines in order to reduce generation costs. In [33], this notion was formalized into a mathematical optimization problem known as the Optimal Transmission Switching (OTS) problem. The OTS problem is the OPF problem augmented with the additional flexibility of changing the network topology by removing transmission lines. While motivated in [84] by an operational problem in which lines may be switched off to improve efficiency, the same mathematical switching structure appears also in longer-term transmission network expansion planning problems.

Because of the mathematical complexity induced by the AC power flow equations, nearly all studies to date on the OTS problem have used the DC approximation to power flow (e.g. see [84, 33, 10, 88, 89, 37, 107]). With this approximation, a mixed-integer linear programming (MILP) model for DC OTS can be constructed and the resulting model can be fed into to existing MILP software. In the MILP model, binary variables are used to model the changing topology and enforce the network power flow constraints on a line if and only if the line is present. Models with many

"indicator constraints" of this form are often intractable for modern computational integer programming software, since the linear relaxations of the formulations are typically very weak. Previous authors have found this to be true for DC OTS, and many heuristic methods have been developed based on ranking lines [10, 37, 107], or by imposing an (artificial) cardinality constraint on the number of lines that may be switched off in a solution [33]. The solutions found from these heuristics have demonstrated that significant efficiency gains are possible via transmission switching.

A survey in [43] enumerates applications of transmission switching to improve voltage profiles, reduce congestion and losses in the system, and to increase reliability of the power grid. A standard reliability criterion for the power grid is that the system must be able to withstand an "$N-1$" event—in an interconnection network with N elements, the system will operate reliably following the failure of any one of them. In [42], authors augment the DC OTS model to ensure that the $N-1$ reliability criteria is satisfied. Heuristics are used to iteratively decide which lines to be switched off, while preserving $N-1$ reliability for standard test cases. The authors show that significant cost savings from transmission switching is still possible even if the $N-1$ contingency is required. Switching can be beneficial in terms of cost savings but one should also consider the reliability side. In [85], "connectivity-ensuring constraints" are developed to prevent islanding, which is not a desirable topology under normal operating conditions.

Transmission switching is also an important subproblem in power grid transmission expansion planning. In [58], a large MILP model is constructed to solve an expansion planning problem with contingencies, where the DC power flow equations are used, and transmission switching is allowed in finding the best configuration. In [103], a slightly different model for expansion planning is developed that also relies on the DC-power flow approximation and transmission switching. They employ their method on a case study for a real power system expansion plan in Denmark (see [104]).

In the context of expansion planning, it is important to note that the mathematical structure of line addition is exactly the same as line removal.

As mentioned above, previous literature on transmission switching mainly focuses on DC version of the problem and exploring fast heuristics to obtain good solutions efficiently. However, it has also been recognized that the optimal topology obtained by solving DC transmission switching is not guaranteed to be AC feasible, also it may over-estimate cost improvements and overlook stability issues [45]. [26] offers a criticism of the use of the DC-approximation to model power flow for the OTS problem. The paper demonstrates that a direct application of the DC power flow equations may *not* be accurate enough to recover useful AC operation solutions in the context of the OTS. They thus argue for the use of the AC-power flow equations in the OTS problem.

The AC optimal transmission switching problem (AC OTS) is much less explored and the limited literature is relatively new. In [45], a convex relaxation of AC OTS is proposed based on trigonometric outer-approximation. The problem is formulated as a mixed integer nonlinear program (MINLP) and solved using the solver BONMIN to obtain upper bounds. In [93], a new ranking heuristic is proposed based on the economic dispatch solutions and the corresponding dual variables. In [11], DC OTS is solved first and then a heuristic correction mechanism is utilized to restore AC feasibility of the solutions.

There are several closely related problems in the literature, which have line switching decisions similar to AC OTS, such as network configuration [51, 32], transmission system planning [50] and intentional islanding [100]. The main ideas are based on conic relaxations or piecewise linear approximations of the nonconvex power flow equations.

1.2.2 Formulation

AC OPF and DC OPF formulations introduced in Sections 1.1.2 and 1.1.3 respectively can be easily adapted to OTS by introducing binary variables x_{ij} that takes the value 1 if line $(i,j) \in \mathcal{L}$ is on, and 0 if the line is disconnected. Let us first start with AC OTS, which can be obtained by replacing constraints (1d)-(1e) in rectangular formulation (1) by

$$p_{ij} = [-G_{ij}(e_i^2 + f_i^2) + G_{ij}(e_i e_j + f_i f_j) - B_{ij}(e_i f_j - e_j f_i)]x_{ij} \tag{4a}$$

$$q_{ij} = [\ \ B_{ij}(e_i^2 + f_i^2) - B_{ij}(e_i e_j + f_i f_j) - G_{ij}(e_i f_j - e_j f_i)]x_{ij}, \tag{4b}$$

or constraints (2d)-(2e) in polar formulation by

$$p_{ij} = [-G_{ij}|V_i|^2 + G_{ij}|V_i||V_j|\cos(\theta_i - \theta_j) + B_{ij}|V_i||V_j|\sin(\theta_i - \theta_j)]x_{ij} \tag{5a}$$

$$q_{ij} = [\ B_{ij}|V_i|^2 - B_{ij}|V_i||V_j|\cos(\theta_i - \theta_j) + G_{ij}|V_i||V_j|\sin(\theta_i - \theta_j)]x_{ij}. \tag{5b}$$

On the other hand, to obtain DC OTS, we simply replace constraint (3c) in DC OPF formulation (3) by

$$f_{ij} = [B_{ij}(\theta_i - \theta_j)]x_{ij}. \tag{6}$$

This nonlinear constraint can be linearized using McCormick envelopes as in [33].

1.3 Contributions

Here, we summarize the key contributions of each chapter.

Chapter 2 is based on [61], in which we consider AC OPF problem over radial networks. It has been recently proven that the SDP relaxation of the OPF problem over radial networks is exact under technical conditions such as not including generation lower bounds or allowing load over-satisfaction. In this chapter, we investigate the situation where generation lower bounds are present. We show that even for a two-bus one-generator system, the SDP relaxation can have all possible approximation outcomes, that is (1) SDP relaxation may be exact or (2) SDP relaxation may

be inexact or (3) SDP relaxation may be feasible while the OPF instance may be infeasible. We provide a complete characterization of when these three approximation outcomes occur and an analytical expression of the resulting optimality gap for this two-bus system. In order to facilitate further research, we design a library of instances over radial networks in which the SDP relaxation has positive optimality gap. Finally, we propose valid inequalities and variable bound tightening techniques that significantly improve the computational performance of a global optimization solver. Our work demonstrates the need of developing efficient global optimization methods for the solution of OPF even in the simple but fundamental case of radial networks.

Chapter 3 is based on [60], in which we focus on the solution of AC OPF problem for the general case of meshed networks. This chapter proposes three strong SOCP relaxations for the AC OPF problem. These three relaxations are incomparable to each other and two of them are incomparable to the standard SDP relaxation of OPF. Extensive computational experiments show that these relaxations have numerous advantages over existing convex relaxations in the literature: (i) their solution quality is extremely close to that of the SDP relaxations (the best one is within 99.96% of the SDP relaxation on average for all the IEEE test cases) and consistently outperform previously proposed convex quadratic relaxations of the OPF problem, (ii) the solutions from the strong SOCP relaxations can be directly used as a warm start in a local solver such as IPOPT to obtain a high quality feasible OPF solution, and (iii) in terms of computation time, the strong SOCP relaxations can be solved an order of magnitude faster than standard SDP relaxations. Overall, the proposed strong SOCP relaxations provide a practical approach to obtain feasible OPF solutions with extremely good quality within a time framework that is compatible with the real-time operation in the current industry practice.

Chapter 4 focuses on solving the challenging OPF instances from the NESTA

archive [25] by making use of global optimization techniques including cutting planes, convex envelopes and bound tightening methods specialized for this problem. In this chapter, we first model the OPF Problem as a SDP with additional minor restrictions. A thorough analysis of the properties of the minors and submatrices of the matrix variable used in the formulation leads us to a stronger SOCP relaxation of the OPF problem than the one proposed in Chapter 3. Our approach efficiently proves stronger dual bounds than the ones obtained from SDP relaxation based approaches in the literature. We also propose an SOCP based spatial branch-and-cut method to globally solve the most challenging instances, for which our algorithm is able to prove 0.79% optimality gap in only 323 seconds on the average. Overall, our methodology provides a computationally tractable approach to prove strong dual bounds for the OPF problem.

Chapter 5 is based on [62], in which we consider the DC OTS problem. Most research on DC OTS has focused on heuristic algorithms for generating quality solutions or on the application of DC OTS to crucial operational and strategic problems such as contingency correction, real-time dispatch, and transmission expansion. The mathematical theory of the DC OTS problem is less well-developed. In this chapter, we formally establish that DC OTS is NP-Hard, even if the power network is a series-parallel graph with at most one load/demand pair. Inspired by Kirchoff's Voltage Law, we give a cycle-based formulation for DC OTS, and we use the new formulation to build a cycle-induced relaxation. We characterize the convex hull of the cycle-induced relaxation, and the characterization provides strong valid inequalities that can be used in a cutting-plane approach to solve the DC OTS. We give details of a practical implementation, and show promising computational results on standard benchmark instances.

Chapter 6 is based on [59], in which we focus on the AC OTS problem. Most recent research has relied on the DC approximation of the power flow model in the

OTS problem as in Chapter 5. However, it is known that DC approximation may lead to inaccurate flow solutions and also overlook stability issues. In this chapter, we focus on the OTS problem with the full AC power flow model. We propose a new exact formulation for AC OTS and its mixed-integer second-order conic programming (MISOCP) relaxation. We improve this relaxation via several types of strong valid inequalities inspired by the developments for the closely related AC OPF problem in Chapter 3. We also propose a practical algorithm to obtain high quality feasible solutions for the AC OTS problem. Extensive computational experiments show that the proposed formulation and algorithms efficiently solve IEEE standard and congested instances and lead to significant cost benefits with provably tight bounds.

CHAPTER II

OPTIMAL POWER FLOW OVER RADIAL NETWORKS

2.1 Introduction

In this chapter, we focus on the OPF problem on radial networks in the presence of generation lower bounds on both real and reactive power. The goal of this chapter is two-fold: To highlight the inexactness of standard convex relaxations for these instances and to make algorithmic progress in solving such instances globally. We make two comments here in relation to the class of OPF problems we consider and our assumptions. First, although most power flow networks are not radial, they are usually quite sparse and analyzing radial networks can therefore be beneficial in their own right, especially in the case of distribution networks [66]. Second, typically power systems have ramping constraints, so that the power generation in the next time period cannot deviate too much from the current one. Hence, it is important to make a study of the effects of lower bounds.

In practice, SDPs may become prohibitively expensive as the size of the network grows larger. One can turn to SOCP relaxations, which are in general weaker than their SDP counterparts. However, in [92], it has been proven that both types of relaxations give the same lower bound for the OPF problem over radial networks even if they are inexact. Therefore any result stated for SOCPs in this chapter holds for SDP relaxations and vice-versa.

The rest of this chapter is organized as follows: In the next section, we revisit the rectangular formulation of the OPF problem and propose a reformulation that leads to the SOCP relaxation. In Section 2.3, we begin working on our first goal by providing a complete characterization of the approximation performance of SOCP relaxation

for a two-bus system. In Section 2.4, we further study the feasible regions of two small systems. Then, in Section 2.5, we begin working on our second goal by providing a library of radial network instances generated from MATPOWER test cases for which SOCP relaxation is inexact. In Section 2.6, we propose valid inequalities for the SOCP relaxation, which significantly improve the computational performance of a global solver. Concluding remarks are made in Section 2.7.

2.2 OPF Formulation Revisited

In Section 1.1.2, we introduced the mathematical programming formulation of the OPF problem. In this section, we first revisit the rectangular formulation and then, propose an alternative formulation.

2.2.1 Rectangular Formulation

Let us first omit transmission line limit constraint (1g) for the brevity of discussion[1] and project out flow variables on lines, namely p_{ij} and q_{ij} to obtain the rectangular formulation:

$$\min \sum_{i \in \mathcal{G}} C_i(p_i^g) \tag{7a}$$

$$\text{s.t. } p_i^g - p_i^d = G_{ii}(e_i^2 + f_i^2) + \sum_{j \in \delta(i)} [G_{ij}(e_i e_j + f_i f_j) - B_{ij}(e_i f_j - e_j f_i)] \quad i \in \mathcal{B} \tag{7b}$$

$$q_i^g - q_i^d = -B_{ii}(e_i^2 + f_i^2) - \sum_{j \in \delta(i)} [B_{ij}(e_i e_j + f_i f_j) + G_{ij}(e_i f_j - e_j f_i)] \quad i \in \mathcal{B}, \tag{7c}$$

$$(1\text{f}), (1\text{h}) - (1\text{i}).$$

2.2.2 Alternative Formulation

Note that the rectangular formulation (7) is a nonconvex quadratic optimization problem. We can observe that all the nonlinearities in (1) are of the following three

[1] These constraints are included in our computational experiments.

types:

$$(1)\ e_i^2 + f_i^2 \quad (2)\ e_i e_j + f_i f_j \quad (3)\ e_i f_j - e_j f_i, \tag{8}$$

which are equal to $|V_i|^2$, $|V_i||V_j|\cos(\theta_i - \theta_j)$, and $-|V_i||V_j|\sin(\theta_i - \theta_j)$ in the polar form, respectively. Let us define new variables c_{ii}, c_{ij}, and s_{ij} for each of these three quantities. Since the cosine function is even and the sine function is odd, we also have $c_{ij} = c_{ji}$ and $s_{ij} = -s_{ji}$. On each line (i, j), these quantities are linked through the fundamental trigonometric identity $\cos^2(\theta_i - \theta_j) + \sin^2(\theta_i - \theta_j) = 1$, which translates into

$$(e_i e_j + f_i f_j)^2 + (e_i f_j - e_j f_i)^2 = (e_i^2 + f_i^2)(e_j^2 + f_j^2) \tag{9}$$

in the rectangular form. In the space of our new variables, this relation is expressed in the following quadratic equation $c_{ij}^2 + s_{ij}^2 = c_{ii} c_{jj}$, which describes the surface of a rotated second-order cone in four dimensions. With a change of variables, we can introduce an alternative formulation of the OPF problem as follows:

$$\min \ \sum_{i \in \mathcal{G}} C_i(p_i^g) \tag{10a}$$

$$\text{s.t. } p_i^g - p_i^d = G_{ii} c_{ii} + \sum_{j \in \delta(i)} (G_{ij} c_{ij} - B_{ij} s_{ij}) \qquad i \in \mathcal{B} \tag{10b}$$

$$q_i^g - q_i^d = -B_{ii} c_{ii} + \sum_{j \in \delta(i)} (-B_{ij} c_{ij} - G_{ij} s_{ij}) \qquad i \in \mathcal{B} \tag{10c}$$

$$\underline{V}_i^2 \le c_{ii} \le \overline{V}_i^2 \qquad i \in \mathcal{B} \tag{10d}$$

$$c_{ij} = c_{ji}, \quad s_{ij} = -s_{ji} \qquad (i, j) \in \mathcal{L} \tag{10e}$$

$$c_{ij}^2 + s_{ij}^2 = c_{ii} c_{jj} \qquad (i, j) \in \mathcal{L}, \tag{10f}$$

$$(1\text{h}) - (1\text{i}).$$

This formulation was first introduced in [30] and [52]. It is an exact formulation for OPF on a radial network in the sense that the optimal power output of (10) is also

optimal for (7) and one can always recover the voltage phase angles θ_i's by solving the following system of linear equations with the optimal solution c_{ij}, s_{ij}:

$$\theta_j - \theta_i = \text{atan2}(s_{ij}, c_{ij}) \quad (i,j) \in \mathcal{L}, \tag{11}$$

which then provide an optimal voltage phasor solution to (7) (see e.g. [110]). Here, in order to cover the entire range of 2π, we use the atan2(y, x) function[2], which takes value in $(-\pi, \pi]$, rather than $[-\pi/2, \pi/2]$ as is the case of the regular arctangent function. Unfortunately, for meshed networks, the above formulation (10) can be a strict relaxation of the OPF problem. The reason is that, given an optimal solution c_{ij}, s_{ij} for all edges (i, j) of (10), it does not guarantee that atan2(s_{ij}, c_{ij}) sums to zero over all cycles. In other words, the optimal solution of (10) may not be feasible for the original OPF problem (7). This issue can be fixed by directly imposing (11) as a constraint [53, 54]. Thus (10) together with (11) is a valid formulation for OPF in mesh networks. Note that the constraints involving the atan2 function are nonconvex.

2.2.3 SOCP Relaxation of Alternative Formulation

Since we focus on radial networks in this chapter, alternative formulation (10) is exact. Now, we consider its SOCP relaxation. Observe that except the coupling constraints (10f), all other constraints in (10) are linear. Hence, all the nonconvexity of the OPF problem (7) in a radial network is captured by (10f), and the feasible region is the intersection of a polytope defined by (10b)-(10e), (1h)-(1i) with the boundaries of rotated second-order cones defined by (10f). It is straightforward to obtain a SOCP

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

relaxation of (10) by relaxing constraint (10f) as follows:

$$c_{ij}^2 + s_{ij}^2 \leq c_{ii}c_{jj} \quad (i,j) \in \mathcal{L}, \tag{12}$$

This constraint can be written more explicitly as a SOCP constraint:

$$c_{ij}^2 + s_{ij}^2 + \left(\frac{c_{ii} - c_{jj}}{2}\right)^2 \leq \left(\frac{c_{ii} + c_{jj}}{2}\right)^2 \quad (i,j) \in \mathcal{L}. \tag{13}$$

The SOCP relaxation is defined as (10a)-(10e), (1h)-(1i) and (13). It is proven that in radial networks, the SOCP relaxation is equivalent to the SDP relaxation [92]. In this chapter, we focus on SOCP relaxation due to its superior computational performance.

For a more general discussion on the pairwise strength of different types of relaxations for rectangular and alternative formulation over general networks, readers are referred to Section 3.2.

2.3 Analytical Study of a Two-Bus System

In this section, we study the two-bus system with one generator and one load. This is arguably the simplest power system, but also one of the most fundamental models in power system analysis. Surprisingly, for this simple system, the SOCP relaxation with generation lower bounds can have all three possible outcomes in terms of optimality gap, namely (1) SOCP obtains exact solution (i.e. optimality gap is zero); (2) SOCP is feasible, yet OPF is infeasible (optimality gap is infinite); (3) SOCP has a finite optimality gap, and we give an analytical expression of this gap. We identify parameter ranges in closed form for each of these outcomes. We also study the feasible region projected in the space of squared bus voltage magnitudes to gain geometric intuition.

Let us assume that bus 1 is a generator bus and bus 2 is a load bus. Further assume that $g_{ii} = b_{ii} = 0$ and $G := G_{12} < 0$ and $B := B_{12} > 0$ (the analysis for $B < 0$ is similar). Also assume the production cost $C_1(p_1^g)$ is linear in p_1^g.

2.3.1 Feasible Region Projected to (c_{11}, c_{22}) space

The linear equality system (10b)-(10c) can be written as

$$\begin{bmatrix} 1 & & G & & -G & B \\ & 1 & -B & & B & G \\ & & & G & -G & -B \\ & & & -B & B & -G \end{bmatrix} \begin{bmatrix} p_1^g \\ q_1^g \\ c_{11} \\ c_{22} \\ c_{12} \\ s_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ p_2^d \\ q_2^d \end{bmatrix}. \tag{14}$$

Let us define

$$\alpha := \frac{Bp_2^d + Gq_2^d}{B^2 + G^2} \quad \text{and} \quad \beta := \frac{Gp_2^d - Bq_2^d}{B^2 + G^2}, \tag{15}$$

which are constant for fixed B, G and load. Solving the linear system (14), we can express $(p_1^g, q_1^g, c_{12}, s_{12})$ in terms of (c_{11}, c_{22}) as follows

$$s_{12} = -\alpha \tag{16a}$$

$$c_{12} = c_{22} - \beta \tag{16b}$$

$$p_1^g = -G(c_{11} - c_{22}) - G\beta + B\alpha \tag{16c}$$

$$q_1^g = B(c_{11} - c_{22}) + B\beta + G\alpha. \tag{16d}$$

We now reformulate constraint (10f) using (16a) and (16b) as

$$(c_{22} - \beta)^2 + \alpha^2 = c_{11}c_{22} \Rightarrow c_{11} = c_{22} - 2\beta + \frac{\alpha^2 + \beta^2}{c_{22}}, \tag{17}$$

which defines a hyperbola for (c_{11}, c_{22}) with two asymptotes: $c_{11} - c_{22} = -2\beta$ and $c_{22} = 0$.

Observe that this hyperbola together with the constraints on c_{11} and c_{22} implied from (10d), (1h)-(1i) define the feasible region of the OPF problem projected to the (c_{11}, c_{22}) space. In particular, (10d) impose a box constraint on c_{11} and c_{22}, whereas

(1h)-(1i) imply upper and lower bounds on the difference $c_{11} - c_{22}$, which defines a region parallel to the first asymptote $c_{11} - c_{22} = -2\beta$. Figure 2a depicts the entire feasible regions of OPF in black curve and of SOCP relaxation in the blue region. Figures 2b-2e zoom in particular parts.

Furthermore, since the objective function $C_1(p_1^g)$ is assumed to be linear in p_1^g and by (16c), we can see that the level set of the objective function in (c_{11}, c_{22}) is also parallel to the first asymptote, and decreases toward the upper left corner as pointed by the arrow in Figure 2. Therefore, only the lower bounds on p_1^g and q_1^g can affect the optimal solution of (10). For this reason, we find the *effective* lower bound for the difference $c_{11} - c_{22}$ as

$$\Delta = \max\left\{\frac{p_1^{\min} + G\beta - B\alpha}{-G}, \frac{q_1^{\min} - B\beta - G\alpha}{B}\right\}, \tag{18}$$

which is given by the lower bounds of (1h)-(1i), and is plotted as magenta lines in Figure 2. Also note that as $p_1^{\min}$ and $q_1^{\min}$ increase, the line $c_{11} - c_{22} \geq \Delta$ moves toward the lower right corner in Figure 2.

2.3.2 Complete Characterization of Approximation Outcomes

At this point, we are ready to explore the optimal solutions of the OPF (10) and its SOCP relaxation and classify all five possible cases of the configurations of their feasible regions and the associated approximation outcomes.

- First of all, let us assume that Δ defined in (18) is small enough. In this case, as depicted in Figure 2a, the optimal solution of both the OPF and the SOCP is unique and given by

$$(c_{11}^O, c_{22}^O) = \begin{cases} \left(\overline{c}_{22} - 2\beta + \frac{\alpha^2+\beta^2}{\overline{c}_{22}}, \overline{c}_{22}\right) & \text{if (a) holds} \\ \left(\overline{c}_{11}, \frac{2\beta+\overline{c}_{11}+\sqrt{(2\beta+\overline{c}_{11})^2-4(\alpha^2+\beta^2)}}{2}\right) & \text{o.w.} \end{cases} \tag{19}$$

  where condition (a) is $(\overline{c}_{22} - \beta)^2 + \alpha^2 \leq \overline{c}_{11}\overline{c}_{22}$ and $\overline{c}_{ii} := \overline{V}_i^2$. Hence, the SOCP relaxation is exact. This result is in accordance with the results in [110].

- Consider the case where Δ is large enough. In particular, $c^O_{11} - c^O_{22} < \Delta$. Define the intersection of $c_{11} - c_{22} = \Delta$ with the binding upper bound of either c_{11} or c_{22} as

$$(c^R_{11}, c^R_{22}) = \begin{cases} (\overline{c}_{22} + \Delta, \overline{c}_{22}) \text{ if } \overline{c}_{11} - \overline{c}_{22} \geq \Delta \\ (\overline{c}_{11}, \overline{c}_{11} - \Delta) \text{ o.w.} \end{cases} \tag{20}$$

Note that this point is OPF infeasible despite being SOCP optimal. Next, define the intersection of the hyperbola (17) and $c_{11} - c_{22} = \Delta$ as

$$(c^E_{11}, c^E_{22}) = \left(\frac{\alpha^2 + \beta^2}{2\beta + \Delta} + \Delta, \frac{\alpha^2 + \beta^2}{2\beta + \Delta} \right). \tag{21}$$

If $c^E_{22} < \underline{c}_{22}$, where $\underline{c}_{ii} := \underline{V}_i^2$, then SOCP is feasible while OPF is infeasible. An example of this case can be seen from Figure 2b, which shows the zoomed in part of the hyperbola.

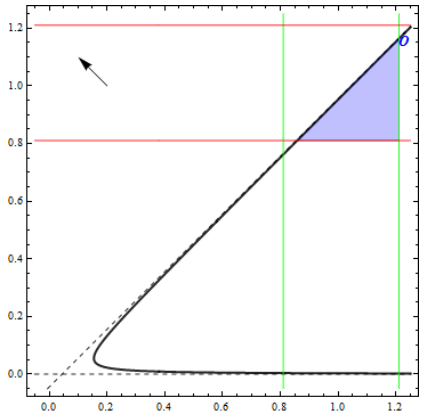
- If $c^O_{11} - c^O_{22} < \Delta$, $c^E_{22} \geq \underline{c}_{22}$, and $c^E_{11} \geq \underline{c}_{11}$, then the SOCP relaxation is exact as in Figure 2c. In fact, any point in the convex combination of c^R and c^E is SOCP optimal. Such a point can always be corrected by reducing c_{11}, c_{22} and c_{12} components by the same amount until we reach c^E, which is the OPF optimal solution.

- If $c^O_{11} - c^O_{22} < \Delta$, $c^E_{22} \geq \underline{c}_{22}$, and $c^E_{11} < \underline{c}_{11}$, define the intersection of $c_{11} - c_{22} = \Delta$ with the bounding lower bound of either c_{11} or c_{22}:

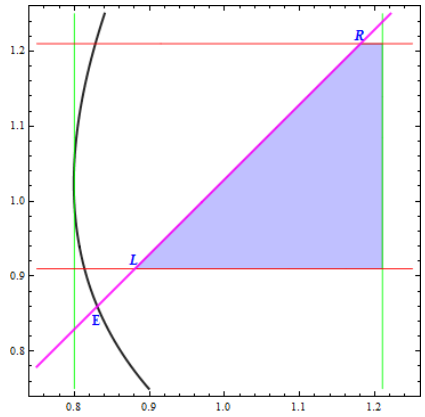
$$(c^L_{11}, c^L_{22}) = \begin{cases} (\underline{c}_{22} + \Delta, \underline{c}_{22}) \text{ if } \underline{c}_{11} - \underline{c}_{22} \leq \Delta \\ (\underline{c}_{11}, \underline{c}_{11} - \Delta) \text{ o.w.} \end{cases} \tag{22}$$

Observe that any point in the convex combination of c^L and c^R is SOCP optimal. However, there is no feasible OPF solution with the same objective function value. Lastly, let us define the lower intersection of the hyperbola (17) and the c_{11} lower bound as (c^I_{11}, c^I_{22}), where $c^I_{11} = \underline{c}_{11}$, and c^I_{22} as

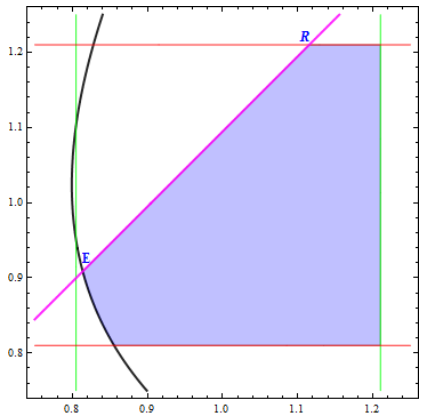
$$c^I_{22} = \frac{2\beta + \underline{c}_{11} - \sqrt{(2\beta + \underline{c}_{11})^2 - 4(\alpha^2 + \beta^2)}}{2}. \tag{23}$$



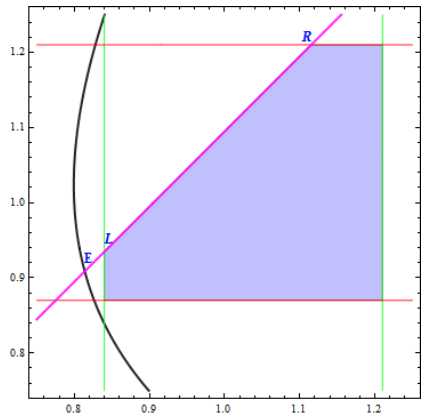
(a) Case 1: SOCP relaxation is exact.



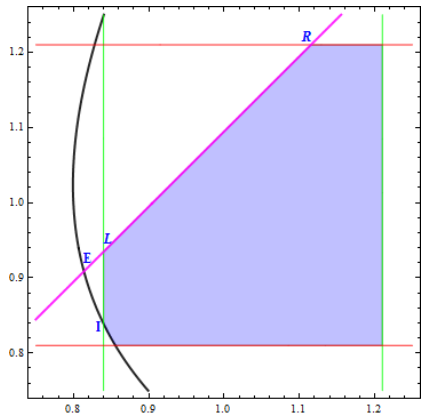
(b) Case 2: SOCP relaxation is feasible, OPF is infeasible.



(c) Case 3: SOCP relaxation is exact.



(d) Case 4: SOCP relaxation is feasible, OPF is infeasible.



(e) Case 5: SOCP relaxation is inexact.

Figure 2: Projection of feasible region of 2-bus, 1-generator examples onto (c_{11}, c_{22}) space for five cases. Horizontal axis is c_{11} and vertical axis is c_{22}. Solid black curve is (17) containing the feasible region of OPF with dashed lines being two asymptotes shown in Fig. 1a. Green and red lines are bounds on c_{11} and c_{22}, resp. Magenta line is the effective lower bound on $c_{11} - c_{22}$. Blue region is the feasible region of SOCP relaxation. All figures are in p.u.

We have two final cases:

- If $c_{22}^I < \underline{c}_{22}$, any point in the convex combination of c^L and c^R is SOCP optimal. However, OPF is infeasible. An example of this case can be seen in Figure 2d.
- If $c_{22}^I \geq \underline{c}_{22}$, any point in the convex combination of c^L and c^R is SOCP optimal. However, OPF has a unique optimal solution at c^I as can be seen in Figure 2e. Hence, relaxation is inexact. Assuming a linear cost function with coefficient 1, optimality gap can be calculated as $-G(c_{22}^L - c_{22}^I)$.

The above analysis proves the following theorem.

Theorem 1. *In a two-bus one-generator system with linear objective, the SOCP/SDP relaxation of the AC OPF problem has the following possible outcomes:*

(i) SOCP relaxation is exact: If $c_{11}^O - c_{22}^O \geq \Delta$ *or if* $c_{11}^O - c_{22}^O < \Delta$, $c_{22}^E \geq \underline{c}_{22}$, $c_{11}^E \geq \underline{c}_{11}$.

(ii) SOCP relaxation is inexact with finite optimality gap: If $c_{11}^O - c_{22}^O < \Delta$, $c_{22}^E \geq \underline{c}_{22}$, $c_{11}^E < \underline{c}_{11}$, $c_{22}^I \geq \underline{c}_{22}$. *The optimality gap is* $-G(c_{22}^L - c_{22}^I)$.

(iii) SOCP relaxation is feasible and OPF is infeasible: If $c_{11}^O - c_{22}^O < \Delta$, $c_{22}^E < \underline{c}_{22}$ *or if* $c_{11}^O - c_{22}^O < \Delta$, $c_{22}^E \geq \underline{c}_{22}$, $c_{11}^E < \underline{c}_{11}$, $c_{22}^I < \underline{c}_{22}$.

Here, c^O, c^E, c^R, c^L, c^I *are defined in* (19)-(23)*, respectively.*

2.4 Examples of Inexact SOCP Relaxations

We have obtained a complete characterization for a 2-bus network with a single generator, and shown that the SOCP relaxation is exact only under certain conditions. In this section, we present further counterexamples of radial networks with two and three buses. Most of the network parameters are selected from IEEE test instances.

Transmission line capacity is assumed to be large. For all the buses, $\underline{V}_i = 0.9$ and $\overline{V}_i = 1.1$. Production costs are taken as linear functions. OPF problem with alternative formulation (10) is solved to global optimality with BARON [95]. SOCP relaxations are solved using interior point solver MOSEK [2].

2.4.1 2-Bus, 2-Generator Example

Let us consider a 2-bus network with one generator located at each bus. Data of this example is given in Table 1. The impedance of line (1,2) is $0.01008 + \mathrm{i}0.0504$.

Table 1: Bus and generator data for 2-bus 2-generator example.

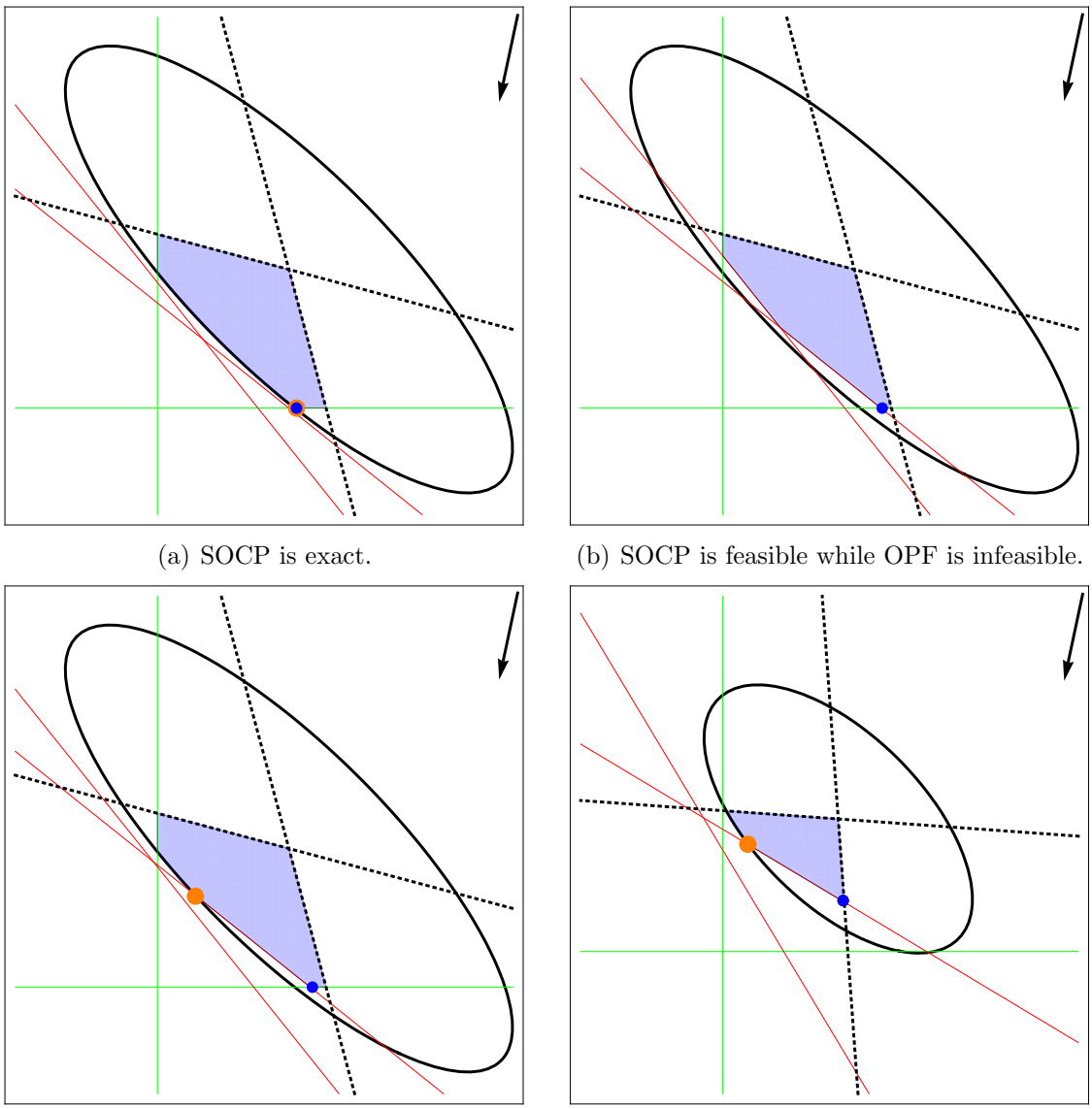
Bus	p_i^d	q_i^d	$p_i^{\min}$	$p_i^{\max}$	$q_i^{\min}$	$q_i^{\max}$	cost
1	75	-84.7	75	250	-30	300	5.0
2	105	22.8	70	300	-30	300	1.2

In Table 2, we compare the SOCP relaxation and the global optimal solution of OPF for different levels of load, where load is varied as $[p_1^d \;\; p_2^d \;\; q_1^d \;\; q_2^d] = \gamma \cdot [75 \;\; 105 \;\; -84.7 \;\; 22.8]$ for some positive parameter γ.

Table 2: Objective costs for 2-bus 2-generator with varying load.

γ	OPF	SOCP
0.12	infeasible	infeasible
0.13	infeasible	459.00
0.80	infeasible	459.00
0.81	460.13	460.13
0.98	496.96	496.96
0.99	499.15	499.15
1.00	563.56	501.46
1.01	641.21	503.76
1.02	infeasible	506.07
2.92	infeasible	1608.75
2.93	infeasible	infeasible

When $\gamma \in [0.81, 0.99]$, we observe that the SOCP relaxation is exact. For γ around 1.00, there is a finite optimality gap, which can be as large as 21.44% at



(a) SOCP is exact.

(b) SOCP is feasible while OPF is infeasible.

(c) SOCP is inexact due to reactive and active lower bounds.

(d) SOCP is inexact due to angle and reactive lower bounds.

Figure 3: Projection of feasible region of 2-bus, 2-generator example onto (p_1^g, p_2^g) space. Horizontal axis is p_1^g and vertical axis is p_2^g. Black curve is an ellipse with counterclockwise orientation that contains the feasible region of OPF problem whereas blue region is its SOCP relaxation. Green lines are the lower bound on p_1^g and p_2^g while red lines are the lower bound on q_1^g and q_2^g. Dashed lines represent angle bounds corresponding to 30°. Assuming linear functions, the arrow shows the cost vector. Blue and orange dots are respectively the optimal solutions of SOCP relaxation and OPF, whenever the latter exists.

$\gamma = 1.01$. Finally, for $\gamma \geq 1.02$, OPF becomes infeasible, whereas SOCP relaxation is still feasible. In fact, SOCP relaxation fails to detect infeasibility of OPF problem until γ exceeds 2.93.

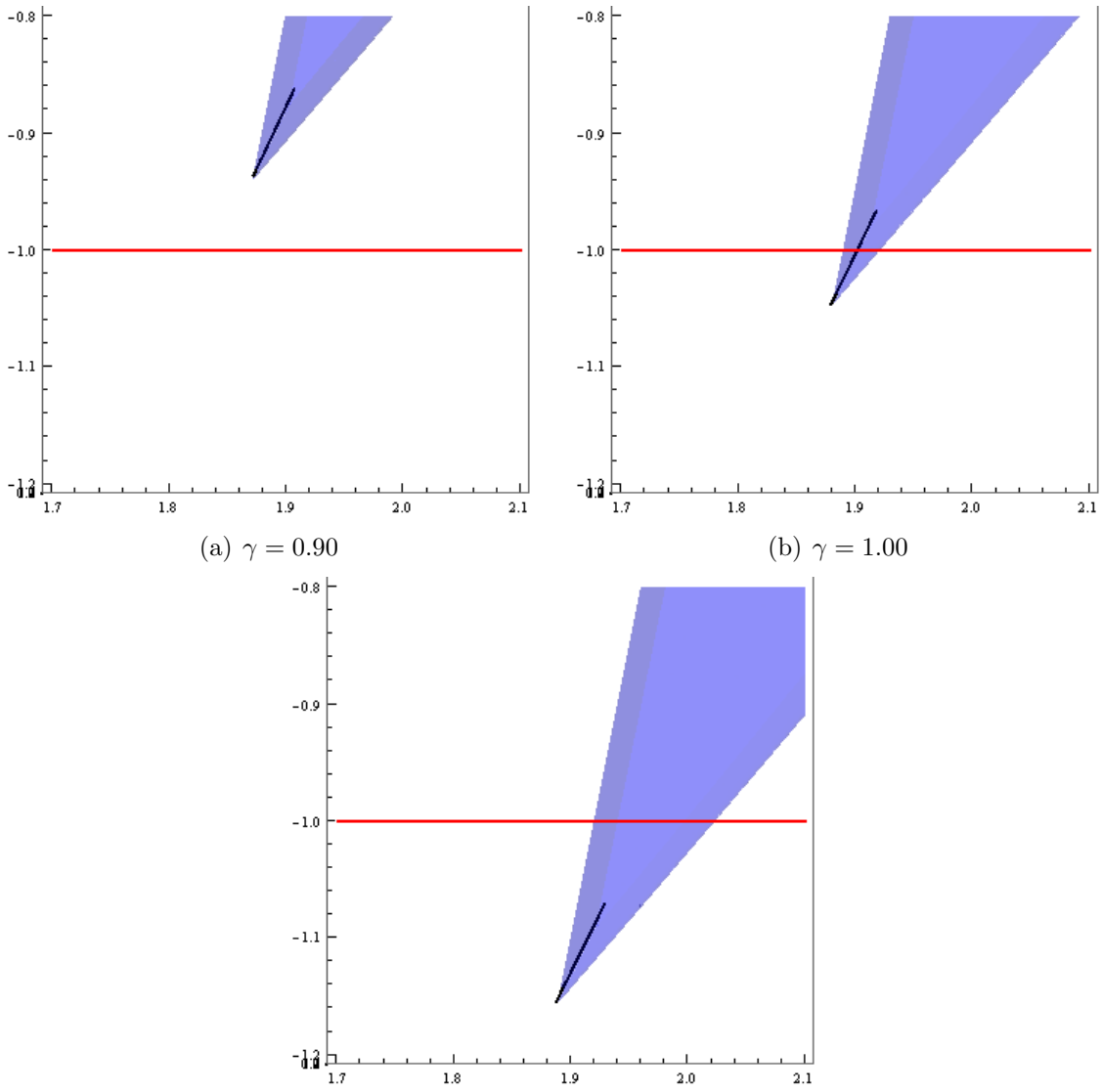
Now, let us consider the case where voltages are fixed. In [66], it has been proven that if angle differences are guaranteed to be small enough, then SDP/SOCP relaxations are tight even if there are real power lower bounds. However, we present an example which demonstrates that this does not extend to the case with reactive power lower bounds. To this end, let us fix the squared voltage magnitudes to $(c_{11}, c_{22}) = (0.874, 0.816)$. In this case, the global optimal solution of OPF is 573.82 while the SOCP relaxation gives 503.37. Hence, there is an optimality gap, even though angle difference is less than 1°.

Figure 3 presents possible configurations of the feasible region of the OPF and the SOCP relaxation projected to the (p_1^g, p_2^g) space. In Figure 3a, the SOCP relaxation is exact, while in Figure 3b, the OPF is infeasible although the SOCP is feasible. In Figure 3c, the SOCP relaxation is inexact due to the combined effect of active and reactive lower bounds. Finally, in Figure 3d, the SOCP relaxation is inexact due to practical angle bounds (30°) and reactive lower bounds.

2.4.2 3-Bus, 1-Generator Example

Consider a 3-bus radial network with three loads $[p_1^d \quad p_2^d \quad p_3^d] = [50 \quad 70 \quad 60]$ and $[q_1^d \quad q_2^d \quad q_3^d] = [-52.3 \quad 14.1 \quad -82.3]$. The impedance of lines (1,2) and (2,3) are $0.01008 + \mathrm{i}0.0504$ and $0.07500 + \mathrm{i}0.0840$. The only generator is located at bus 1 with $150 \leq p_1^g \leq 550$ and $-100 \leq q_1^g \leq 500$. The cost of power generation is \$5 per MW. Assume that the reactive load is scaled as $[q_1^d \;\; q_2^d \;\; q_3^d] = \gamma[-52.3 \;\; 14.1 \;\; -82.3]$ for some positive γ. Table 3 shows the optimal costs of the OPF and the SOCP relaxation for different values of γ.

For small values of γ, e.g. $\gamma \leq 0.96$, SOCP is exact. For values around $\gamma = 1$,



(a) $\gamma = 0.90$

(b) $\gamma = 1.00$

(c) $\gamma = 1.10$

Figure 4: Projection of feasible region of 3-bus example onto (p_1^g, q_1^g) space with respect to different load levels. Horizontal axis is p_1^g and vertical axis is q_1^g. Black curve and blue region are the feasible regions of OPF and SOCP relaxation, respectively. Red line is the lower bound on q_1^g. All figures are in p.u.

Table 3: Objective costs for 3-bus example with varying load.

γ	OPF	SOCP
0.95	939.45	939.45
0.96	939.90	939.90
0.97	941.57	940.87
1.00	950.70	945.45
1.03	959.91	950.05
1.04	infeasible	951.60

we observe a finite optimality gap between OPF and SOCP, where for larger values of $\gamma \geq 1.04$, the OPF becomes infeasible while the SOCP relaxation is still feasible. The infeasibility is exactly caused by the lower bound on reactive generation power.

For this example, we also give the feasible region of OPF problem projected onto the (p_1^g, q_1^g) space in Figure 4. When $\gamma = 0.90$, reactive power lower bound is redundant and the optimal solution of SOCP relaxation is feasible for OPF. However, for $\gamma = 1.00$, constraint $q_1^{\min} \leq q_1^g$ is binding. Note that the optimal solution of the SOCP relaxation is not feasible for OPF and hence, the relaxation is not exact. Finally, when $\gamma = 1.10$, SOCP is feasible whereas OPF is infeasible.

2.5 Library of Radial Networks with Inexact SDP/SOCP Relaxation

2.5.1 Generation of Instances

To facilitate further research, we generate several radial network instances from meshed networks in MATPOWER [111]. Given a meshed network, we first find a spanning tree by switching off lines to obtain a radial network. Then, only load values and generation lower/upper bounds are changed, whenever necessary, to construct examples where the SOCP/SDP relaxation is not exact. New instances can be downloaded from `https://sites.google.com/site/burakkocuk/research`.

Our examples are based on 9-, 14-, 30-, 39- and 57-bus standard instances. Due to our construction of the network topology, AC feasibility becomes a major issue.

Although unrealistic examples can be constructed for even larger networks by reducing load values considerably, we choose not to sacrifice the realistic features of the instances.

2.5.2 Computational Results for SDP Relaxation vs. Global Optimal Solution

For each instance generated as described above, we solve the SDP relaxation using MOSEK [2]. The code is written in C# language and Visual Studio 2010 is used as the compiler. We report the value of the objective function, computation time and the rank of the solution. Here, rank is determined as the number of eigenvalues that are larger than 10^{-5}.

SDP relaxation is compared against global optimal solution found using BARON [95] and local solution found by MATPOWER [111] and IPOPT [105]. Relative optimality gap for BARON is set to 0 so that global optimality can be certified. We should note that performance of BARON on rectangular formulation (1) is very poor as it requires hours to prove global optimality. Instead, we use reformulation (10), which is valid for radial networks.

For all experiments, we used a 64-bit computer with Intel Core i5 CPU 3.33GHz processor and 4 GB RAM. Each instance is solved twice with quadratic and linear objectives. For the latter, we simply ignore the quadratic cost coefficients.

Our findings are summarized in Table 4. One can see that the SDP relaxation solution can be of high-rank (up to 12 for case_ieee30 and 20 for case57). Also, the optimality gap (column "% gap") computed as $100 \times (1 - z^{SDP}/z^{BARON})$, where z^{SDP} and z^{BARON} are respectively the values of the SDP relaxation and the global optimal solution found by BARON, can be quite large (more than 52% for case9 with quadratic objective). Our examples clearly show that the optimal value of the SDP relaxation can be quite different from the global optimal value. We also compare the optimal dispatch solutions p^{SDP} and p^{BARON} computed by the SDP relaxation and BARON

Table 4: SDP relaxation vs. global solver BARON vs. local solvers MATPOWER and IPOPT.

		SDP Relaxation			BARON			MATPOWER		IPOPT	
based on	type	objective	time(s)	rank	objective	time(s)	% gap	objective	time(s)	objective	time(s)
case9	quadratic	5335.70	0.04	8	11277.95	1.17	52.69	-	0.17	-	0.17
case9	linear	1481.93	0.06	8	1756.47	1.11	15.63	-	0.08	-	0.20
case9Q	quadratic	10835.70	0.04	8	16778.87	1.36	35.42	-	0.08	16779.48	0.31
case14	quadratic	11861.87	0.07	8	11932.07	35.32	0.59	11932.25	0.11	11932.25	0.28
case14	linear	9892.70	0.09	4	9952.42	0.79	0.60	9952.59	0.09	9952.58	0.23
case_ieee30	quadratic	4244.53	0.17	12	4336.03	8347.79	2.11	-	0.12	4794.32	0.15
case_ieee30	linear	3035.61	0.22	12	3606.91	2494.31	15.84	-	0.09	4562.26	0.14
case30	quadratic	607.72	0.15	8	619.01	2.52	1.82	619.04	0.09	619.04	0.23
case30	linear	435.58	0.23	6	445.83	8.50	2.30	445.84	0.11	445.84	0.14
case30Q	quadratic	676.88	0.20	4	690.06	5.16	1.91	690.08	0.11	690.08	0.36
case39	quadratic	44869.01	0.29	4	45035.32	110.59	0.37	-	0.14	45037.05	0.27
case39	linear	1900.09	0.36	4	1903.07	1566.88	0.16	-	0.16	1903.14	0.15
case57	quadratic	10458.06	0.92	20	12100.00	> 10800	13.57	12100.90	0.15	12100.86	0.27
case57	linear	8399.82	0.96	20	10173.10	> 10800	17.43	10173.00	0.16	10172.98	0.26

to show that large differences in the objective function values are not artifacts of the cost parameters. In fact, the 2-norm $\|p^{SDP} - p^{BARON}\|$ is large, varying from 0.16 p.u. to 3.16 p.u. for our instances. This illustrates that the optimal solutions are quite different from one another.

In general, MATPOWER is accepted to be a reliable and efficient OPF solver. It manages to find the global optimal solution up to a negligible difference for seven of the instances from our library. However, we observe that it fails to solve the remaining seven instances due to numerical issues. There are other robust NLP solvers available, e.g. IPOPT, which gives near global optimal solution for nine instances in the library, where small discrepancies in optimal objective function values compared to BARON are due to numerical errors. On the other hand, IPOPT fails in two 9-bus examples and it finds suboptimal solutions three times for both of the case_ieee30 instances and case39 with quadratic objective.

We should note that the global solver BARON can be computationally expensive. For instance, for case_ieee30 with a quadratic objective, it requires more than 2 hours to prove optimality whereas for 57-bus instances, BARON is not able to certify the global optimal solution within 3 hours time limit. Upon termination, the optimality gaps are 38.40% and 29.37% for quadratic and linear objectives, respectively. Also, the reformulation of OPF (10) is only valid for radial networks. Hence, in general, using BARON as it is may not be applicable to large-scale OPFs.

2.6 Bound Tightening and Valid Inequalities for Global Optimization

In this section, we propose valid inequalities for the SOCP relaxation of the OPF problem to improve the computational time of the global solver BARON. The main algorithm of BARON is based on spatial branch-and-bound [95]. It utilizes convex envelopes of the feasible region and polyhedral relaxations to improve lower bounds and prove global optimality. Therefore, it is very important to add valid inequalities

and variable bounds so that BARON can obtain tighter relaxations.

To begin with, let us focus on formulation (10). Observe that c_{ij} and s_{ij} do not have explicit variable bounds although they have implied bounds due to (10d) and (10f) as

$$-\overline{V}_i\overline{V}_j \leq c_{ij}, s_{ij} \leq \overline{V}_i\overline{V}_j \quad (i,j) \in \mathcal{L}. \tag{24}$$

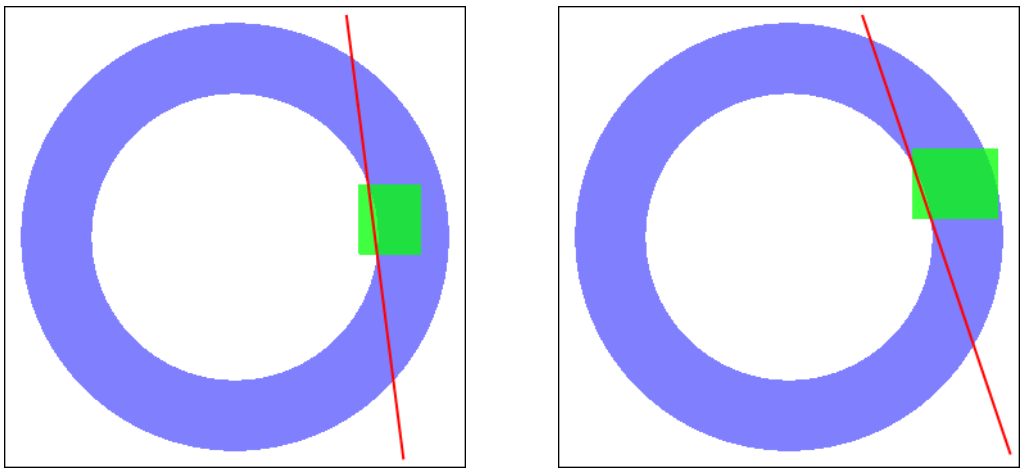
However, these bounds are very loose knowing that angle differences are generally small. This fact suggests that these bounds can be improved. One way to obtain variable bounds is to optimize c_{ij} and s_{ij} over the set $\mathcal{S} = \{(p,q,c,s) : (10b) - (10f)\}$, which is a nonconvex set. Alternatively, one can find weaker bounds over the set $\mathcal{S}' = \{(p,q,c,s) : (10b) - (10e), (10f)\}$ by solving SOCP relaxations. Let $\underline{c}_{ij}$ ($\underline{s}_{ij}$) and $\overline{c}_{ij}$ ($\overline{s}_{ij}$) denote lower and upper bounds found for c_{ij} (s_{ij}), respectively.

Now, let us investigate how the box $\mathcal{B}_{ij} = [\underline{c}_{ij}, \overline{c}_{ij}] \times [\underline{s}_{ij}, \overline{s}_{ij}]$ is positioned with respect to the "ring"-like set $\mathcal{R}_{ij} = \{(c_{ij}, s_{ij}) : \underline{R}_{ij}^2 \leq c_{ij}^2 + s_{ij}^2 \leq \overline{R}_{ij}^2\}$ where $\underline{R}_{ij} = \underline{V}_i\underline{V}_j$ and $\overline{R}_{ij} = \overline{V}_i\overline{V}_j$. In our experiments, we observe that $\underline{c}_{ij} > 0$, which we assume hereafter. We should note that this is not a restrictive assumption, similar valid inequalities described below can be generated even if this assumption does not hold.

Let us focus on the case with $\underline{c}_{ij} < \underline{R}_{ij}$, which gives rise to four possibilities:

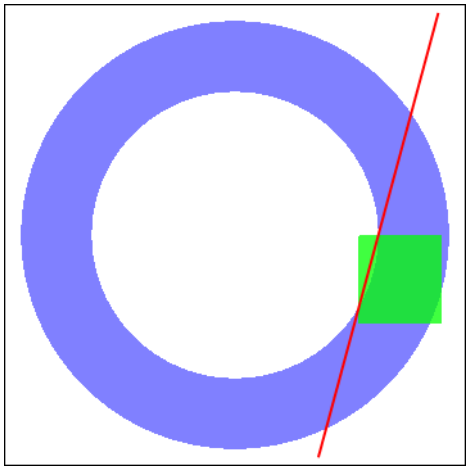
- Case 1: $\|(\underline{c}_{ij}, \underline{s}_{ij})\| < \underline{R}_{ij}$, $\|(\underline{c}_{ij}, \overline{s}_{ij})\| < \underline{R}_{ij}$
- Case 2: $\|(\underline{c}_{ij}, \underline{s}_{ij})\| < \underline{R}_{ij}$, $\|(\underline{c}_{ij}, \overline{s}_{ij})\| \geq \underline{R}_{ij}$
- Case 3: $\|(\underline{c}_{ij}, \underline{s}_{ij})\| \geq \underline{R}_{ij}$, $\|(\underline{c}_{ij}, \overline{s}_{ij})\| < \underline{R}_{ij}$
- Case 4: $\|(\underline{c}_{ij}, \underline{s}_{ij})\| \geq \underline{R}_{ij}$, $\|(\underline{c}_{ij}, \overline{s}_{ij})\| \geq \underline{R}_{ij}$

Figure 5 shows typical examples for each of four cases. In the rest of this section, we concentrate on how we can obtain valid inequalities for Cases 1, 2, and 3.

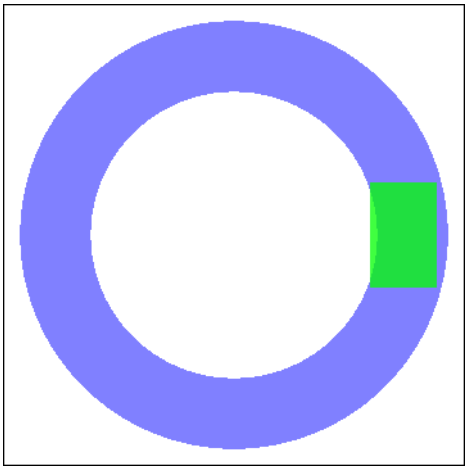


(a) Case 1

(b) Case 2



(c) Case 3



(d) Case 4

Figure 5: Positioning of $\mathcal{B}_{ij}$ and $\mathcal{R}_{ij}$. Red line is the cut produced by Algorithm 1, when applicable.

2.6.1 Valid Inequalities

These cuts are designed to cut off the portion of $\mathcal{B}_{ij}$ inside the inner circle for Cases 1, 2, and 3 as depicted in Figure 5. Algorithm 1 gives the exact procedure. Note that the validity of the inequality follows from the fact that the points cut off from the box have norm less than $\underline{R}_{ij}$. Note that for Case 4, the algorithm would produce the inequality $c_{ij} \geq \underline{c}_{ij}$, hence it is omitted.

Algorithm 1 Generation of Valid Inequalities.

for all $(i,j) \in \mathcal{L}$ **do**
  Compute $\underline{c}_{ij}$, $\overline{c}_{ij}$, $\underline{s}_{ij}$ and $\overline{s}_{ij}$ over $\mathcal{S}'$.
  Update $\mathcal{S}' = \mathcal{S}' \cap \mathcal{B}_{ij}$.
  if $0 < \underline{c}_{ij} < \underline{R}_{ij}$ **then**
    if $\|(\underline{c}_{ij}, \underline{s}_{ij})\| < \underline{R}_{ij}$, $\|(\underline{c}_{ij}, \overline{s}_{ij})\| < \underline{R}_{ij}$ **then**
      Set $y_1 = \overline{s}_{ij}$, $y_2 = \underline{s}_{ij}$ and compute
      $x_1 = \sqrt{\underline{R}_{ij}^2 - \overline{s}_{ij}^2}$, $x_2 = \sqrt{\underline{R}_{ij}^2 - \underline{s}_{ij}^2}$
    else if $\|(\underline{c}_{ij}, \underline{s}_{ij})\| < \underline{R}_{ij}$, $\|(\underline{c}_{ij}, \overline{s}_{ij})\| \geq \underline{R}_{ij}$ **then**
      Set $x_1 = \underline{c}_{ij}$, $y_2 = \underline{s}_{ij}$ and compute
      $y_1 = \sqrt{\underline{R}_{ij}^2 - \underline{c}_{ij}^2}$, $x_2 = \sqrt{\underline{R}_{ij}^2 - \underline{s}_{ij}^2}$
    else if $\|(\underline{c}_{ij}, \underline{s}_{ij})\| \geq \underline{R}_{ij}$, $\|(\underline{c}_{ij}, \overline{s}_{ij})\| < \underline{R}_{ij}$ **then**
      Set $y_1 = \overline{s}_{ij}$, $x_2 = \underline{c}_{ij}$ and compute
      $x_1 = \sqrt{\underline{R}_{ij}^2 - \overline{s}_{ij}^2}$, $y_2 = -\sqrt{\underline{R}_{ij}^2 - \underline{c}_{ij}^2}$
    end if
    Add $(y_1 - y_2)c_{ij} - (x_1 - x_2)s_{ij} \geq x_2 y_1 - x_1 y_2$ as a valid inequality and update $\mathcal{S}'$.
  end if
end for

2.6.2 Computational Experiments

The effect of valid inequalities are tested on our library of instances. The results are summarized in Table 5. We should note that MATPOWER is very efficient and accurate for the seven instances it is able to solve as shown in Table 4. Therefore, we mainly focus on the other seven instances where MATPOWER fails to solve. In Table 5, preprocessing refers to computing variable bounds and valid cuts.

For the 9-bus instances (case9, case9Q), BARON's computation time reduces slightly with the addition of cuts. However, the preprocessing time dominates the total computation time, which is larger than the case without cuts.

For the 30-bus IEEE instances, BARON can require hours to terminate. With the addition of variable bounds, total computation time reduces by 89% and 90% for quadratic and linear objectives, respectively. Quite impressively, the inclusion of valid inequalities further reduces the total computation time to only 17 seconds, less than 0.1% of the computation time without variable bounds and cuts.

For 39-bus instances, the addition of variable bounds brings down total computation time by 76% and 95% for quadratic and linear objectives, respectively. In this case, the inclusion of valid inequalities decreases the computational time for linear objective. On the other hand, cuts slightly increases the total computational time in the case of quadratic objective. However, compared to the case without bounds and cuts, BARON still requires less amount of time.

For 57-bus instances, BARON without bounds was not able to certify the global optimal solution within 3 hours time limit. However, the strengthened variable bounds and valid inequalities enable BARON to solve these instances to global optimality within only 46 seconds.

As a final note, we should note that the applicability of the valid inequalities proposed in this section is not limited to the global optimization of radial networks, they can be used in meshed networks as well. Moreover, precisely the same valid inequalities can be used in SOCP relaxation whereas the transformations $c_{ij} = e_i e_j + f_i f_j$ and $s_{ij} = e_i f_j - e_j f_i$ enable us to obtain linear matrix inequalities to be added to SDP relaxation. Although, for our instances, we have not observed any lower bound improvement in SOCP/SDP relaxations by the inclusion of the valid inequalities, we obtain stronger root node relaxations in BARON. Let RG represent the percentage root gap calculated as $100 \times (1 - z_{\mathrm{r}}/z_{\mathrm{g}})$, where z_{r} and z_{g} are respectively the values

Table 5: BARON with bounds and cuts. PT, BT and TT represent times of preprocessing, BARON solution and total computation in seconds. RG represents the percentage root gap.

		BARON		BARON with bounds				BARON with bounds and cuts				
based on	type	BT (s)	RG (%)	PT (s)	BT (s)	TT (s)	RG (%)	PT (s)	BT (s)	TT (s)	#cuts	RG (%)
case9	quadratic	1.17	11.72	4.34	1.08	5.42	9.72	4.41	1.01	5.42	6	9.71
case9	linear	1.11	16.13	4.12	0.86	4.98	16.02	4.42	1.00	5.42	6	16.05
case9Q	quadratic	1.36	16.91	4.36	1.22	5.58	8.32	4.34	1.11	5.46	6	7.50
case14	quadratic	35.32	9.98	7.11	30.46	37.56	1.98	6.89	41.99	48.88	7	1.98
case14	linear	0.79	0.21	6.95	0.83	7.79	0.15	6.89	0.91	7.80	7	0.39
case_ieee30	quadratic	8347.79	46.93	16.91	900.50	917.41	29.88	17.28	0.36	17.63	14	0.00
case_ieee30	linear	2494.31	46.67	16.89	249.48	266.37	33.69	16.96	0.34	17.30	14	0.00
case30	quadratic	2.52	9.13	17.21	1.91	19.12	7.74	16.94	4.42	21.35	13	7.39
case30	linear	8.50	5.79	17.60	2.40	19.99	5.15	16.23	1.91	18.14	13	4.41
case30Q	quadratic	5.16	13.25	16.53	2.39	18.93	12.11	16.80	1.83	18.64	13	4.27
case39	quadratic	110.59	8.89	28.07	26.03	54.10	0.48	27.72	33.25	60.98	12	1.12
case39	linear	1566.88	2.56	26.94	72.80	99.74	0.51	28.17	42.62	70.79	12	0.52
case57	quadratic	> 10800	46.69	41.57	0.66	42.23	0.00	40.17	0.80	40.97	14	0.00
case57	linear	> 10800	45.17	42.19	0.67	42.87	0.00	45.23	0.67	45.91	14	0.00

of root node relaxation for BARON and global optimal solution. As we can see from Table 5 that addition of bounds and valid inequalities strengthen the root node relaxation of BARON in general. In fact, case_ieee30 and case57 instances are already solved at the root node. We should note that occasionally RG of BARON with bounds and cuts is slightly worse than BARON with bounds. However, this is due to the fact that valid inequalities change the problem structure and may lead to different preprocessing procedures carried out by the solver at the root node.

2.7 Conclusions

In this chapter, we study the impact of generation lower bounds on the performance of convex relaxations of AC OPF problems. For the fundamental two-bus one-generator model, we provide a complete characterization of all possible outcomes of the SOCP relaxation together with a detailed study of the projected feasible regions of the OPF and SOCP relaxation. We provide a library of radial network instances that demonstrate large optimality gaps for SDP and SOCP relaxations. We also propose valid inequalities for the SOCP relaxation, which prove to be useful in reducing the computation time of global solver BARON. We remind the reader here that SDP relaxations are very powerful and their importance is definite. Our work only serves to demonstrate the limitations of SDP relaxations and emphasizes the importance and the need to develop efficient global methods in solving OPF problems.

CHAPTER III

STRONG SOCP RELAXATIONS FOR THE OPTIMAL POWER FLOW PROBLEM OVER MESHED NETWORKS

3.1 Introduction

This chapter proposes new strong SOCP relaxations of the OPF problem and demonstrates their computational advantages over the SDP relaxation and previously described convex quadratic relaxations for the purpose of practically solving large-scale OPF problems. Our starting point is the alternative formulation for the OPF problem introduced in Section 2.2 and proposed before in [30] and [52]. In this formulation, the nonconvexities are present in two types of constraints: one type is the surface of a rotated second-order cone, and the other type involves arctangent functions on voltage angles. The SOCP relaxation in [52] is obtained by convexifying the first type of constraints to obtain SOCP constraints and completely ignoring the second type constraints. We refer to this relaxation as the *classic SOCP relaxation* of the OPF problem. We prove that the standard SOCP relaxation of the rectangular formulation of OPF provides the same bounds as the classic SOCP relaxation. Therefore, if we are able to add convex constraints that are implied by the original constraints involving the arctangent function to the classic SOCP relaxation, then this could yield *stronger relaxation than the classic SOCP relaxation that may also potentially be incomparable to (i.e. not dominated by nor dominates) the standard SDP relaxation.* In this chapter, we propose three efficient ways to achieve this goal.

In the following, we summarize the key contributions of the chapter.

(i) We theoretically analyze the relative strength of the McCormick (linear programming), SOCP, and SDP relaxations of the rectangular and alternative formulations of the OPF problem. As discussed above, this analysis leads us to consider strengthening the classic SOCP relaxation as a way forward to obtaining strong and tractable convex relaxations.

(ii) We propose three efficient methods to strengthen the classic SOCP relaxation.

   (a) In the first approach, we begin by reformulating the arctangent constraints as polynomial constraints whose degrees are proportional to the length of the cycles. This yields a bilinear relaxation of the OPF problem in extended space (that is by addition of artificial variables), where the new variables correspond to edges obtained by triangulating cycles. With this reformulation, we use the McCormick relaxation of the proposed bilinear constraints to strengthen the classic SOCP relaxation. The resulting SOCP relaxation is shown to be incomparable to the SDP relaxation.

   (b) In the second approach, we construct a polyhedral envelope for the arctangent functions in 3-dimension, which are then incorporated into the classic SOCP relaxation. This SOCP relaxation is also shown to be incomparable to the standard SDP relaxation.

   (c) In the third approach, we strengthen the classic SOCP relaxation by dynamically generating valid linear inequalities that separate the SOCP solution from the SDP cone constraints over cycles. We observe that running such a separation oracle a few iterations already produces SOCP relaxation solutions very close to the quality of the full SDP relaxation.

(iii) We conduct extensive computational tests on the proposed SOCP relaxations and compare them with results for an existing SDP relaxation [66] and quadratic relaxations [28, 27]. The computational results can be summarized as follows.

(a) Lower bounds: The lower bounds obtained by the third proposed SOCP relaxation for all MATPOWER test cases from 6-bus to 3375-bus are on average within 99.96% of the lower bounds of the SDP relaxation. The other two proposed relaxation are also on average within 99.7% of the SDP relaxation.

(b) Computation time: Overall, the proposed SOCP relaxations can be solved orders of magnitude faster than the SDP relaxation. The computational advantage is even more evident when a feasible solution of the OPF problem is needed. As an example, consider the largest test instance of the IEEE 3375-bus system. Our proposed SOCP relaxation together with IPOPT provides a solution for this instance and also certifies that this solution is within 0.13% of global optimality, all computed in 157.20 seconds on a modest personal computer.

(c) Comparison with other convex quadratic relaxation: The proposed SOCP relaxations consistently outperform the existing quadratic relaxation in [28] and [27] on the test instances of typical, congested, and small angle difference conditions.

(d) Non-dominance with standard SDP relaxation: The computation also shows that the proposed SOCP relaxations are neither dominated by nor dominate the standard SDP relaxation.

(e) Robustness: The proposed SOCP relaxations perform consistently well on IEEE test cases with randomly perturbed load profiles.

This chapter is organized as follows. Section 3.2 compares six different convex relaxations for the OPF problem based on the rectangular and alternative formulations introduced in Section 2.2. Section 3.3 proposes three ways to strengthen the classic SOCP relaxation. Section 3.4 presents extensive computational experiments.

We make concluding remarks in Section 3.5.

3.2 Comparison of Convex Relaxations

In this section, we first present six different convex relaxations of the OPF problem. In particular, we consider the McCormick, SOCP, and SDP relaxations of both the rectangular formulation (7) and the alternative formulation (10). Then, we analyze their relative strength by comparing their feasible regions. This comparison is an important motivator for the approach we take in the rest of the chapter to generate strong SOCP relaxations. We discuss this in Section 3.2.3.

3.2.1 Standard Convex Relaxations

3.2.1.1 McCormick Relaxation of Rectangular Formulation ($\mathcal{R}_M$).

As shown in [77], the convex hull of the set $\{(x, y, w) : w = xy,\ (x, y) \in [\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}]\}$ is given by

$$\left\{(x, y, w) : \max\{\underline{y}x + \underline{x}y - \underline{x}\underline{y}, \overline{y}x + \overline{x}y - \overline{x}\overline{y}\} \leq w \leq \min\{\underline{y}x + \overline{x}y - \overline{x}\underline{y}, \overline{y}x + \underline{x}y - \underline{x}\overline{y}\}\right\},$$

which we denote as $M(w = xy)$. We use this result to construct McCormick envelopes for the quadratic terms in the rectangular formulation (7). In particular, let us first define the following new variables for each edge $(i, j) \in \mathcal{L}$: $E_{ij} = e_i e_j$, $F_{ij} = f_i f_j$, $H_{ij} = e_i f_j$, and for each bus $i \in \mathcal{B}$: $E_{ii} = e_i^2$, $F_{ii} = f_i^2$. Consider the following set of constraints:

$$p_i^g - p_i^d = G_{ii}(E_{ii} + F_{ii}) + \sum_{j \in \delta(i)} [G_{ij}(E_{ij} + F_{ij}) - B_{ij}(H_{ij} - H_{ji})] \qquad i \in \mathcal{B} \quad (25a)$$

$$q_i^g - q_i^d = -B_{ii}(E_{ii} + F_{ii}) + \sum_{j \in \delta(i)} [-B_{ij}(E_{ij} + F_{ij}) - G_{ij}(H_{ij} - H_{ji})] \quad i \in \mathcal{B} \quad (25b)$$

$$\underline{V}_i^2 \leq E_{ii} + F_{ii} \leq \overline{V}_i^2 \qquad i \in \mathcal{B} \quad (25c)$$

$$-\overline{V}_i \leq e_i, f_i \leq \overline{V}_i \qquad i \in \mathcal{B} \quad (25d)$$

$$M(E_{ij} = e_i e_j), M(F_{ij} = f_i f_j), M(H_{ij} = e_i f_j) \qquad (i,j) \in \mathcal{L} \quad (25e)$$

$$M(E_{ii} = e_i^2), M(F_{ii} = f_i^2),\ E_{ii}, F_{ii} \geq 0 \qquad i \in \mathcal{B}, \quad (25f)$$

where the McCormick envelopes in (25e)-(25f) are constructed using the bounds given in (25d). Using these McCormick envelopes, we obtain a convex relaxation of the rectangular formulation (7) with the feasible region denoted as

$$\mathcal{R}_M = \{(p, q, e, f, E, F, H) : (25), (1h) - (1i)\}. \quad (26)$$

Note that this feasible region is a polyhedron. If the objective function $C_i(p_i^g)$ is linear, then we have a linear programming relaxation of the OPF problem.

3.2.1.2 McCormick Relaxation of Alternative Formulation ($\mathcal{A}_M$).

Using the similar technique on the alternative formulation (10), we define new variables $C_{ij} = c_{ij}^2$, $S_{ij} = s_{ij}^2$, $D_{ij} = c_{ii}c_{jj}$ for each edge $(i,j) \in \mathcal{L}$, and consider the following set of constraints:

$$C_{ij} + S_{ij} = D_{ij} \qquad (i,j) \in \mathcal{L} \quad (27a)$$

$$-\overline{V}_i\overline{V}_j \leq c_{ij}, s_{ij} \leq \overline{V}_i\overline{V}_j \qquad (i,j) \in \mathcal{L} \quad (27b)$$

$$M(C_{ij} = c_{ij}^2), M(S_{ij} = s_{ij}^2), M(D_{ij} = c_{ii}c_{jj}), C_{ij}, S_{ij} \geq 0 \qquad (i,j) \in \mathcal{L}, \quad (27c)$$

where the McCormick envelops in (27c) are constructed using the bounds given in (27b) and (10d). Denote the feasible region of the corresponding convex relaxation

as

$$\mathcal{A}_M = \{(p, q, c, s, C, S, D) : (27), (10b) - (10e), (1h) - (1i)\}. \tag{28}$$

Again, $\mathcal{A}_M$ is a polyhedron.

3.2.1.3 SDP Relaxations of Rectangular Formulation ($\mathcal{R}_{SDP}, \mathcal{R}^c_{SDP}, \mathcal{R}^r_{SDP}$).

To apply SDP relaxation to the rectangular formulation (7), 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:

$$p^g_i - p^d_i = G_{ii}X_{ii} + \sum_{j\in\delta(i)} [G_{ij}\Re(X_{ij}) + B_{ij}\Im(X_{ij})] \qquad i \in \mathcal{B} \tag{29a}$$

$$q^g_i - q^d_i = -B_{ii}X_{ii} + \sum_{j\in\delta(i)} [-B_{ij}\Re(X_{ij}) + G_{ij}\Im(X_{ij})] \qquad i \in \mathcal{B} \tag{29b}$$

$$\underline{V}_i^2 \le X_{ii} \le \overline{V}_i^2 \qquad i \in \mathcal{B} \tag{29c}$$

$$X \text{ is hermitian} \tag{29d}$$

$$X \succeq 0 \tag{29e}$$

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 $i \in \mathcal{B}$. If $X = VV^*$, then $\text{rank}(\Re(X))$ and $\text{rank}(\Im(X))$ are both equal to 2, and (29a)-(29c) exactly recovers (7b)-(7c), (1f). By ignoring the rank constraints, we come to the standard SDP relaxation of the rectangular formulation (7), whose feasible region is defined as

$$\mathcal{R}^c_{SDP} = \{(p, q, W) : (29), (1h) - (1i)\}. \tag{30}$$

This SDP relaxation is in the complex domain. There is also an SDP relaxation in the real domain by defining $W \in \mathbb{R}^{2|C|\times 2|C|}$. The associated constraints are

$$p_i^g - p_i^d = G_{ii}(W_{ii} + W_{i'i'}) + \sum_{j\in\delta(i)} [G_{ij}(W_{ij} + W_{i'j'}) - B_{ij}(W_{ij'} - W_{ji'})] \qquad i \in \mathcal{B} \tag{31a}$$

$$q_i^g - q_i^d = -B_{ii}(W_{ii} + W_{i'i'}) + \sum_{j\in\delta(i)} [-B_{ij}(W_{ij} + W_{i'j'}) - G_{ij}(W_{ij'} - W_{ji'})] \quad i \in \mathcal{B} \tag{31b}$$

$$\underline{V}_i^2 \leq W_{ii} + W_{i'i'} \leq \overline{V}_i^2 \qquad i \in \mathcal{B} \tag{31c}$$

$$W \succeq 0, \tag{31d}$$

where $i' = i + |\mathcal{B}|$. If $W = [e; f][e^T, f^T]$, i.e. rank(W)=1, then (31a)-(31c) exactly recovers (7b)-(7c), (1f). We denote the feasible region of this SDP relaxation in the real domain as

$$R_{SDP}^r = \{(p, q, W) : (31a) - (31d), (1h) - (1i)\}. \tag{32}$$

The SDP relaxation in the real domain is first proposed in [6, 5], and [67]. The SDP relaxation in the complex domain is formulated in [18] and [109] and is widely used in the literature now for its notational simplicity. These two SDP relaxations produce the same bound, since the solution of one can be used to derive a solution to the other with the same objective function value. See e.g., Section 3.3 in [97] for a formal proof. Henceforth, we refer to the standard SDP relaxation as $\mathcal{R}_{SDP}$ that is $\mathcal{R}_{SDP} := \mathcal{R}_{SDP}^c = \mathcal{R}_{SDP}^r$, where the second equality (with some abuse of notation) is meant to imply that the two relaxations have the same projection in the space of the p, q variables.

3.2.1.4 SOCP Relaxation of Rectangular Formulation ($\mathcal{R}_{SOCP}$).

We can further apply SOCP relaxation to the SDP constraint (29e) by only imposing the following constraints on all the 2×2 submatrices of X for each line $(i, j) \in \mathcal{L}$,

$$\begin{bmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{bmatrix} \succeq 0 \quad (i,j) \in \mathcal{L}. \tag{33}$$

This is a relaxation of (29e), because (29e) requires all principal submatrices of X be SDP (see e.g., [47]). Note that (33) has a 2×2 hermitian matrix, i.e., $X_{ij} = X_{ji}^*$. Using the Sylvester criterion, (33) is equivalent to $X_{ij}X_{ji} \leq X_{ii}X_{jj}$ and $X_{ii}, X_{jj} \geq 0$. The first inequality can be written as $\Re(X_{ij})^2 + \Im(X_{ij})^2 + \left(\frac{X_{ii}-X_{jj}}{2}\right)^2 \leq \left(\frac{X_{ii}+X_{jj}}{2}\right)^2$, which is an SOCP constraint in the real domain. Thus, we have an SOCP relaxation of the rectangular formulation with the feasible region defined as

$$\mathcal{R}_{SOCP} = \{(p, q, X) : (29a) - (29c), (33), (1\text{h}) - (1\text{i})\}. \tag{34}$$

This SOCP relaxation is also proposed in the literature, see e.g., [76]. In [71], this relaxation is proven to be equivalent to the SOCP relaxation of DistFlow model proposed in [8, 9].

3.2.1.5 SOCP Relaxation of Alternative Formulation ($\mathcal{A}^*_{SOCP}$).

This is the classic SOCP relaxation introduced in Section 2.2.3 and first proposed in [52]. We denote the feasible region of this SOCP relaxation as

$$\mathcal{A}^*_{SOCP} = \{(p, q, c, s) : (10\text{b}) - (10\text{e}), (1\text{h}) - (1\text{i}), (13)\}. \tag{35}$$

3.2.1.6 SDP Relaxation of Alternative Formulation ($\mathcal{A}_{SDP}$).

We can also apply SDP relaxation to the nonconvex quadratic constraints (10f) in the alternative formulation (10). In particular, define a vector $z \in \mathbb{R}^{2|\mathcal{L}|+|\mathcal{B}|}$, of which the first $2|\mathcal{L}|$ components are indexed by the set of branches $(i, j) \in \mathcal{L}$, and the last $|\mathcal{B}|$ components are indexed by the set of buses $i \in \mathcal{B}$. Essentially, z represents

$((c_{ij})_{(i,j)\in\mathcal{L}}, (s_{ij})_{(i,j)\in\mathcal{L}}, (c_{ii})_{i\in\mathcal{B}})$. Then define a real matrix variable Z to approximate zz^T and consider the following set of constraints:

$$Z_{(ij),(ij)} + Z_{(i'j'),(i'j')} = Z_{(ii),(jj)} \qquad (i,j) \in \mathcal{L} \tag{36a}$$

$$Z \succeq zz^T \tag{36b}$$

$$Z_{(ij),(ij)} \leq (\overline{V}_i \overline{V}_j)^2,\ Z_{(i'j'),(i'j')} \leq (\overline{V}_i \overline{V}_j)^2 \qquad (i,j) \in \mathcal{L} \tag{36c}$$

$$Z_{(ii),(ii)} \leq (\underline{V}_i^2 + \overline{V}_i^2) c_{ii} - (\underline{V}_i \overline{V}_i)^2 \qquad i \in \mathcal{B}, \tag{36d}$$

where $i' = i + |\mathcal{L}|$. Constraints (36a) and (36b) are the usual SDP relaxation of (10f), and constraints (36c) and (36d) are used to properly upper bound the diagonal elements of Z, where constraint (36d) follows from applying McCormick envelopes on the squared terms c_{ii}^2. In particular, if $z_{(ij)}$ is restricted to be between $[\underline{z}_{(ij)}, \overline{z}_{(ij)}]$, then the McCormick upper envelope for the diagonal element $Z_{(ij),(ij)}$ is given as $Z_{(ij),(ij)} \leq (\underline{z}_{(ij)} + \overline{z}_{(ij)}) z_{(ij)} - \underline{z}_{(ij)} \overline{z}_{(ij)}$. Denote the feasible region of this SDP relaxation of the alternative formulation (10) as

$$\mathcal{A}_{SDP} = \{(p, q, c, s, Z) : (36), (10b) - (10e), (1h) - (1i)\}. \tag{37}$$

Note that this SDP relaxation of the alternative formulation is derived using standard techniques, but to the best of our knowledge, it has not been discussed in the literature.

3.2.2 Comparison of Relaxations

The main result of this section is Theorem 2, which presents the relative strength of the various convex relaxations introduced above. In order to compare relaxations, they must be in the same variable space. Also the objective function depends only on the value of the real powers. Therefore, we study the feasible regions of the above convex relaxations projected to the space of real and reactive powers, i.e. the (p, q) space. We use '$\hat{}$' to denote this projection. For example, $\hat{\mathcal{R}}_M$ is the projection of $\mathcal{R}_M$ to the (p, q) space.

Theorem 2. *Let $\mathcal{R}_M$, $\mathcal{A}_M$, $\mathcal{R}_{SDP}$, $\mathcal{R}_{SOCP}$, $\mathcal{A}^*_{SOCP}$, and $\mathcal{A}_{SDP}$ be the McCormick relaxation of the rectangular formulation* (7)*, the McCormick relaxation of the alternate formulation* (10)*, the SDP relaxation of the rectangular formulation* (30)*, the SOCP relaxation of the rectangular formulation* (34)*, the SOCP relaxation of the alternative formulation* (35)*, and the SDP relaxation of the alternative formulation* (37)*, respectively. Then:*

(i) The following relationships hold between the feasible regions of the convex relaxations when projected onto the (p, q) space:

$$\begin{array}{ccc} \hat{\mathcal{R}}_{SDP} \subseteq \hat{\mathcal{R}}_{SOCP} = \hat{\mathcal{A}}^*_{SOCP} & \subseteq & \hat{\mathcal{A}}_{SDP} \\ \cap| & & \\ \hat{\mathcal{R}}_M \quad \supseteq & & \hat{\mathcal{A}}_M \end{array} \tag{38}$$

(ii) All the inclusions in (38) *can be strict.*

In Section 3.2.2.1 we present the proof of part (i) of Theorem 2, and in Section 3.2.2.2 we present examples to verify part (ii) of the theorem.

3.2.2.1 Pairwise comparison of relaxations

The proof of part (i) of Theorem 2 is divided into Propositions 1-4.

Proposition 1. $\hat{\mathcal{R}}_M \supseteq \hat{\mathcal{A}}_M$.

Proof. We want to show that for any given $(p, q, c, s, C, S, D) \in \mathcal{A}_M$, we can find (e, f, E, F, H) such that (p, q, e, f, E, F, H) is in $\mathcal{R}_M$. For this purpose, set $e_i = f_i = 0$ for $i \in \mathcal{B}$, $E_{ij} = c_{ij}$, F_{ij}=0, $H_{ij} = s_{ij}$, $H_{ji} = 0$ for each $(i, j) \in \mathcal{L}$, and $E_{ii} = c_{ii}$ and $F_{ii} = 0$ for each $i \in \mathcal{B}$. By this construction, we have $E_{ii}+F_{ii} = c_{ii}$, $E_{ij}+F_{ij} = c_{ij}$, and $H_{ij} - H_{ji} = s_{ij}$. Therefore, (10b)-(10c) implies that the constructed E, F, H satisfy (25a)-(25b); (10d) implies (25c); (25d) is trivially satisfied since $e_i = f_i = 0$. Now to

see the McCormick envelopes (25e)-(25f) are satisfied, consider $M(E_{ij} = e_i e_j)$. Using the bounds (25d) on e_i, f_i, $M(E_{ij} = e_i e_j)$ can be written as

$$\max\{-\overline{V}_i e_j - \overline{V}_j e_i - \overline{V}_i \overline{V}_j,\ \overline{V}_i e_j + \overline{V}_j e_i - \overline{V}_i \overline{V}_j\} \leq E_{ij}$$
$$E_{ij} \leq \min\{-\overline{V}_i e_j + \overline{V}_j e_i + \overline{V}_i \overline{V}_j,\ \overline{V}_i e_j - \overline{V}_j e_i + \overline{V}_i \overline{V}_j\}.$$

Since $e_i = 0$ for all $i \in \mathcal{B}$, the above inequalities reduce to $-\overline{V}_i \overline{V}_j \leq E_{ij} \leq \overline{V}_i \overline{V}_j$, which is implied by $E_{ij} = c_{ij}$ and the bounds (27b). Similar reasoning can be applied to verify that the other McCormick envelopes in (25e)-(25f) are satisfied. Finally, it is straightforward to see that $E_{ii} = c_{ii} \geq 0, F_{ii} = 0$. Therefore, the constructed (p, q, e, f, E, F, H) is in $\mathcal{R}_M$. □

In fact, the above argument only relies on constraints (10b)-(10d) and (27b) in $\mathcal{A}_M$. This suggests that $\hat{\mathcal{A}}_M$ may be strictly contained in $\hat{\mathcal{R}}_M$, which is indeed the case shown in Section 3.2.2.2.

Proposition 2. $\hat{\mathcal{A}}_M \supseteq \hat{\mathcal{A}}^*_{SOCP}$.

Proof. It suffices to prove that $\text{proj}_{p,q,c,s}\mathcal{A}_M \supseteq \mathcal{A}^*_{SOCP}$. For this purpose, we want to show that for each $(p, q, c, s) \in \mathcal{A}^*_{SOCP}$, there exists (C, S, D) such that $(p, q, c, s, C, S, D) \in \mathcal{A}_M$. In particular, set $C_{ij} = c^2_{ij}$, $S_{ij} = c_{ii}c_{jj} - c^2_{ij} \geq s^2_{ij}$, and $D_{ij} = c_{ii}c_{jj}$ for each $(i, j) \in \mathcal{L}$. Note that C_{ij} and D_{ij} satisfy constraints (27c) by the definition of McCormick envelopes. So, it remains to verify if S_{ij} satisfies $M(S_{ij} = s^2_{ij})$. This involves verifying:

$$2(\overline{V}_j \overline{V}_i)|s_{ij}| - (\overline{V}_j \overline{V}_i)^2 \leq S_{ij} \leq (\overline{V}_j \overline{V}_i)^2. \tag{39}$$

The first inequality holds because $S_{ij} \geq s^2_{ij}$ and $s^2_{ij} - 2(\overline{V}_j \overline{V}_i)|s_{ij}| + (\overline{V}_j \overline{V}_i)^2 = (|s_{ij}| - \overline{V}_j \overline{V}_i)^2 \geq 0$. The second inequality holds since $S_{ij} = c_{ii}c_{jj} - c^2_{ij} \leq c_{ii}c_{jj} \leq (\overline{V}_i \overline{V}_j)^2$. Thus, the result follows. □

The next result is straightforward, however we present it for completion.

Proposition 3. $\hat{\mathcal{A}}^*_{SOCP} = \hat{\mathcal{R}}_{SOCP}$.

Proof. Note that $X_{ii} = c_{ii}$, $X_{jj} = c_{jj}$ and $X_{ij} = c_{ij} + \mathrm{i}s_{ij}$ forms a bijection between the (c, s) variables in $\mathcal{A}^*_{SOCP}$ and the X variable in $\mathcal{R}_{SOCP}$. Given this bijection, (7b)-(7c), (1f) are equivalent to (29a)-(29c) and (10e) is equivalent to $X_{ij} = X^*_{ji}$. As mentioned in Section 3.2.1.4, the hermitian condition in (33) is equivalent to $|X_{ij}|^2 \leq X_{ii}X_{jj}$ and $X_{ii}, X_{jj} \geq 0$, which is exactly (12). □

Proposition 4. $\hat{\mathcal{A}}^*_{SOCP} \subseteq \hat{\mathcal{A}}_{SDP}$.

Proof. It suffices to show that $\mathcal{A}^*_{SOCP} \subseteq \mathrm{proj}_{p,q,c,s}\mathcal{A}_{SDP}$. For this, we can show that given any $(p, q, c, s) \in \mathcal{A}^*_{SOCP}$, there exists a Z such that $(p, q, c, s, Z) \in \mathcal{A}_{SDP}$. In particular, we construct $Z = zz^T + Z'$, where $z = ((c_{ij})_{(i,j)\in\mathcal{L}}, (s_{ij})_{(i,j)\in\mathcal{L}}, (c_{ii})_{i\in\mathcal{B}})$, and Z' is a diagonal matrix with the first $|\mathcal{L}|$ entries on the diagonal equal to $c_{ii}c_{jj} - (c^2_{ij} + s^2_{ij})$ for each $(i, j) \in \mathcal{L}$ and all other entries equal to zero. By this construction, (36a) is satisfied. Also $Z \succeq zz^T$, i.e. (36b) is satisfied. Note that the first $|\mathcal{L}|$ entries on the diagonal of Z is equal to $c_{ii}c_{jj} - s^2_{ij}$. Therefore, the bounds in (36c) are equivalent to $c_{ii}c_{jj} - s^2_{ij} \leq \overline{V}^2_i\overline{V}^2_j$, $s^2_{ij} \leq \overline{V}^2_i\overline{V}^2_j$, where the first one follows from (10d) and the second one follows from (10d) and (12). Finally, constraint (36d) is the McCormick envelope of c^2_{ii} with the same bounds in (10d), therefore it is also satisfied by the solution of $\mathcal{A}^*_{SOCP}$. □

3.2.2.2 Strictness of Inclusions.

Now, we prove by examples that all the inclusions in Theorem 2 can be strict.

(i) $\hat{\mathcal{A}}^*_{SOCP} \subset \hat{\mathcal{A}}_M \subset \hat{\mathcal{R}}_M$: Consider the 9-bus radial network with quadratic objective function from Chapter 2. We compare the strength of these three types of relaxations in Table 6, which shows that the inclusions can be strict.

Table 6: Percentage optimality gap for $\mathcal{R}_M$, $\mathcal{A}_M$, and $\mathcal{A}^*_{SOCP}$ with respect to global optimality found by BARON.

case	objective	$\mathcal{R}_M$	$\mathcal{A}_M$	$\mathcal{A}^*_{SOCP}$
9-bus	quadratic	89.46	69.13	52.69

Here we use BARON [95] to obtain the global optimal solutions of the OPF problem.

(ii) $\hat{\mathcal{A}}^*_{SOCP} \subset \hat{\mathcal{A}}_{SDP}$: Consider a 2-bus system. Since there is only one transmission line, $\mathcal{A}^*_{SOCP}$ has only one SOCP constraint

$$c_{12}^2 + s_{12}^2 \leq c_{11}c_{22}, \tag{40}$$

while the SDP relaxation $\mathcal{A}_{SDP}$ has

$$Z \succeq zz^T \iff \begin{bmatrix} 1 & c_{12} & s_{12} & c_{11} & c_{22} \\ c_{12} & Z_{11} & Z_{12} & Z_{13} & Z_{14} \\ s_{12} & Z_{21} & Z_{22} & Z_{23} & Z_{24} \\ c_{11} & Z_{31} & Z_{32} & Z_{33} & Z_{34} \\ c_{22} & Z_{41} & Z_{42} & Z_{43} & Z_{44} \end{bmatrix} \succeq 0 \tag{41}$$

with the additional constraints $Z_{11} + Z_{22} = Z_{43}$, $Z_{11} \leq 1.4641$, $Z_{22} \leq 1.4641$, $Z_{33} \leq 2.02c_{12} - 0.9801$ and $Z_{44} \leq 2.02s_{12} - 0.9801$, assuming that $\underline{V}_1 = \underline{V}_2 = 0.9$ and $\overline{V}_1 = \overline{V}_2 = 1.1$.

Now, consider a point $(c_{12}, s_{12}, c_{11}, c_{22}) = (1.000, 0.100, 1.000, 1.000)$, which clearly violates constraint (40). However, one can extend this point in SDP

relaxation as follows:

$$\begin{bmatrix} 1 & z^T \\ z & Z \end{bmatrix} = \begin{bmatrix} 1.000 & 1.000 & 0.100 & 1.000 & 1.000 \\ 1.000 & 1.006 & 0.100 & 0.997 & 0.997 \\ 0.100 & 0.100 & 0.017 & 0.100 & 0.100 \\ 1.000 & 0.997 & 0.100 & 1.029 & 1.023 \\ 1.000 & 0.997 & 0.100 & 1.023 & 1.029 \end{bmatrix} \succeq 0 \tag{42}$$

This proves our claim that the SDP relaxation $\hat{\mathcal{A}}_{SDP}$ can be weaker than the SOCP relaxation $\hat{\mathcal{A}}^*_{SOCP}$.

(iii) $\mathcal{R}_{SDP} \subset \mathcal{R}_{SOCP}$: Although this relation holds as equality for radial networks [92], the inclusion can be strict for meshed networks. For example, Table 7 demonstrates this fact, e.g., the instance case6ww from the MATPOWER library [111].

3.2.3 Our Choice of Convex Relaxation

We discuss some consequences of Theorem 2 here.

Among LPs, SOCPs, and SDPs, the most tractable relaxation are LP relaxations. In Chapter 2, we showed how $\mathcal{A}_M$ may be used (together with specialized cutting planes) to solve tree instances of OPF globally. Theorem 2 provides a theoretical basis for selection of $\mathcal{A}_M$ over $\mathcal{R}_M$ if one wishes to use linear programming relaxations. However, as seen in Theorem 2, both McCormick relaxations are weaker than the SOCP relaxations. In the context of meshed systems, in our preliminary experiments, the difference in quality of bounds produced by the LP and SOCP based relaxations is quite significant.

As stated in Section 3.1, the goal of this chapter is to avoid using SDP relaxations. However, it is interesting to observe the relative strength of different SDP relaxations. On the one hand, $\mathcal{R}_{SDP}$ is the best relaxation among the relaxation considered in Theorem 2. On the other hand, quite remarkably, $\mathcal{A}_{SDP}$ is weaker than $\mathcal{A}^*_{SOCP}$.

Clearly, if one chooses to use SDPs, $\mathcal{A}_{SDP}$ is not a good choice. One may define a different SOCP relaxation applied to the SDP relaxation of the alternative formulation by relaxing constraint (36b) and replacing it with 2×2 principle submatrices. Such a relaxation would yield very poor bounds and thus undesirable.

We note that [28] and [27] show that a relaxation with same bound as $\mathcal{R}_{SOCP}$ is quite strong. The equality $\mathcal{R}_{SOCP} = \mathcal{A}^*_{SOCP}$ is a straightforward observation. However, between these two relaxations, working with the classic SOCP relaxation $\mathcal{A}^*_{SOCP}$ provides a very natural way to strengthen these SOCP relaxations. In particular, $\mathcal{A}^*_{SOCP}$ was obtained by first dropping the nonconvex arctangent constraint (11). If one is able to incorporate LP/SOCPs based convex outer approximations of these constraints, then $\mathcal{A}^*_{SOCP}$ could be significantly strengthened. Indeed one may be even able to produce relaxations that are incomparable to $\mathcal{R}_{SDP}$. We show how to accomplish this in the next section.

3.3 Strong SOCP Relaxation for Meshed Networks

In this section, we propose three methods to strengthen the classic SOCP relaxation $\mathcal{A}^*_{SOCP}$. In Section 3.3.1, we propose a new relaxation of the arctangent constraints (11) as polynomial constraints over cycles in the power network. These polynomial constraints with degrees proportional to the length of the cycles are then transformed to systems of bilinear equations by triangulating the cycles. We apply McCormick relaxation to the bilinear constraints. The resulting convex relaxation is incomparable to the standard SDP relaxation, i.e. the former does not dominate nor is dominated by the latter. In Section 3.3.2, we construct a polyhedral envelope for the arctangent functions in 3-dimension and incorporate it into the classic SOCP relaxation, which again results in a convex relaxation incomparable to the SDP relaxation. In Section 3.3.3, we strengthen the classic SOCP relaxation by dynamically generating valid linear inequalities that separate the SOCP solution from the SDP cones. This approach

takes the advantage of the efficiency of the SOCP relaxation and the accuracy of the SDP relaxation. It very rapidly produces solutions of quality extremely close to the SDP relaxation. In Section 3.3.4, we propose variable bounding techniques that provide tight variable bounds for the first two strengthening approaches.

3.3.1 A New Cycle-based Relaxation of OPF

Our first method is based on the following observation: instead of satisfying the angle condition (11) for each $(i,j) \in \mathcal{L}$, we consider a relaxation that guarantees angle differences sum up to 0 modulo 2π over every cycle C in the power network, i.e.

$$\sum_{(i,j)\in C} \theta_{ij} = 2\pi k, \quad \text{for some } k \in \mathbb{Z}, \tag{43}$$

where θ_{ij} is the angle difference between adjacent buses i and j. Although the number of cycles in a power network may be large, it suffices to enforce (43) only over cycles in a cycle basis. For a formal definition of cycle basis, see Appendix B. Since $\theta = 2\pi k$ for some $k \in \mathbb{Z} \iff \cos\theta = 1$, we can equivalently write (43) as follows:

$$\cos\left(\sum_{(i,j)\in C} \theta_{ij}\right) = 1. \tag{44}$$

We call (44) *cycle constraints*. By expanding the cosine term appropriately, we can express (44) in terms of $\cos(\theta_{ij})'s$ and $\sin(\theta_{ij})$'s. According to the construction of the alternative formulation (10), we have the following relationship between c, s and θ

$$\cos(\theta_{ij}) = \frac{c_{ij}}{\sqrt{c_{ii}c_{jj}}} \quad \text{and} \quad \sin(\theta_{ij}) = -\frac{s_{ij}}{\sqrt{c_{ii}c_{jj}}} \quad \forall (i,j) \in \mathcal{L}. \tag{45}$$

Using (45), the cycle constraint (44) can be reformulated as a degree $|C|$ homogeneous polynomial equality in the c_{ii}, c_{ij}, and s_{ij} variables. Denote it as $p_{|C|}$.

Unfortunately, directly solving the polynomial reformulation can be intractable, especially for large cycles, since $p_{|C|}$ can have up to $2^{|C|-1}+1$ monomials and each monomial is of degree $|C|$. It is well known that any polynomial constraint can be

written as a set of quadratic constraints using additional variables. In our case, there is a natural way to obtain a $O(|C|)$ sized system of bilinear constraints by decomposing large cycles into smaller ones. Before introducing the cycle decomposition method, we present two building blocks in the construction, namely, the cycle constraints over 3- and 4-cycles.

3.3.1.1 3-Cycle.

Let us first analyze the simplest case, namely the cycle constraint over a 3-cycle. Expansion of $\cos(\theta_{12} + \theta_{23} + \theta_{31}) = 1$ gives $p_3 = 0$, where p_3 is the following cubic polynomial:

$$p_3 = c_{12}(c_{23}c_{31} - s_{23}s_{31}) - s_{12}(s_{23}c_{31} + c_{23}s_{31}) - c_{11}c_{22}c_{33}. \tag{46}$$

Note that if the entire power network is a 3-cycle, then adding (46) to the alternative formulation (10) makes it exactly equivalent to the rectangular formulation (7). Now we claim that p_3 can be replaced with two bilinear constraints. To start with, let us define the polynomials

$$p_{ij} = c_{ij}^2 + s_{ij}^2 - c_{ii}c_{jj}, \quad (i,j) \in \{(1,2),(2,3),(3,1)\} \tag{47}$$

and

$$q_3^1 = s_{12}c_{33} + c_{23}s_{31} + s_{23}c_{31} \tag{48a}$$

$$q_3^2 = c_{12}c_{33} - c_{23}c_{31} + s_{23}s_{31}. \tag{48b}$$

Then, we have the following result.

Proposition 5. $\{(c,s) : p_3 = p_{12} = p_{23} = p_{31} = 0\} = \{(c,s) : q_3^1 = q_3^2 = p_{12} = p_{23} = p_{31} = 0\}$.

Proof. We prove the above two sets are equal by showing they contain each other. ($\subseteq$): $p_{12} = p_{23} = p_{31} = 0$ ensures that there exist θ_{12}, θ_{23}, and θ_{31} such that the c, s, θ

variables satisfy (45). Then, using the fact that (46) is derived using (45) and (44), $p_3 = 0$ implies that

$$\theta_{12} = -(\theta_{23} + \theta_{31}) + 2k\pi \quad \forall k \in \mathbb{Z}, \tag{49}$$

which is equivalent to the following two equalities,

$$\sin(\theta_{12}) = \sin(-(\theta_{23} + \theta_{31})) \text{ and } \cos(\theta_{12}) = \cos(-(\theta_{23} + \theta_{31})). \tag{50}$$

Note that the sine constraint in (50) implies that

$$\theta_{12} = -(\theta_{23} + \theta_{31}) + 2k\pi \text{ or } \theta_{12} = (\theta_{23} + \theta_{31}) + (2k+1)\pi \quad \forall k \in \mathbb{Z},$$

while the cosine constraint in (50) implies that

$$\theta_{12} = -(\theta_{23} + \theta_{31}) + 2k\pi \text{ or } \theta_{12} = (\theta_{23} + \theta_{31}) + 2k\pi \quad \forall k \in \mathbb{Z}.$$

Therefore, we need both the sine and cosine constraints in (50) to enforce (49). Then, expanding the sine and cosine constraints in (50) using the sum formulas and replacing the trigonometric terms with their algebraic equivalents in terms of the c and s variables via (45), we obtain $q_3^1 = q_3^2 = 0$.

($\supseteq$): $q_3^1 = q_3^2 = p_{12} = p_{23} = p_{31} = 0$ imply that we have

$$\begin{aligned} p_3 &= c_{12}(c_{23}c_{31} - s_{23}s_{31}) - s_{12}(s_{23}c_{31} + c_{23}s_{31}) - c_{11}c_{22}c_{33} \\ &= c_{12}(c_{12}c_{33}) - s_{12}(-s_{12}c_{33}) - c_{11}c_{22}c_{33} \\ &= (c_{12}c_{12} + s_{12}s_{12})c_{33} - c_{11}c_{22}c_{33} \\ &= 0, \end{aligned}$$

where the second equality follows due to $q_3^1 = 0$ and $q_3^2 = 0$, and the last one follows due to $p_{12} = 0$. This completes the proof. □

Finally, we note that one can obtain two more pairs of equalities constructed in the same fashion as (48) by considering other permutations such as $\theta_{23} = -(\theta_{12}+\theta_{31})+2k\pi$ and $\theta_{31} = -(\theta_{12} + \theta_{23}) + 2k\pi$.

3.3.1.2 4-Cycle.

Now, let us analyze the cycle constraint over a 4-cycle. Expansion of $\cos(\theta_{12} + \theta_{34} + \theta_{23} + \theta_{41}) = 1$ together with (45) gives $p_4 = 0$, where p_4 is the following quartic polynomial,

$$p_4 = (c_{12}c_{34} - s_{12}s_{34})(c_{23}c_{41} - s_{23}s_{41}) - (s_{12}c_{34} + c_{12}s_{34})(s_{23}c_{41} + c_{23}s_{41}) - c_{11}c_{22}c_{33}c_{44}. \tag{51}$$

We again claim that p_4 can be replaced with two bilinear constraints. To start with, define the following polynomials

$$p_{ij} = c_{ij}^2 + s_{ij}^2 - c_{ii}c_{jj}, \quad (i,j) \in \{(1,2),(2,3),(3,4),(4,1)\}$$

and

$$q_4^1 = s_{12}c_{34} + c_{12}s_{34} + s_{23}c_{41} + c_{23}s_{41} \tag{52a}$$

$$q_4^2 = c_{12}c_{34} - s_{12}s_{34} - c_{23}c_{41} + s_{23}s_{41}. \tag{52b}$$

Then, we have the following proposition.

Proposition 6. $\{(c,s) : p_4 = p_{12} = p_{23} = p_{34} = p_{41} = 0\} = \{(c,s) : q_4^1 = q_4^2 = p_{12} = p_{23} = p_{34} = p_{41} = 0\}$.

Proof. The proof is similar to that of Proposition 5.
($\subseteq$): $p_4 = p_{12} = p_{23} = p_{34} = p_{41} = 0$ imply that we can find θ_{12}, θ_{23}, θ_{34} and θ_{41} satisfying

$$\theta_{12} + \theta_{34} = -(\theta_{23} + \theta_{41}) + 2k\pi \quad \forall k \in \mathbb{Z}. \tag{53}$$

Then, by taking sine and cosine of both sides, expanding the right hand side using sine and cosine sum formulas and replacing the trigonometric terms with their algebraic equivalents in terms of c and s variables via (45), we obtain $q_4^1 = q_4^2 = 0$.

($\supseteq$): $q_4^1 = q_4^2 = p_{12} = p_{23} = p_{34} = p_{41} = 0$ imply that we have

$$\begin{aligned} p_4 &= (c_{12}c_{34} - s_{12}s_{34})(c_{23}c_{41} - s_{23}s_{41}) - (s_{12}c_{34} + c_{12}s_{34})(s_{23}c_{41} + c_{23}s_{41}) - c_{11}c_{22}c_{33}c_{44} \\ &= (c_{23}c_{41} - s_{23}s_{41})^2 + (s_{23}c_{41} + c_{23}s_{41})^2 - c_{11}c_{22}c_{33}c_{44} \\ &= (c_{23}^2 + s_{23}^2)(c_{41}^2 + s_{41}^2) - c_{11}c_{22}c_{33}c_{44} \\ &= 0, \end{aligned}$$

where the second equality follows due to $q_4^1 = 0$ and $q_4^2 = 0$, and the last one follows due to $p_{23} = 0$ and $p_{41} = 0$. This completes the proof. □

3.3.1.3 Larger Cycles.

Now we introduce the cycle decomposition procedure so that the cycle constraint (43) over any cycle C can be reformulated as a system of bilinear equalities, where the number of bilinear equations is $O(|C|)$. Let C be a cycle with buses numbered from 1 to n.

3-decomposition of a cycle C. Suppose $|C| = n \geq 4$. We can decompose C into 3-cycles by creating artificial edges $(1, i)$ for $i = 3, \ldots, n-1$. Now, we apply the exact reformulation from Section 3.3.1.1 for each of these cycles. The polynomial cycle constraint p_n is replaced by the following set of bilinear equalities:

$$\tilde{s}_{1,i}c_{i+1,i+1} + s_{i,i+1}\tilde{c}_{1,i+1} - \tilde{s}_{1,i+1}c_{i,i+1} = 0 \qquad i = 2, 3, \ldots, n-1 \tag{54a}$$

$$\tilde{c}_{1,i}c_{i+1,i+1} - c_{i,i+1}\tilde{c}_{1,i+1} - \tilde{s}_{1,i+1}s_{i,i+1} = 0 \qquad i = 2, 3, \ldots, n-1 \tag{54b}$$

$$\tilde{c}_{1,i}^2 + \tilde{s}_{1,i}^2 = c_{11}c_{ii} \qquad i = 2, 3, \ldots, n-1 \tag{54c}$$

$$c_{ij}^2 + s_{ij}^2 = c_{ii}c_{jj} \qquad (i, j) \in C. \tag{54d}$$

Here, $\tilde{c}_{1i}$ and $\tilde{s}_{1i}$ are extra variables representing $\sqrt{c_{11}c_{ii}}\cos(\theta_1 - \theta_i)$ and $-\sqrt{c_{11}c_{ii}}\sin(\theta_1 - \theta_i)$, respectively, for $i = 3, \ldots, n-1$, and $\tilde{c}_{1i}$ and $\tilde{s}_{1i}$ for $i = 2$ and $i = n$ coincidence with the original variables c_{1i}, s_{1i}.

Proposition 7. *Suppose a cycle basis is given for the power network $\mathcal{N} = (\mathcal{B}, \mathcal{L})$. Then, constraints (48) for every 3-cycle in the cycle basis and constraints (54) for every cycle with length $n \geq 4$ in the cycle basis define a valid bilinear extended relaxation of OPF (10)-(11). Moreover, it implies that $\sum_{(i,j)\in C} \operatorname{atan2}(s_{ij}, c_{ij}) = 2\pi k$ for some integer k for each cycle in the cycle basis.*

Proof. First, let us prove that the proposed relaxation is valid for any feasible solution of the OPF formulation over any cycle C with length $|C| = n \geq 4$ in the cycle basis. Without loss of generality, assume that the buses in the cycle are numbered from 1 to n. Let (c, s, θ) be a feasible solution for OPF (10)-(11). Recall that that we have $c_{i,i+1} = \sqrt{c_{ii}c_{i+1,i+1}}\cos(\theta_i - \theta_{i+1})$ and $s_{i,i+1} = -\sqrt{c_{ii}c_{i+1,i+1}}\sin(\theta_i - \theta_{i+1})$ for $i = 1, \ldots, n-1$. Then, choose $\tilde{c}_{1i} = \sqrt{c_{11}c_{ii}}\cos(\theta_1 - \theta_i)$, $\tilde{s}_{1i} = -\sqrt{c_{11}c_{ii}}\sin(\theta_1 - \theta_i)$ for each artificial line $(1, i)$ for $i = 2, \ldots, n-1$. Now, we have

$$
\begin{aligned}
&\tilde{s}_{1,i}c_{i+1,i+1} + s_{i,i+1}\tilde{c}_{1,i+1} - \tilde{s}_{1,i+1}c_{i,i+1} \\
&= [-\sqrt{c_{11}c_{ii}}\sin(\theta_1 - \theta_i)]c_{i+1,i+1} + [-\sqrt{c_{ii}c_{i+1,i+1}}\sin(\theta_i - \theta_{i+1}][\sqrt{c_{11}c_{ii}}\cos(\theta_1 - \theta_i)] \\
&\quad + [-\sqrt{c_{11}c_{i+1,i+1}}\sin(\theta_1 - \theta_{i+1})][\sqrt{c_{ii}c_{i+1,i+1}}\cos(\theta_i - \theta_{i+1})] \\
&= c_{i+1,i+1}\sqrt{c_{11}c_{ii}}\big[\sin(\theta_1 - \theta_i) - \sin(\theta_i - \theta_{i+1})\cos(\theta_1 - \theta_i) + \cos(\theta_i - \theta_{i+1})\sin(\theta_1 - \theta_{i+1}\big] \\
&= 0,
\end{aligned}
$$

which proves the validity of (54a). A similar argument can be used to prove the validity of (54b) as well. Also, since we have $\tilde{c}_{1,i}^2 + \tilde{s}_{1,i}^2 - c_{11}c_{ii} = (\sqrt{c_{11}c_{ii}}\cos(\theta_1 - \theta_i))^2 + (-\sqrt{c_{11}c_{ii}}\sin(\theta_1 - \theta_i))^2 - c_{11}c_{ii} = c_{11}c_{ii}[\cos^2(\theta_1 - \theta_i) + \sin^2(\theta_1 - \theta_i) - 1] = 0$, (54c) follows. Hence, constraints (54) for every cycle with length $n \geq 4$ are valid for OPF (10)-(11).

For the second part, it is sufficient to show that adding (54) to (10) implies $\cos\left(\sum_{i=1}^{n-1}\theta_{i,i+1} + \theta_{n1}\right) = 1$ for every cycle C with length $n \geq 4$. Using the argument in Proposition 5, we know that the equalities in (54) and (10f) enforce the cycle

constraint over the 3-cycle $(1, i, i+1)$

$$\theta_{1,i} + \theta_{i,i+1} + \theta_{i+1,1} = 2\pi k_i \qquad i = 2, \cdots, n-1, \tag{55}$$

for some integers k_i. Therefore, summing (55) over all i and canceling $\theta_{1,i} + \theta_{i,1} = 0$, we conclude that $\sum_{i=1}^{n-1} \theta_{i,i+1} + \theta_{n1} = 2\pi k$, for some $k \in \mathbb{Z}$. □

4-decomposition of a cycle C. Suppose $|C| \geq 5$ and odd. We can decompose the cycle C into 4-cycles by creating artificial edges $(1, 2i)$ for $i = 2, 3, \ldots, \frac{n-1}{2}$ and one 3-cycle. Now, we apply the exact reformulation from Section 3.3.1.1 and 3.3.1.2 for each of these cycles. Finally, polynomial p_n is replaced by the following set of bilinear equalities:

$$\tilde{c}_{1,2i-2}c_{2i-1,2i} - \tilde{s}_{1,2i-2}s_{2i-1,2i} - \tilde{c}_{1,2i}c_{2i-2,2i-1} - \tilde{s}_{1,2i}s_{2i-2,2i-1} = 0 \quad i = 2, \ldots, \frac{n-1}{2} \tag{56a}$$

$$\tilde{s}_{1,2i-2}c_{2i-1,2i} + \tilde{c}_{1,2i-2}s_{2i-1,2i} - \tilde{s}_{1,2i}c_{2i-2,2i-1} + \tilde{c}_{1,2i}s_{2i-2,2i-1} = 0 \quad i = 2, \ldots, \frac{n-1}{2} \tag{56b}$$

$$\tilde{c}_{1,n-1}c_{n-1,n} - \tilde{s}_{1,n-1}s_{n-1,n} - c_{n,1}c_{n-1,n-1} = 0 \tag{56c}$$

$$\tilde{s}_{1,n-1}c_{n-1,n} + \tilde{c}_{1,n-1}s_{n-1,n} + s_{n,1}c_{n-1,n-1} = 0 \tag{56d}$$

$$\tilde{s}_{1,i}^2 + \tilde{c}_{1,i}^2 = c_{11}c_{ii} \qquad i = 2, 3, \ldots, n-1 \tag{56e}$$

$$c_{ij}^2 + s_{ij}^2 = c_{ii}c_{jj} \qquad (i,j) \in C, \tag{56f}$$

where (56a)-(56b) are constraints on 4-cycles and (56c)-(56d) are constraints on the last 3-cycle, $\tilde{c}_{1,2i}, \tilde{s}_{1,2i}$ for $i = 2, 3, \ldots, (n-1)/2$ are additional variables and $\tilde{c}_{12}, \tilde{s}_{12}$ coincide with the original variables c_{12}, s_{12}.

Suppose $|C| \geq 6$ and even. We can decompose the cycle C into 4-cycles by creating the artificial edges $(1, 2i)$ for $i = 2, \ldots, \frac{n-2}{2}$. Now, we apply the exact reformulation from Section 3.3.1.2 for each of these cycles. Finally, polynomial p_n is replaced by

the following set of bilinear equalities:

$$\tilde{c}_{1,2i-2}c_{2i-1,2i} - \tilde{s}_{1,2i-2}s_{2i-1,2i} - \tilde{c}_{1,2i}c_{2i-2,2i-1} - \tilde{s}_{1,2i}s_{2i-2,2i-1} = 0 \qquad i = 2, \dots, \frac{n}{2} \tag{57a}$$

$$\tilde{s}_{1,2i-2}c_{2i-1,2i} + \tilde{c}_{1,2i-2}s_{2i-1,2i} - \tilde{s}_{1,2i}c_{2i-2,2i-1} + \tilde{c}_{1,2i}s_{2i-2,2i-1} = 0 \qquad i = 2, \dots, \frac{n}{2} \tag{57b}$$

$$\tilde{s}_{1,i}^2 + \tilde{c}_{1,i}^2 = c_{1,1}c_{i,i} \qquad i = 2, 3, \dots, n-1 \tag{57c}$$

$$c_{ij}^2 + s_{ij}^2 = c_{ii}c_{jj} \qquad (i,j) \in C, \tag{57d}$$

where $\tilde{c}_{1,2i}, \tilde{s}_{1,2i}$ are additional variables for $i = 2, \dots, n/2 - 1$, and $\tilde{c}_{1,n}, \tilde{s}_{1,n}$ coincide with the original variables $c_{1,n}, s_{1,n}$.

Proposition 8. *Suppose a cycle basis is given for the power network $\mathcal{N} = (\mathcal{B}, \mathcal{L})$. Then,*

(i) Constraints (48) for every 3-cycle in the cycle basis,

(ii) Constraints (52) for every 4-cycle in the cycle basis,

(iii) Constraints (56) *for every odd cycle with length $n \geq 5$ in the cycle basis,*

(iv) Constraints (57) *for every even cycle with length $n \geq 6$ in the cycle basis,*

define a valid bilinear extended relaxation of OPF (10)-(11)*. Moreover, it implies that $\sum_{(i,j)\in C} \operatorname{atan2}(s_{ij}, c_{ij}) = 2\pi k$ for some integer k for each cycle in the cycle basis.*

The proof is similar to the proof of Proposition 7.

3.3.1.4 McCormick Based LP Relaxation and Separation.

The cycle-based relaxations presented in Proposition 7 and Proposition 8 are non-convex quadratic problems, for which we can obtain LP relaxation by using McCormick relaxations of the bilinear constraints over cycles. For large networks, including McCormick relaxations for all the cycle constraints may be computationally inefficient. Therefore, we propose a separation routine which generates cutting planes to separate a solution of the classic SOCP relaxation from the McCormick envelopes of the cycle constraints. The separation is applied to every cycle in the cycle basis individually.

For a given cycle C, the McCormick relaxation of the bilinear cycle constraints, which could be any one of: (48), (52), (54), (56), or (57), can be written compactly as follows:

$$Az + \tilde{A}\tilde{z} + By \leq c \tag{58a}$$

$$Ey = 0, \tag{58b}$$

where z is a vector composed of the c, s variables in the alternative formulation (10), $\tilde{z}$ is a vector composed of the additional $\tilde{c}, \tilde{s}$ variables introduced in the cycle decomposition, and y is a vector of new variables defined to linearize the bilinear terms in the cycle constraints. Constraint (58a) contains the McCormick envelopes of the bilinear terms and bounds on the c, s variables, while (58b) includes the linearized cycle equality constraints.

Given an optimal solution z^* of the classic SOCP relaxation $\mathcal{A}^*_{SOCP}$, we can solve

the following separation problem for a cycle C,

$$
\begin{aligned}
v^* := \min_{\alpha,\beta,\gamma,\mu,\lambda} \quad & \beta - \alpha^T z^* && \text{(59a)}\\
\text{s.t.} \quad & A^T\lambda = \alpha && \text{(59b)}\\
& \tilde{A}^T\lambda = 0 && \text{(59c)}\\
& B^T\lambda + E^T\mu = 0 && \text{(59d)}\\
& c^T\lambda \le \beta, \;\; \lambda \ge 0 && \text{(59e)}\\
& -e \le \alpha \le e && \text{(59f)}\\
& -1 \le \beta \le 1, && \text{(59g)}
\end{aligned}
$$

where (59b)-(59e) is the dual system equivalent to the condition that $\alpha^T z \le \beta$ for all $(z, \tilde{z}, y)$ satisfies (58); (59f)-(59g) bounds the coefficients α, β, and e is the vector of 1's. If $v^* < 0$, then the corresponding optimal solution (α, β) of (59) gives a separating hyperplane such that $\alpha^T z^* > \beta$ and $\alpha^T z \le \beta$ for all $(z, \tilde{z}, y)$ in (58). If $v^* \ge 0$, then (59) certifies that $(z^*, \tilde{z}, y)$ is contained in the McCormick relaxation (58) for some $\tilde{z}, y$.

We remark that the McCormick relaxations obtained from 3-decomposition and 4-decomposition of cycles do not dominate one another. Therefore, we use both of them in the separation routine. We also note that the classic SOCP relaxation strengthened by dynamically adding valid inequalities through separation over the McCormick relaxations of the cycle constraints is incomparable to the standard SDP relaxations of OPF.

Proposition 9. *$\mathcal{A}^*_{SOCP}$ strengthened by the valid inequalities from the McCormick relaxation of the cycle constraints is not dominated by nor dominates $\mathcal{R}_{SDP}$.*

This result is verified by an example in Section 3.4.4.

3.3.2 Arctangent Envelopes

In this section, we propose a second approach to strengthen the classic SOCP relaxation $\mathcal{A}^*_{SOCP}$. The key idea is still to incorporate convex approximation of the angle condition (11) to the SOCP relaxation. This time, instead of reformulating polynomial constraints over cycles such as (54), (56), and (57), we propose linear envelopes for the arctangent function over a box, and incorporate this relaxation to $\mathcal{A}^*_{SOCP}$.

Our construction uses four linear inequalities to approximate the convex envelope for the following set defined by the arctangent constraint (11) for each line $(i, j) \in \mathcal{L}$,

$$\mathcal{AT} := \left\{(c, s, \theta) \in \mathbb{R}^3 : \theta = \arctan\left(\frac{s}{c}\right), (c, s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}]\right\}, \tag{60}$$

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

$$\begin{aligned} \zeta^1 &= (\underline{c}, \overline{s}, \arctan(\overline{s}/\underline{c})), \quad \zeta^2 = (\overline{c}, \overline{s}, \arctan(\overline{s}/\overline{c})), \\ \zeta^3 &= (\overline{c}, \underline{s}, \arctan(\underline{s}/\overline{c})), \quad \zeta^4 = (\underline{c}, \underline{s}, \arctan(\underline{s}/\underline{c})). \end{aligned} \tag{61}$$

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

Proposition 10. *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*

$$\gamma'_k + \alpha_k c + \beta_k s \geq \arctan\left(\frac{s}{c}\right) \tag{62}$$

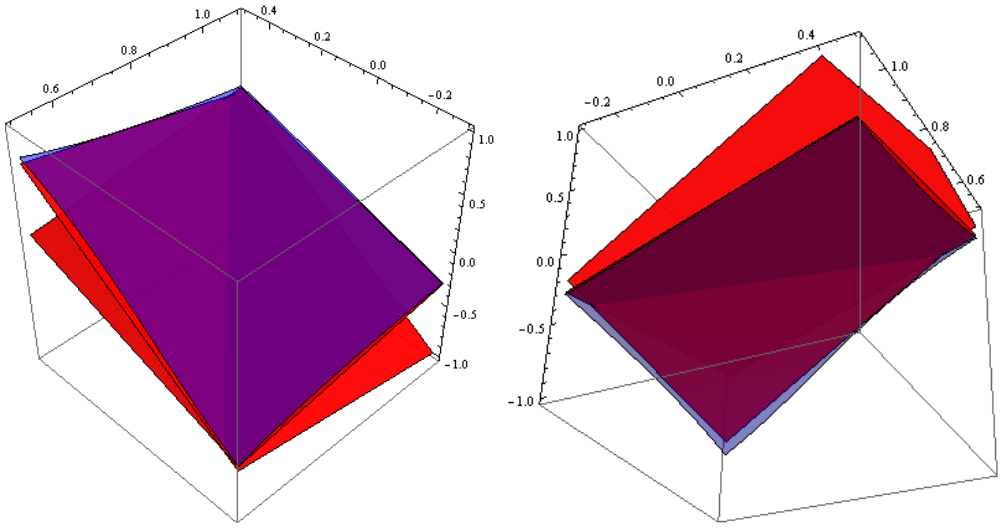
for all $(c, s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}]$ with $\gamma'_k = \gamma_k + \Delta\gamma_k$, where

$$\Delta\gamma_k = \max\left\{\arctan\left(\frac{s}{c}\right) - (\gamma_k + \alpha_k c + \beta_k s) : c \in [\underline{c}, \overline{c}],\ s \in [\underline{s}, \overline{s}]\right\}, \tag{63}$$

for $k = 1, 2$.

Note that by the construction of (63), it is evident that $\gamma'_k + \alpha_k c + \beta_k s$ dominates the $\arctan(s/c)$ over the box. The nonconvex optimization problem (63) can be solved by enumerating all possible Karush-Kuhn-Tucker (KKT) points.

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



(a) Lower envelopes. (b) Upper envelopes.

Figure 6: Arctangent envelopes from different viewpoints. Red planes are the envelopes.

Proposition 11. *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*

$$\gamma'_k + \alpha_k c + \beta_k s \leq \arctan\left(\frac{s}{c}\right) \tag{64}$$

for all $(c, s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}]$ with $\gamma'_k = \gamma_k - \Delta\gamma_k$, where

$$\Delta\gamma_k = \max\left\{(\gamma_k + \alpha_k c + \beta_k s) - \arctan\left(\frac{s}{c}\right) : c \in [\underline{c}, \overline{c}],\ s \in [\underline{s}, \overline{s}]\right\}, \tag{65}$$

for $k = 3, 4$.

Figure 6 shows an example of these upper and lower envelopes. One may further strengthen these envelopes via additional inequalities, although the benefit is minimal according to our experiments.

We have the following proposition, whose proof is provided by an example in Section 3.4.4.

Proposition 12. *$\mathcal{A}^*_{SOCP}$ strengthened by arctangent envelopes defined in* (152)-(106) *is not dominated by nor dominates the SDP relaxation $\mathcal{R}_{SDP}$.*

3.3.3 SDP Separation

The last approach we propose to strengthen the classic SOCP relaxation is similar in spirit to the separation approach in Section 3.3.1.4, but here, instead of separating over the McCormick relaxations of the cycle constraints (58), we separate a given SOCP relaxation solution from the feasible region of the SDP relaxation of cycles. In the following, we first explore the relationship between the classic SOCP relaxation $\mathcal{A}^*_{SOCP}$ and the SDP relaxation $\mathcal{R}^r_{SDP}$. Then, we explain the separation procedure over cycles.

Let $x \in \mathbb{R}^{2|\mathcal{B}|}$ be a vector of bus voltages defined as $x = [e; f]$ such that $x_i = e_i$ for $i \in \mathcal{B}$ and $x_{i'} = f_i$ for $i' = i + |\mathcal{B}|$. Observe that if we have a set of c, s variables satisfying the cosine, sine definition (45) and a matrix variable $W = xx^T$, then the following linear relationship between c, s and W holds,

$$c_{ij} = e_i e_j + f_i f_j = W_{ij} + W_{i'j'} \qquad (i,j) \in \mathcal{L} \tag{66a}$$

$$s_{ij} = e_i f_j - e_j f_i = W_{ij'} - W_{ji'} \qquad (i,j) \in \mathcal{L} \tag{66b}$$

$$c_{ii} = e_i^2 + f_i^2 = W_{ii} + W_{i'i'}, \qquad i \in \mathcal{B}. \tag{66c}$$

Given a solution of the classic SOCP relaxation, denoted as (p^*, q^*, c^*, s^*), if there exists a symmetric matrix $W^* \in \mathbb{R}^{2|\mathcal{B}| \times 2|\mathcal{B}|}$ such that (c^*, s^*) and W^* satisfy the linear system (66), then (p^*, q^*, W^*) satisfies the flow conservation and voltage bound constraints (31a)-(31c) (because (c^*, s^*) satisfies (10b)-(10d)) as well as the generator real and reactive bounds (1h)-(1i). If, furthermore, $W^* \succeq 0$, then (p^*, q^*, W^*) is a feasible solution to the standard SDP relaxation $\mathcal{R}^r_{SDP}$, therefore, optimal for the SDP relaxation. If there does not exists such a W^*, then we can add a valid inequality to the SOCP relaxation to separate (c^*, s^*) from the set defined by (66) and $W \succeq 0$. This procedure can be repeated until the optimal SDP relaxation solution is obtained.

Notice that the above separation procedure requires solving an SDP problem with a matrix of the size equal to the original SDP relaxation, which can be quite time

consuming. Instead of separating over the full matrix, or equivalently the entire power network, we can consider a separation only over cycles. In this way, we effectively use an SDP relaxation to provide a convex approximation of the angle condition (11) over a cycle.

In particular, for any cycle C in the power network, we only consider the equalities in (66) associated with this cycle and the corresponding submatrix matrix $\tilde{W} \in \mathbb{R}^{2|C|\times 2|C|}$ of W. Consider the following set $\mathcal{S}$

$$\mathcal{S} := \{z \in \mathbb{R}^{2|C|} : \exists \tilde{W} \in \mathbb{R}^{2|C|\times 2|C|} \text{ s.t. } -z_l + A_l \bullet \tilde{W} = 0 \quad \forall l \in L, \quad \tilde{W} \succeq 0\}, \quad (67)$$

where $z = (c, s)$, the linear equality represents the linear system (66) restricted to the cycle C, and L is the index set for all these equalities; $\bullet$ denotes the Frobenius inner product between matrices. We suppress labeling variables with C for conciseness. It should be understood that the construction is done for each cycle in a cycle basis.

For a given z^*, the separation problem over $\mathcal{S}$ can be written as follows,

$$v^* := \min \quad -\alpha^T z^* \quad (68a)$$

$$\text{s.t.} \quad \sum_{l\in L} \lambda_l A_l \succeq 0 \quad (68b)$$

$$\alpha + \lambda = 0 \quad (68c)$$

$$-e \leq \alpha \leq e. \quad (68d)$$

Since the system (67) is strictly feasible (e.g., $\tilde{W} = I$ and $z_l = A_l \cdot I$, for $l \in L$), strong duality holds between the primal system in $\mathcal{S}$ and the dual system in (68). In particular, (68b)-(68c) is equivalent to $\max_{z\in\mathcal{S}} \alpha^T z \leq 0$. Therefore, the solution of (68) either gives a separating hyperplane of the form $\alpha^T z \leq 0$ such that $\alpha^T z^* > 0$ and $\alpha^T z \leq 0 \ \ \forall z \in \mathcal{S}$, or certifies that $z^* \in \mathcal{S}$. If the optimal objective value v^* of (68) is strictly less than 0, then we add the homogeneous inequality $\alpha^T z \leq 0$ to the classic SOCP relaxation. We can now apply this procedure to every element of a cycle basis and resolve SOCP with the added linear inequalities. In computational experiments,

we observe that a few iterations of this algorithm give very tight approximations to SDP relaxation of rectangular formulation.

3.3.4 Obtaining Variable Bounds

The proposed McCormick relaxations of the cycle constraints and the convex envelopes for the arctangent functions are useful only when good variable upper and lower bounds are available for the c and s variables. In this section, we explain how to obtain good bounds, which is the key ingredient in the success of the first two proposed methods.

Recall that c_{ij} and s_{ij} do not have explicit variable bounds except the implied bounds in (24). However, these bounds may be loose, since it is usually the case that phase angle differences in a power network under normal operation are small, implying $c_{ij} \approx 1$ and $s_{ij} \approx 0$. Therefore, one can try to improve these bounds. A straightforward approach is to optimize c_{ij} and s_{ij} over the feasible region of the SOCP relaxation as is proposed in Section 2.6. However, this procedure can be expensive because we need to solve $4|\mathcal{L}|$ SOCPs, each of the size of the classic SOCP relaxation.

To be computationally efficient, instead of solving the full size SOCPs to tighten variable bounds, we can obtain potentially weaker bounds by solving a reduced version of the full SOCP relaxation. In particular, to find variable bounds for c_{kl} and s_{kl} for some $(k,l) \in \mathcal{L}$, consider the buses which can be reached from either k or l in at most r steps. Denote these buses by a set $\mathcal{B}_{kl}(r)$. For instance, $\mathcal{B}_{kl}(0) = \{k,l\}$, $\mathcal{B}_{kl}(1) = \delta(k) \cup \delta(l)$, etc. Also define $\mathcal{G}_{kl}(r) = \mathcal{B}_{kl}(r) \cap \mathcal{G}$ and $\mathcal{L}_{kl}(r) = \{(i,j) \in \mathcal{L} :$

$i \in \mathcal{B}_{kl}(r)$ or $j \in \mathcal{B}_{kl}(r)\}$. Consider the following SOCP relaxation,

$$p_i^g - p_i^d = G_{ii}c_{ii} + \sum_{j\in\delta(i)} [G_{ij}c_{ij} - B_{ij}s_{ij}] \qquad i \in \mathcal{B}_{kl}(r) \tag{69a}$$

$$q_i^g - q_i^d = -B_{ii}c_{ii} + \sum_{j\in\delta(i)} [-B_{ij}c_{ij} - G_{ij}s_{ij}] \qquad i \in \mathcal{B}_{kl}(r) \tag{69b}$$

$$\underline{V}_i^2 \leq c_{ii} \leq \overline{V}_i^2 \qquad i \in \mathcal{B}_{kl}(r+1) \tag{69c}$$

$$p_i^{\min} \leq p_i^g \leq p_i^{\max} \qquad i \in \mathcal{G}_{kl}(r) \tag{69d}$$

$$q_i^{\min} \leq q_i^g \leq q_i^{\max} \qquad i \in \mathcal{G}_{kl}(r) \tag{69e}$$

$$c_{ij} = c_{ji}, \quad s_{ij} = -s_{ji} \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{69f}$$

$$c_{ij}^2 + s_{ij}^2 \leq c_{ii}c_{jj} \qquad (i,j) \in \mathcal{L}_{kl}(r). \tag{69g}$$

Essentially, (69) is the classic SOCP relaxation applied to the part of the power network within r steps of the buses k and l. Note that (69) for each edge (k,l) can be solved in parallel, since they are independent of each other. We observed that a good trade-off between accuracy and speed is to select $r = 2$. In our experiments, larger values of r improve variable bounds marginally.

For artificial edges, we cannot use the above procedure as they do not appear in the flow balance constraints. Instead, we use bounds on the original variables that are computed through (69) to obtain some improved bounds for the variables on the artificial edges. Since any large cycle can be decomposed into 3-cycles and/or 4-cycles as shown in Section 3.3.1.3, we only need to consider 3- and 4-cycles here. Let us start from a 3-cycle. Assume the upper and lower bounds on $c_{12}, s_{12}, c_{23}, s_{23}$ are already known, and we want to tighten the bounds on the artificial edge c_{13}, s_{13}. Then, the bilinear constraints (48) over the cycle can be written as follows:

$$c_{13} = \frac{c_{12}c_{23} - s_{12}s_{23}}{c_{22}} \tag{70a}$$

$$s_{13} = \frac{s_{12}c_{23} + c_{12}s_{23}}{c_{22}}. \tag{70b}$$

Now, we can obtain variable bounds on c_{13}, s_{13} by bounding the right-hand sides of

(70a)-(70b) over the box for $c_{12}, s_{12}, c_{23}, s_{23}$, and c_{22}. In particular, an upper bound on c_{13} can be computed as

$$\bar{c}_{13} = \begin{cases} \hat{c}_{13}/\underline{c}_{22} & \text{if } \hat{c}_{13} > 0 \\ \hat{c}_{13}/\overline{c}_{22} & \text{if } \hat{c}_{13} \leq 0, \end{cases} \tag{71}$$

where

$$\begin{aligned} \hat{c}_{13} = & \max\{c_{12}c_{23} : c_{12} \in [\underline{c}_{12}, \overline{c}_{12}], c_{23} \in [\underline{c}_{23}, \overline{c}_{23}]\} \\ & - \min\{s_{12}s_{23} : s_{12} \in [\underline{s}_{12}, \overline{s}_{12}], s_{23} \in [\underline{s}_{23}, \overline{s}_{23}]\}. \end{aligned} \tag{72}$$

A similar procedure can be applied to obtain lower bounds on c_{13} and s_{13}.

For a 4-cycle of buses $1, 2, 3, 4$, assume we have bounds on $c_{12}, s_{12}, c_{23}, s_{23}, c_{34}, s_{34}$, and want to find variable bounds on the artificial edge c_{14}, s_{14}. Using the two bilinear constraints for the 4-cycle in (52) and (10f), we can express c_{14} and s_{14} in terms of the other variables as follows

$$c_{14} = \frac{c_{12}(c_{23}c_{34} - s_{23}s_{34}) - s_{12}(s_{23}c_{34} + c_{23}s_{34})}{c_{22}c_{33}} \tag{73a}$$

$$s_{14} = \frac{s_{12}(c_{23}c_{34} - s_{23}s_{34}) + c_{12}(s_{23}c_{34} + c_{23}s_{34})}{c_{22}c_{33}}. \tag{73b}$$

Now, proceed in two steps. (1) Define $a := c_{23}c_{34} - s_{23}s_{34}$ and $b := s_{23}c_{34} + c_{23}s_{34}$, and calculate bounds a, b as described for the 3-cycle case. (2) Repeat this process to obtain bounds on c_{14}, s_{14}.

3.4 Computational Experiments

In this section, we present the results of extensive computational experiments on standard IEEE instances available from MATPOWER [111] and instances from NESTA 0.3.0 archive [25]. The code is written in the C# language with Visual Studio 2010 as the compiler. For all experiments, a 64-bit laptop with Intel Core i7 CPU with 2.00GHz processor and 8 GB RAM is used. Time is measured in seconds, unless otherwise stated. Conic interior point solver MOSEK 7.1 [2] is used to solve SOCPs and SDPs.

3.4.1 Methods

We report the results of the following four algorithmic settings:

- SOCP: The classic SOCP formulation $\mathcal{A}^*_{SOCP}$ without any improvement.
- SOCPA: SOCP strengthened by the arctangent envelopes (152)-(106).
- S34A: SOCPA further strengthened by dynamically generating linear valid inequalities from the McCormick relaxation of the cycle constraints via the 3- and 4-cycle decompositions and the separation routine developed in Sections 3.3.1.3 and 3.3.1.4.
- SSDP: SOCP strengthened by dynamically generating linear valid inequalities obtained from separating an SOCP feasible solution from the SDP relaxation over cycles. The separation routine is developed in Section 3.3.3.

We note that SOCPA and S34A require preprocessing to improve variable bounds on the c and s variables as developed in Section 3.3.4. This process is parallelized but still it constitutes a sizable portion in the computational cost of the method. Constraint generation procedures are also parallelized, since each separation problem is defined for a different cycle in the cycle basis. We use a Gaussian elimination based approach to construct a cycle basis proposed in Appendix B. We repeat the constraint generation algorithm for five iterations or terminate when there are no cuts to be added.

In the following, we compare the above four methods with the SDP relaxation based approaches in Section 3.4.2 and with a recent quadratic convex relaxation approach in Section 3.4.3. We also show that the proposed methods are not dominated by nor dominate the SDP relaxations in Section 3.4.4. Finally, in Section 3.4.5 we demonstrate the robustness of the proposed methods by solving randomly perturbed instances from the standard IEEE instances.

3.4.2 Comparison to SDP Relaxation Based Methods

It is well known in the power systems literature that SDP relaxations have small duality gaps for the standard IEEE instances. However, the computational burden of SDP relaxations is typically very high. Chordal extensions and matrix completion type methods are used to significantly accelerate the solution time of the SDP relaxations. A publicly available implementation is called OPF Solver [73]. This package exploits the sparsity of underlying network to solve large-scale SDPs more efficiently as discussed in [74, 75]. In this section, we compare the accuracy and performance of the four proposed SOCP relaxation based methods to the SDP relaxation implemented in OPF Solver.

3.4.2.1 Lower Bound and Computation Time Comparison

We first compare the computation time and the lower bounds obtained by the SDP relaxation with those obtained by the four types of SOCP relaxations. Table 7 shows the results. Here, "ratio" is defined as the lower bound of an SOCP relaxation divided by that of the SDP relaxation. We can see that the arctangent envelopes in SOCPA give non-trivial strengthening to the classic SOCP relaxation SOCP. On the other hand, having the arctangent envelopes, the effect of the valid inequalities due to the McCormick relaxation of the cycle constraints is small. The SDP separation approach, SSDP, is the most successful among the four methods, which achieves the same lower bound as the SDP relaxation in nine instances and provides very tight bounds for the others (99.96% on average). In terms of computational time, SOCPA, S34A, and SSDP are roughly one order-of-magnitude faster than the SDP relaxation for large problems (2383-bus and above). We also note that OPF Solver does not support instances with reactive power cost functions, hence the case9Q and case30Q instances are solved using the standard rectangular SDP formulation. The largest instance case3375wp requires at least 3 hours to even construct the SDP model.

Table 7: Comparison of lower bounds and computation time.

	SDP	SOCP		SOCPA		S34A		SSDP	
case	time	ratio	time	ratio	time	ratio	time	ratio	time
6ww	1.66	0.9937	0.02	0.9998	0.40	0.9999	0.43	1.0000	0.46
9	0.84	1.0000	0.02	1.0000	0.17	1.0000	0.18	1.0000	0.12
9Q	NS	1.0000	0.02	1.0000	0.18	1.0000	0.19	1.0000	0.12
14	1.07	0.9992	0.02	0.9992	0.41	0.9994	0.45	1.0000	0.64
ieee30	1.84	0.9996	0.03	0.9996	0.78	0.9996	0.84	1.0000	1.15
30	2.19	0.9943	0.06	0.9963	0.95	0.9966	1.07	0.9993	1.22
30Q	NS	0.9753	0.07	0.9765	1.02	0.9769	1.11	1.0000	1.32
39	2.20	0.9998	0.04	0.9999	0.90	0.9999	0.99	1.0000	0.72
57	2.60	0.9994	0.04	0.9994	1.43	0.9994	1.47	1.0000	2.14
118	4.58	0.9976	0.11	0.9976	3.69	0.9984	4.83	0.9997	5.19
300	9.81	0.9985	0.21	0.9988	7.62	0.9989	10.40	1.0000	9.83
2383wp	682.86	0.9932	7.11	0.9949	92.83	0.9950	130.03	0.9984	101.31
2736sp	853.92	0.9970	5.14	0.9977	90.93	0.9976	163.80	0.9994	94.48
2737sop	792.25	0.9974	3.85	0.9979	95.28	0.9979	158.80	0.9997	78.70
2746wop	1138.06	0.9963	4.35	0.9971	102.37	0.9973	180.42	0.9995	109.65
2746wp	941.04	0.9967	5.79	0.9975	109.82	0.9975	186.31	0.9998	102.16
3012wp	746.08	0.9936	7.28	0.9946	143.10	0.9946	185.56	0.9974	109.19
3120sp	904.90	0.9955	7.33	0.9962	127.90	0.9965	196.05	0.9987	103.77
3375wp	$>$ 3hr	NA	8.25	NA	149.03	NA	422.35	NA	133.62
Average	**380.37**	**0.9959**	**2.62**	**0.9968**	**48.88**	**0.9970**	**86.59**	**0.9996**	**45.04**

3.4.2.2 Upper Bound and Optimality Gap Comparison

In this part, we compare the quality of the feasible solutions to the original OPF problem derived from relaxation solutions of our approaches to that of OPF Solver. Let us first describe our method of finding an OPF feasible solution. The procedure is simple: we use an optimal solution of one of the SOCP relaxations as a starting point to the nonlinear interior point solver IPOPT [105], which produces a locally optimal solution to the OPF problem. We observed empirically that independent of the relaxation we use, the method always converges to the same OPF solution. We also note that this method gives the same OPF feasible solutions as MATPOWER.

OPF Solver utilizes the SDP relaxation solution to obtain OPF feasible solutions. When the optimal matrix variable is rank optimal, e.g., rank one in the SDP relaxation in the real domain (32), a vector of feasible voltages e, f can be easily derived by computing the leading eigenvalue and the corresponding eigenvector of the SDP

optimal matrix. However, when the rank is greater than one, it is difficult to put a physical meaning to the relaxation solution. OPF Solver uses a penalization approach to reduce the rank of the matrices in order to obtain nearly feasible solutions to OPF. In particular, the reactive power dispatch and the total apparent power on some lines are penalized with certain penalty coefficients. Empirical results show that these coefficients are problem dependent and fine-tuning seems to be essential to obtain high quality feasible solutions. In our comparison, we use the suggested penalty parameters in [74] and exclude the computational burden of fine-tuning these parameters.

We compare the OPF feasible solutions found by our methods against the nearly feasible solutions obtained by OPF Solver. The results are shown in Table 8. Here, "ratio" is calculated as the objective cost of an OPF feasible solution of our methods divided by that of OPF Solver. A ratio less than 1 means our approach produces a better OPF feasible solution than OPF Solver. The percentage optimality gap, "%gap", is calculated as $\%\text{gap} = 100 \times \frac{z^{\text{UB}} - z^{\text{LB}}}{z^{\text{UB}}}$, where z^{UB} is the objective cost of an OPF feasible solution derived from a relaxation, and z^{LB} is the optimal objective cost of this relaxation. The total computation time, reported as "time", includes the time solving the corresponding relaxation and deriving a feasible solution to OPF. We observe that SSDP significantly closes the optimality gap to 0.08% or 99.92% to the global optimum on average, improving over the SDP relaxation's 0.19%. The ratio of upper bounds is less than 1 for large systems, which implies the quality of the OPF feasible solutions obtained by the penalization method in OPF Solver are not as good as our approaches, even though best known penalty parameters are used. The reason is that the penalization method does not produce locally optimal solutions. This issue may perhaps be fixed by applying a local solver to improve the solution obtained from the penalization method at the cost of converting the optimal matrix variable to a vector of voltages and calling a local solver. We also

note that computing a feasible solution from the SDP relaxation is rather difficult, demonstrated by the large computational time of the SDP column, whereas SOCP is about two orders of magnitude faster than SDP, and SOCPA, S34A, SSDP are roughly one order of magnitude faster.

We also compare the SDP bound with the feasible solution found by our SOCP based methods and calculate the percentage optimality gap. Under an optimality threshold of 0.01%, we observe that SDP is tight for 14 instances out of 19 (the gaps for cases 9Q, 2383wp, 3012wp, 3120sp and 3375wp are respectively 0.04%, 0.37%, 0.15%, 0.09%, NA). We note that our SOCP based relaxations are not as successful according to this comparison. SOCP, SOCPA, S34A and SSDP are tight for 1, 2, 3 and 8 instances, respectively. Nevertheless, we remind the reader that SOCP based methods have small optimality gap (e.g. 0.08% on the average for SSDP) as can be seen from Table 8.

Table 8: Comparison of upper bounds and percentage optimality gap.

	SDP			SOCP		SOCPA		S34A		SSDP	
case	%gap	time	ratio	%gap	time	%gap	time	%gap	time	%gap	time
6ww	NA	NR	NA	0.63	0.13	0.02	0.48	0.01	0.45	0.00	0.53
9	NA	NR	NA	0.00	0.04	0.00	0.19	0.00	0.20	0.00	0.17
9Q	NA	NR	NA	0.04	0.04	0.04	0.20	0.04	0.21	0.04	0.17
14	0.00	4.49	1.0000	0.08	0.05	0.08	0.44	0.06	0.48	0.00	0.68
ieee30	NA	NR	NA	0.04	0.07	0.04	0.83	0.04	0.88	0.00	1.20
30	0.00	6.54	1.0000	0.57	0.12	0.37	1.01	0.34	1.13	0.07	1.28
30Q	NA	NR	NA	2.48	0.11	2.35	1.07	2.32	1.16	0.00	1.36
39	0.01	5.09	1.0000	0.02	0.10	0.01	0.96	0.01	1.05	0.01	0.78
57	0.00	6.68	1.0000	0.06	0.11	0.06	1.50	0.06	1.55	0.00	2.22
118	0.00	11.16	1.0000	0.25	0.27	0.24	3.86	0.16	5.00	0.03	5.34
300	0.00	22.65	1.0000	0.15	0.62	0.12	8.04	0.11	10.83	0.00	10.33
2383wp	0.68	911.47	0.9969	1.05	21.39	0.89	104.71	0.88	145.29	0.54	124.34
2736sp	0.03	1181.09	0.9997	0.30	16.15	0.23	97.37	0.24	170.92	0.06	114.81
2737sop	0.00	1093.29	1.0000	0.26	12.05	0.21	102.27	0.21	167.59	0.03	103.81
2746wop	0.01	1470.10	0.9999	0.37	9.19	0.29	108.53	0.27	186.91	0.05	138.39
2746wp	0.04	1251.95	0.9996	0.33	14.08	0.25	116.18	0.25	193.91	0.02	124.07
3012wp	0.81	1314.16	0.9934	0.79	19.65	0.70	154.72	0.70	195.56	0.41	134.19
3120sp	0.93	1633.28	0.9916	0.54	16.14	0.47	137.70	0.44	206.20	0.22	121.77
3375wp	NA	>3hr	NA	0.26	18.66	0.24	158.21	0.23	431.87	0.13	157.20
Average	**0.19**	**685.53**	**0.9985**	**0.43**	**6.79**	**0.35**	**52.54**	**0.34**	**90.59**	**0.08**	**54.88**

3.4.3 Comparison to Other SOCP Based Methods

Now, we compare the strength of our SOCP based relaxations to other similar methods. A recent work utilizing SOCP relaxations is [27], in which a Quadratic Convex (QC) relaxation of OPF is proposed. It is empirically observed that the phase angles of neighboring buses in a power network are usually close to each other in OPF problems and the QC relaxation is specialized to take advantage of this observation. However, very tight angle bounds are typically not available in practice and choosing very small angles may restrict the feasible region of the OPF problem. In this regards, we remind the reader that our proposed methods do not depend on the availability of such tight angle bounds and our methods use a preprocessing procedure to obtain bounds on the c and s variables. Explicit angle difference bounds can be incorporated into the SOCP relaxations by addition of the following constraints for $(i, j) \in \mathcal{L}$,

$$-\tan(\overline{\theta}_{ij})c_{ij} \leq s_{ij} \leq \tan(\overline{\theta}_{ij})c_{ij} \tag{74}$$

and

$$-\overline{\theta}_{ij} \leq \theta_i - \theta_j \leq \overline{\theta}_{ij}, \tag{75}$$

where $\overline{\theta}_{ij}$ is the maximum absolute difference between phase angles at buses i and j. We also note that although the bounding techniques in Section 3.3.4 and the arctangent envelopes in SOCPA, S34A may be adapted to exploit the availability of such bounds, we choose not to do so in the experiments.

In Table 9, we compare the percentage optimality gaps of all the NESTA instances obtained by our methods and those achieved by the QC approach reported in [27]. The percentage optimality gap is defined the same as "%gap" in the previous section. For the QC results, only instances with an optimality gap more than 1% are reported in [27]. Those instances of optimal gaps less than 1% are indicated by blanks in Table 9. The average optimality gap for the QC approach is taken over instances with reported values. The NESTA library has three types of instances, namely, the typical operating

conditions, congested operating conditions, and small angle difference conditions [27].

From Table 9 we have the following observations.

(i) For instances from Typical Operating Conditions, each of our three strong SOCP relaxations dominates QC for all instances, except for the 3-bus instance 3lmbd SOCPA has an optimality gap 1.25% comparing to QC's 1.24%. For all the instances where QC achieves an optimality gap less than 1%, SOCP relaxations also achieve less than 1% gaps, except for the 1460wp instance, for which a gap slightly higher than 1% is obtained by the strong SOCP relaxations. The SSDP approach significantly outperforms QC in all instances and on average achieves 1.82% gap versus QC's 5.17%.

(ii) A similar picture holds for the Congested Operating Conditions, where the three strong SOCP relaxations dominate QC for all instances, except for SOCPA on the 3-bus instance 3lmbd. All three strong SOCP relaxations achieve less than 1% optimality gaps on all instances that QC achieves less than 1% gaps. SSDP again has the best performance and significantly outperforms QC.

(iii) For the instances from Small Angle Difference Conditions, which is a condition that is most suitable for QC, QC only dominates SSDP 4 times out of 19 instances reported in [27], and QC is better than all of strong SOCP relaxation in 3 out of the 19 instances. In terms of the average optimality gap, both SOCPA and S34A outperform QC.

Computational costs of different relaxation methods for NESTA instances are provided in Table 10.

Table 9: Comparison of percentage optimality gap for NESTA instances.

	Typical Operating Conditions					Congested Operating Conditions					Small Angle Difference Conditions				
case	SOCP	SOCPA	S34A	SSDP	QC	SOCP	SOCPA	S34A	SSDP	QC	SOCP	SOCPA	S34A	SSDP	QC
3lmbd	1.32	1.25	0.97	0.43	1.24	3.30	1.97	1.20	1.31	1.83	4.28	2.33	1.51	2.13	1.24
4gs	0.00	0.00	0.00	0.01		0.65	0.16	0.12	0.00		4.90	0.42	0.02	0.14	0.81
5pjm	14.54	14.47	14.26	6.22	14.54	0.45	0.11	0.06	0.00		3.61	0.45	0.34	0.01	1.10
6ww	0.63	0.02	0.01	0.00		13.33	0.35	0.14	0.00	13.14	0.80	0.02	0.01	0.00	
9wscc	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00		1.50	0.43	0.37	0.01	0.41
14ieee	0.11	0.11	0.07	0.00		1.35	1.32	1.32	0.00	1.35	0.07	0.06	0.06	0.00	
29edin	0.14	0.08	0.05	0.00		0.44	0.40	0.36	0.03		34.47	25.94	21.06	31.33	20.57
30as	0.06	0.05	0.05	0.00		4.76	2.02	1.89	1.72	4.76	9.16	2.43	2.36	0.95	3.07
30fsr	0.39	0.23	0.23	0.03		45.97	42.22	41.85	40.28	45.97	0.62	0.33	0.27	0.12	
30ieee	15.65	5.24	4.79	0.00	15.44	0.99	0.86	0.85	0.08		5.87	2.07	1.98	0.00	3.95
39epri	0.05	0.02	0.02	0.01		2.99	0.77	0.77	0.00	2.97	0.11	0.09	0.09	0.09	
57ieee	0.06	0.06	0.06	0.00		0.21	0.21	0.20	0.13		0.11	0.09	0.09	0.05	
118ieee	2.10	1.12	0.94	0.25	1.75	44.19	40.18	38.22	39.09	44.03	12.88	7.77	7.32	9.50	8.30
162ieee	4.19	3.99	3.95	3.50	4.17	1.52	1.44	1.43	1.20	1.51	7.06	5.94	5.81	6.36	6.88
189edin	0.22	0.22	0.22	0.07		5.59	3.34	3.33	0.22	5.56	2.27	2.21	2.25	1.23	2.24
300ieee	1.19	0.78	0.71	0.30	1.18	0.85	0.51	0.47	0.15		1.27	0.77	0.70	0.33	1.16
1460wp	1.22	1.18	1.18	1.04		1.10	0.98	0.84	0.68		1.37	1.33	1.32	1.22	
2224edin	6.22	4.30	4.25	4.60	6.16	3.16	2.51	2.43	2.58	3.15	6.43	3.91	3.87	4.80	5.79
2383wp	1.06	0.87	0.87	0.54	1.04	1.12	0.91	0.87	0.52	1.12	4.01	2.92	2.80	2.82	2.97
2736sp	0.30	0.21	0.20	0.08		1.33	1.14	1.12	0.91	1.32	2.34	1.86	1.86	1.92	2.01
2737sop	0.26	0.20	0.20	0.03		1.06	0.86	0.86	0.54	1.05	2.43	2.23	2.23	1.97	2.21
2746wop	0.37	0.28	0.27	0.06		0.49	0.35	0.34	0.17		2.94	2.30	2.31	2.60	1.83
2746wp	0.32	0.22	0.22	0.03		0.58	0.34	0.34	0.07		2.44	1.68	1.67	1.83	2.48
3012wp	1.04	0.90	0.89	0.50	1.01	1.25	0.90	0.89	0.58	1.24	2.14	2.00	1.96	1.54	1.92
3120sp	0.56	0.45	0.44	0.23		3.03	2.78	2.78	2.34	3.02	2.79	2.60	2.57	2.19	2.56
3375wp	0.53	0.47	0.46	0.29		0.82	0.64	0.64	0.39		0.53	0.45	0.45	0.28	
Average	**5.26**	**3.66**	**3.51**	**1.82**	**5.17**	**8.93**	**6.85**	**6.62**	**6.14**	**8.80**	**5.94**	**3.70**	**3.36**	**4.38**	**3.76**

Table 10: Computational costs of different methods for NESTA instances.

	Typical Operating Conditions				Congested Operating Conditions				Small Angle Conditions			
case	SOCP	SOCPA	S34A	SSDP	SOCP	SOCPA	S34A	SSDP	SOCP	SOCPA	S34A	SSDP
3lmbd	0.02	0.13	0.31	0.21	0.03	0.09	0.14	0.11	0.02	0.14	0.32	0.13
4gs	0.04	0.11	0.10	0.08	0.02	0.09	0.11	0.10	0.02	0.09	0.14	0.14
5pjm	0.02	0.13	0.15	0.21	0.02	0.14	0.16	0.23	0.02	0.12	0.13	0.21
6ww	0.02	0.22	0.24	0.39	0.02	0.22	0.24	0.51	0.02	0.21	0.25	0.39
9wscc	0.02	0.17	0.18	0.11	0.02	0.18	0.19	0.13	0.02	0.17	0.19	0.17
14ieee	0.02	0.45	0.48	0.67	0.02	0.42	0.46	0.66	0.02	0.40	0.44	0.65
29edin	0.20	3.89	4.86	2.44	0.14	3.61	4.53	2.50	0.14	3.42	5.01	2.56
30as	0.04	1.00	1.08	1.09	0.04	0.93	1.16	1.14	0.04	0.99	1.14	1.15
30fsr	0.03	0.99	1.06	1.11	0.04	0.91	1.05	1.15	0.04	0.88	1.16	1.14
30ieee	0.04	0.88	1.03	1.08	0.04	0.91	1.00	1.14	0.04	0.89	0.97	1.14
39epri	0.04	0.89	0.97	0.64	0.04	0.95	0.96	0.86	0.04	0.90	0.98	0.82
57ieee	0.08	2.04	2.09	2.17	0.07	1.71	1.94	2.33	0.07	1.84	2.06	2.10
118ieee	0.25	4.98	5.57	5.97	0.19	4.75	5.84	5.79	0.33	4.86	5.92	6.34
162ieee	0.25	8.97	12.14	11.37	0.23	9.30	13.50	11.93	0.34	9.04	13.65	11.53
189edin	0.40	4.84	5.79	2.89	0.35	4.89	5.38	3.21	0.43	5.35	6.34	3.31
300ieee	0.47	9.93	14.96	11.58	0.44	10.54	14.26	11.11	0.55	9.74	13.11	11.76
1460wp	2.96	63.26	83.04	45.46	3.88	74.73	117.83	52.09	3.64	78.12	114.10	42.69
2224edin	5.99	118.38	167.43	95.95	8.72	130.32	234.88	112.95	6.65	110.46	159.24	96.03
2383wp	7.59	120.68	165.48	117.79	8.39	132.29	296.21	127.18	7.08	117.14	205.04	108.82
2736sp	5.67	119.35	221.79	130.44	10.28	151.94	294.40	191.63	5.11	112.51	150.25	121.21
2737sop	4.61	114.62	149.41	95.37	9.59	148.50	218.83	142.93	3.90	108.21	143.75	85.97
2746wop	5.22	118.63	215.98	116.85	6.48	130.92	260.75	142.77	4.49	113.43	221.35	109.04
2746wp	6.84	123.01	203.85	117.24	7.42	136.72	229.48	137.48	5.65	116.49	209.42	127.23
3012wp	7.12	142.06	242.37	123.21	13.30	184.93	373.88	179.50	7.97	162.75	370.68	132.40
3120sp	7.72	147.32	255.43	125.04	11.34	164.37	310.54	150.02	9.38	177.76	337.48	137.24
3375wp	9.15	184.14	306.22	169.13	18.31	239.98	600.27	247.87	10.66	208.05	436.30	194.67

Table 11: Percentage optimality gap of three instances from NESTA with small angle difference conditions.

case	S34	SOCPA	SDP
5pjm	0.40	0.45	0.00
3lmbd	1.53	2.33	2.06
29edin	31.39	25.94	28.44

3.4.4 Incomparability of the Proposed Methods with SDP Relaxation

We now prove the incomparability of our proposed methods with the SDP relaxation as stated in Propositions 9 and 12 using three specific instances from the NESTA archive. To start with, let S34 denote a variant of S34A without the arctangent envelopes. The percentage optimality gaps of these instances are presented in Table 11. First of all, note that the SDP relaxation is not dominated by S34 or SOCPA as the 5-bus case 5pjm demonstrates. And SDP does not dominate S34 due to case 3lmbd, which proves Proposition 9; also SDP does not dominate SOCPA due to case 29edin, which proves Proposition 12. These instances also show that S34 and SOCPA are incomparable.

3.4.5 Robustness of the Proposed Methods

We test the robustness of our methods by solving perturbed instances to the standard IEEE benchmarks. In particular, load values are randomly perturbed 5% to obtain 10 similar and realistic instances. Results in Table 12 show that our methods consistently provide provably good solutions and tight relaxations for OPF problem.

We note that we have not reported 3 instances, namely cases 30, 30Q and 2383wp, in Table 12. In fact, for case2383wp, all the instances are proven to be infeasible by SOCP. On the other hand, for cases 30 and 30Q, we are able to find feasible solutions for only one instance. For the remaining nine instances, six of them are proven to be infeasible by SOCPA, S34A, and SSDP but not by SOCP. We also note that eight

Table 12: Average percentage optimality gaps of perturbed IEEE standard benchmarks.

	SOCP		SOCPA		S34A		SSDP	
case	%gap	time	%gap	time	%gap	time	%gap	time
6ww	0.62	0.06	0.02	0.26	0.01	0.32	0.00	0.46
9	0.00	0.04	0.00	0.19	0.00	0.20	0.00	0.11
9Q	0.09	0.04	0.09	0.19	0.09	0.20	0.09	0.12
14	0.08	0.05	0.08	0.38	0.06	0.41	0.00	0.61
ieee30	0.04	0.06	0.04	0.78	0.04	0.81	0.00	1.03
39	0.03	0.09	0.01	0.91	0.01	0.99	0.00	0.82
57	0.07	0.11	0.07	1.45	0.07	1.51	0.00	1.93
118	0.25	0.30	0.25	3.64	0.17	5.12	0.04	5.04
300	0.63	0.66	0.60	7.90	0.58	13.71	0.33	10.05
2736sp	0.30	12.67	0.23	110.42	0.23	201.81	0.05	120.92
2737sop	0.26	11.98	0.22	108.89	0.22	188.52	0.03	92.18
2746wop	0.38	9.46	0.30	114.89	0.28	215.16	0.06	115.31
2746wp	0.32	12.43	0.25	125.95	0.25	217.88	0.05	117.24
3012wp	0.81	16.77	0.71	125.77	0.71	167.80	0.43	116.88
3120sp	0.53	16.19	0.44	134.98	0.44	180.07	0.25	115.47
3375wp	0.26	19.30	0.24	179.34	0.23	481.92	0.19	161.67
Average	**0.29**	**6.26**	**0.22**	**57.25**	**0.21**	**104.78**	**0.10**	**53.74**

instances are proven to be infeasible by SDP.

3.5 Conclusions

In this chapter, we study the OPF problem, a fundamental optimization problem in electric power system analysis. We have the following main conclusions:

(i) The proposed strong SOCP relaxations offer a computationally attractive alternative to the SDP relaxations for practically solving large-scale OPF problems. The lower bounds obtained by the strong SOCP relaxations are extremely close to those of the SDP relaxations, and are not always dominated by the SDP relaxations, but within a computation time that is an order of magnitude faster than the latter. In case tight bounds on phase angles are known, the strong SOCP relaxation involving arctangent linearization together with McCormick

constraints from the cycle decompositions is recommended. In case explicit bounds on phase angles are not known, the third approach of SOCP with SDP separation is recommended.

(ii) The proposed SOCP relaxation produces a solution that can be conveniently used as a good warm start for a local solver, such as IPOPT. In comparison, recovering a feasible solution from SDP relaxations is a computationally challenging task, even when the SDP relaxation is tight.

(iii) The proposed SOCP relaxations provide stronger bounds than existing quadratic relaxation approaches [28, 27] on most instances.

CHAPTER IV

MINOR RELAXATION AND SOCP BASED SPATIAL BRANCH-AND-CUT METHOD FOR THE OPTIMAL POWER FLOW PROBLEM

4.1 Introduction

In this chapter, we aim to solve the OPF problem using global optimization methods. This chapter builds on Chapter 3 and proposes several enhancements to the ideas originated in that chapter with the addition of new ones which helps us to obtain significantly better results. Throughout the chapter, we compare and contrast the common ideas presented here with Chapter 3. Our analysis starts with a reformulation of the rectangular formulation using a rank-constrained SDP problem with a complex matrix variable. In this reformulation, there are two difficult constraints: the positive semidefiniteness of the matrix variable and the rank-one condition. Although the former is a convex constraint, solving SDP problems involving large matrices remains a challenge. Therefore, we seek alternative approaches that would avoid including a positive semidefiniteness restriction on the original matrix variable. Instead, we consider principal submatrices, which may correspond to any subgraph (in particular, cycles) of the original power network. This clearly relaxes the positive semidefiniteness constraint on a large matrix into constraints involving smaller matrices. Finally, we outer-approximate these small positive semidefiniteness conditions via hyperplanes in a cutting-plane framework.

The rank-one restriction, the second difficult constraint, is nonconvex and, in general, difficult to deal with. We first write it as an equivalent set of 2×2 minor conditions. It turns out that there are three types of such minors and we propose

outer-approximation methods for each of them, using either SOCP or LP relaxations. We again use these relaxations as a basis to generate cutting planes. We also propose an alternative characterization of the minor conditions using a constraint involving the arctangent function, which we then convexify via linear envelopes. Our construction heavily depends on the availability of tight variable bounds obtained by solving small-size SOCP problems.

We use these several different kinds of relaxations in a spatial branch-and-cut framework. Our algorithm is based on computationally attractive SOCP relaxations, and designed to spend a non-trivial amount of time at the root node relaxation to improve the dual bound on the problem. Many of the difficult test cases are solved at the root node relaxation, and for the others we utilize spatial branching. Extensive computational experiments show that our approach scales reasonably well and provides strong dual guarantees, often significantly better than the ones obtained from the SDP relaxation.

The rest of the chapter is organized as follows: In Section 4.2, we give an alternative formulation of the OPF based on an SDP reformulation with a rank, or equivalently, minor constraint. We propose several outer-approximation schemes to incorporate the minor conditions in a convex relaxation of the OPF problem. Section 4.3 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 [25] are summarized in Section 4.4. Finally, Section 4.5 concludes the chapter with some further remarks and possible future research directions.

4.2 Rank Constrained OPF Problem and Minor Relaxations

4.2.1 Formulation

The OPF problem given in (7) is a polynomial optimization problem, and it can be posed as a quadratically constrained quadratic program (QCQP). For nonconvex QCQPs, a general strategy is *lifting*. This can be accomplished by defining a matrix variable which will replace the quadratic terms in the original variables with linear ones in the lifted matrix variable. In this reformulation, the only nonconvexity will be a rank-one requirement on the matrix variable. We formally obtain this lifted formulation for the OPF problem as

$$z_{\text{rank}} = \min \left\{ \sum_{i \in \mathcal{G}} C_i(p_i^g) : (29), (77), (1\text{h}) - (1\text{i}), \text{rank}(X) = 1 \right\}, \tag{76}$$

where the apparent flow limit constraint can be stated as follows:

$$[-G_{ij}X_{ii} + G_{ij}\Re(X_{ij}) + B_{ij}\Im(X_{ij})]^2 + [B_{ij}X_{ii} - B_{ij}\Re(X_{ij}) + G_{ij}\Im(X_{ij})]^2 \leq (S_{ij}^{\max})^2, \quad (i,j) \in \mathcal{L}. \tag{77}$$

Let z_{rect} denote the optimal value of the OPF problem (7). Then, clearly $z_{\text{rect}} = z_{\text{rank}}$. The sole nonconvexity in the rank-constrained OPF formulation (76) is captured by the rank constraint $\text{rank}(X) = 1$, which is difficult to analyze in general.

Definition 1. *Let X be an $n \times n$ complex matrix. Then, a 2×2 minor of the matrix X is the determinant of a 2×2 submatrix of X obtained by deleting $n - 2$ rows and columns.*

Another characterization of the rank constraint is due to Proposition 13.

Proposition 13. *For a nonzero, hermitian matrix X, $X \succeq 0$ and rank$(X) = 1$ iff all the 2×2 minors are zero and the diagonal elements are nonnegative.*

Proof. ($\Rightarrow$) Let $X \succeq 0$ and $\text{rank}(X) = 1$. 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 $X_{ik} = x_i \bar{x}_k$, $X_{il} = x_i \bar{x}_l$, $X_{jk} = x_j \bar{x}_k$, and $X_{jl} = x_j \bar{x}_l$. 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 $X_{ii} = x_i \bar{x}_i = |x_i|^2 \geq 0$. Therefore, we conclude that the diagonal elements of X are nonnegative.

($\Leftarrow$) Recall that $\text{rank}(X)$ is equal to the size of the largest invertible submatrix of X. Since all the 2×2 minors of X are zero, none of the 2×2 submatrices of X are invertible. This implies that $\text{rank}(X) = 1$ 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 $X \succeq 0$. □

Then, we can further reformulate the OPF problem as

$$z_{\text{minor}} = \min \Bigg\{ \sum_{i \in \mathcal{G}} C_i(p_i^g) : (29a) - (29e), (77), (1h) - (1i), \\ \text{all } 2 \times 2 \text{ minors of } X \text{ are zero} \Bigg\}, \tag{78}$$

where we again have $z_{\text{rect}} = z_{\text{rank}} = z_{\text{minor}}$.

Now, let us discuss some possible relaxations. A straightforward SDP relaxation for the OPF problem can be obtained by relaxing the rank, or equivalently, minor constraints as

$$z_{\text{SDP}} = \min \Bigg\{ \sum_{i \in \mathcal{G}} C_i(p_i^g) : (29a) - (29e), (77), (1h) - (1i) \Bigg\}. \tag{79}$$

By construction, $z_{\text{SDP}} \leq z_{\text{rect}}$. In practice, this SDP relaxation has been shown to be quite strong in terms of the dual bound it provides for standard IEEE test cases [67, 76]. However, for challenging test cases such as the instances from NESTA archive [25], the duality gap can be quite large [27]. Another challenge is due to the computational difficulty of solving large-scale SDP problems. Although sparsity

exploitation methods help reduce the solution times significantly, one may prefer to use computationally less demanding methods to obtain dual bounds.

A simple example of a computationally attractive relaxation can be obtained by relaxing further the positive definiteness requirement of the matrix X. In this case, we obtain a flow relaxation as

$$z_{\text{flow}} = \min \left\{ \sum_{i \in \mathcal{G}} C_i(p_i^g) : (29a) - (29d), (77), (1h) - (1i) \right\}. \tag{80}$$

This relaxation is clearly weaker than the SDP relaxation, hence we have $z_{\text{flow}} \leq z_{\text{SDP}} \leq z_{\text{rect}}$. On the other hand, flow relaxation is much easier than solving the full SDP relaxation.

Our approach will be based on solving an improved version of the flow relaxation by incorporating convex relaxations of the minor restrictions in the form of linear or second-order cone representable constraints. Some of the additional constraints will be a relaxed version of the 2×2 minor restrictions (Sections 4.2.2 and 4.2.3) and some of them will be a weaker version of the semidefiniteness requirement on X (Sections 4.2.2 and 4.2.4). In this way, our computational cost will be between the SDP relaxation and the flow relaxation. Also, by construction, our approach can be expected to give better bounds than the flow relaxation. Numerical experiments demonstrate that the proposed approach is incomparable to, and often provides much better dual bounds than, the SDP relaxation, especially for challenging NESTA instances.

4.2.2 Analysis of 2×2 Minors

In this section, we analyze the 2×2 minors of the matrix X in detail, which are one of the three different types as listed below. 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

$$\begin{vmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{vmatrix} = 0, \tag{81}$$

which is equivalent to

$$0 = X_{ii}X_{jj} - X_{ij}X_{ji} = c_{ii}c_{jj} - (c_{ij} - \mathrm{i}s_{ij})(c_{ij} + \mathrm{i}s_{ij}) = c_{ii}c_{jj} - (c_{ij}^2 + s_{ij}^2). \tag{82}$$

Note that this relation defines the boundary of the rotated 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

$$\begin{vmatrix} X_{ii} & X_{ij} \\ X_{ki} & X_{kj} \end{vmatrix} = 0, \tag{83}$$

which is equivalent to

$$\begin{aligned} 0 =& X_{ii}X_{kj} - X_{ij}X_{ki} = c_{ii}(c_{kj} - \mathrm{i}s_{kj}) - (c_{ij} - \mathrm{i}s_{ij})(c_{ki} - \mathrm{i}s_{ki}) \\ =& (c_{ii}c_{kj} - c_{ij}c_{ki} + s_{ij}s_{ki}) - \mathrm{i}(c_{ii}s_{kj} - s_{ij}c_{ki} - c_{ij}s_{ki}). \end{aligned} \tag{84}$$

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

$$\begin{vmatrix} X_{ij} & X_{ik} \\ X_{lj} & X_{lk} \end{vmatrix} = 0, \tag{85}$$

which is equivalent to

$$\begin{aligned} 0 =& X_{ij}X_{lk} - X_{ij}X_{lj} = (c_{ij} - \mathrm{i}s_{ij})(c_{lk} - \mathrm{i}s_{lk}) - (c_{ij} - \mathrm{i}s_{ij})(c_{lj} - \mathrm{i}s_{lj}) \\ =& (c_{ij}c_{lk} - s_{ij}s_{lk} - c_{lj}c_{ik} + s_{lj}s_{ik}) - \mathrm{i}(s_{ij}c_{lk} - c_{ij}s_{lk} - s_{lj}c_{ik} - c_{lj}s_{ik}). \end{aligned} \tag{86}$$

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

We analyze these minors in detail below. At this point, we would like to contrast our approach with the existing literature. In [22, 23, 44], 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.

4.2.2.1 Type 1: Edge Minors

Let i and j be distinct elements of the set $\mathcal{B}$ such that $(i, j) \in \mathcal{L}$. Note that since this minor is also principal, guaranteeing its nonnegativity also relaxes the positive semidefinitess restriction. Therefore, an accurate relaxation of Type 1 minors will help two-fold.

Suppose that we are interested in the convex hull of the set defined by a Type 1 minor in (81) over the box $\mathcal{D}_{ij} := [\underline{c}_{ii}, \overline{c}_{ii}] \times [\underline{c}_{jj}, \overline{c}_{jj}] \times [\underline{c}_{ij}, \overline{c}_{ij}] \times [\underline{s}_{ij}, \overline{s}_{ij}]$. More precisely, let us define the following nonconvex set

$$\mathcal{K}_{ij}^{=} := \left\{ (c_{ii}, c_{jj}, c_{ij}, s_{ij}) \in \mathcal{D}_{ij} : c_{ij}^2 + s_{ij}^2 = c_{ii} c_{jj} \right\}. \tag{87}$$

It follows from [96] that

$$\text{conv}(\mathcal{K}_{ij}^{=}) = \mathcal{K}_{ij}^{\leq} \cap \text{conv}(\mathcal{K}_{ij}^{\geq}), \tag{88}$$

where

$$\mathcal{K}_{ij}^{\leq} := \left\{ (c_{ii}, c_{jj}, c_{ij}, s_{ij}) \in \mathcal{D}_{ij} : c_{ij}^2 + s_{ij}^2 \leq c_{ii} c_{jj} \right\}, \tag{89}$$

and

$$\mathcal{K}_{ij}^{\geq} := \left\{ (c_{ii}, c_{jj}, c_{ij}, s_{ij}) \in \mathcal{D}_{ij} : c_{ij}^2 + s_{ij}^2 \geq c_{ii} c_{jj} \right\}. \tag{90}$$

First of all, $\mathcal{K}_{ij}^{\leq}$ 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 $\text{conv}(\mathcal{K}_{ij}^{=})$, it suffices to find $\text{conv}(\mathcal{K}_{ij}^{\geq})$, which is the intersection of a polytope and the complement of a convex set. Here, we use the following result:

Theorem 3. *(Theorem 1 in [46]) 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 is also constructive: Let E_l, $l = 1, \ldots, L$ be one-dimensional faces of the polytope P. Then, we have

$$\text{conv}(P \cap G) = \text{conv}\left(\bigcup_{l=1}^{L} \text{conv}(E_l \cap G)\right). \tag{91}$$

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

We can easily apply Theorem 3 to our case by choosing $P = \mathcal{D}_{ij}$ and $G = \left\{(c_{ii}, c_{jj}, c_{ij}, s_{ij}) \in \mathbb{R}^4 : c_{ij}^2 + s_{ij}^2 \geq c_{ii}c_{jj}\right\}$ since one-dimensional faces of the box $\mathcal{D}_{ij}$ are trivial to obtain (fix three of the variables to one of their bounds). The exact procedures to obtain extreme points are given in Algorithms 3-4.

Algorithm 2 Extreme points 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.

$$E = \begin{cases} \{\underline{s}_{ij}, \overline{s}_{ij}\} & \text{if } \phi < 0 \\ \{\underline{s}_{ij}, \overline{s}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \overline{s}_{ij} < -\sqrt{\phi} \\ \{\underline{s}_{ij}, -\sqrt{\phi}\} & \text{if } \phi \geq 0 \text{ and } \underline{s}_{ij} \leq -\sqrt{\phi} \leq \overline{s}_{ij} \\ \{\underline{s}_{ij}, -\sqrt{\phi}, \sqrt{\phi}, \overline{s}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \underline{s}_{ij} \leq -\sqrt{\phi}, \sqrt{\phi} \geq \overline{s}_{ij} \\ \emptyset & \text{if } \phi \geq 0 \text{ and } -\sqrt{\phi} \leq \underline{s}_{ij}, \overline{s}_{ij} \leq \sqrt{\phi} \\ \{\sqrt{\phi}, \overline{s}_{ij}\} & \text{if } \phi \geq 0 \text{ and } -\sqrt{\phi} \leq \underline{s}_{ij} \leq \sqrt{\phi} \leq \overline{s}_{ij} \\ \{\underline{s}_{ij}, \overline{s}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \underline{s}_{ij} > \sqrt{\phi} \end{cases}$$

Algorithm 3 Extreme points 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.

$$E = \begin{cases} \{\underline{c}_{ij}, \overline{c}_{ij}\} & \text{if } \phi < 0 \\ \{\underline{c}_{ij}, \overline{c}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \sqrt{\phi} < \underline{c}_{ij} \\ \{\sqrt{\theta}, \overline{c}_{ij}\} & \text{if } \phi \geq 0 \text{ and } \underline{c}_{ij} \leq \sqrt{\phi} \leq \overline{c}_{ij} \\ \emptyset & \text{if } \phi \geq 0 \text{ and } \sqrt{\phi} > \overline{c}_{ij} \end{cases}$$

Algorithm 4 Extreme points 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.

$$E = \begin{cases} \emptyset & \text{if } \phi < \underline{c}_{ii} \\ \{\theta, \overline{c}_{ii}\} & \text{if } \underline{c}_{ii} \leq \phi \leq \overline{c}_{ii} \\ \{\underline{c}_{ii}, \overline{c}_{ii}\} & \text{if } \theta > \overline{c}_{ii} \end{cases}$$

So far, we have computed the extreme points of one-dimensional faces, say z_{ij}^k, $k = 1, \ldots, K$, where $z_{ij} = (c_{ii}, c_{jj}, c_{ij}, s_{ij})$. Now, we give the convex hull description of $\mathcal{K}_{ij}^{\geq}$ as follows:

$$\text{conv}(\mathcal{K}_{ij}^{\geq}) = \text{conv}(\{z_{ij}^k\}_{k=1}^K) = \{z_{ij} : \exists \lambda : z_{ij} = \sum_{k=1}^K \lambda_k z_{ij}^k, \sum_{k=1}^K \lambda_k = 1, \lambda \geq 0\}. \quad (92)$$

Finally, the above discussion proves the following proposition:

Proposition 14. *Let z_{ij}^k be computed using Algorithms 2-4. Then,*

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

An Outer Approximation to conv($\mathcal{K}_{ij}^{=}$): Since Proposition 14 describes the convex hull of conv($\mathcal{K}_{ij}^{=}$) in an extended space, it may have several facets in the original space of the variables. Instead, we propose an outer approximation to conv($\mathcal{K}_{ij}^{=}$) in the space of (c, s) variables using only four linear inequalities. Our preliminary experiments showed that this approximation is quite accurate and computationally cheap.

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

$$f(c_{ij}, s_{ij}) := \sqrt{c_{ij}^2 + s_{ij}^2} \geq \sqrt{c_{ii}c_{jj}} =: g(c_{ii}, c_{jj}). \tag{93}$$

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. We have the following propositions which formalize this idea:

Proposition 15. *The affine functions $\nu_{ij}^m + \eta_{ij}^m c_{ii} + \delta_{ij}^m c_{jj}$, $m = 1, 2$, 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 16. *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 $\nu_{ij}^n + \eta_{ij}^n c_{ij} + \delta_{ij}^l s_{ij}$, $n = 3, 4$, 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 17. *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 $\nu_{ij}^n + \eta_{ij}^n c_{ij} + \delta_{ij}^n s_{ij}$, $n = 3, 4$, 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 18. *Let ν_{ij}^m, η_{ij}^m and δ_{ij}^m, $m = 1,\ldots,4$ be calculated using Propositions 15-17. Then, the edge cuts (EC) defined as*

$$EC_{ij}(\underline{c},\overline{c},\underline{s},\overline{s}):\ \nu_{ij}^n + \eta_{ij}^n c_{ij} + \delta_{ij}^n s_{ij} \geq \nu_{ij}^m + \eta_{ij}^m c_{ii} + \delta_{ij}^m c_{jj},\ m = 1,2 \text{ and } n = 3,4 \quad (94)$$

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

A different analysis related to Type 1 minor condition is carried out in [23] by considering the following set:

$$\begin{aligned}\mathcal{K}'_{ij} = \{(c_{ii},c_{jj},c_{ij},s_{ij}) : c_{ij}^2 + s_{ij}^2 = c_{ii}c_{jj}, (c_{ii},c_{jj}) \in [\underline{c}_{ii},\overline{c}_{ii}]\times[\underline{c}_{jj},\overline{c}_{jj}],\\ c_{ij}\tan\underline{\theta}_{ij} \leq s_{ij} \leq c_{ij}\tan\overline{\theta}_{ij}\}.\end{aligned} \quad (95)$$

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.

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

Real and imaginary parts of Type 2 and Type 3 minors in (83) and (85) 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:

$$\mathcal{S}_a = \{(x,y) \in \mathbb{R}^N \times \mathbb{R}^N : \sum_{i=1}^N a_i x_i y_i = 0,\ \underline{x} \leq x \leq \overline{x},\ \underline{y} \leq y \leq \overline{y}\}. \quad (96)$$

We verify the following theorem, whose proof can be seen in Appendix A:

Theorem 4. *conv($\mathcal{S}_a$) is second-order cone representable.*

An Outer Approximation to conv(S_a) In the proof of Theorem 4, we construct the convex hull of conv(S_a) in an extended space with exponentially many disjunctions in N, where some of the descriptions may contain second-order cones. Therefore, including conv(S_a) as part of the relaxation might be quite costly. Instead, we propose a linear outer-approximation to conv(S_a) using McCormick envelopes [77] as follows:

$$\mathcal{S}_a^M = \{(x,y) \in \mathbb{R}^N \times \mathbb{R}^N : \exists w \in \mathbb{R}^N : \sum_{i=1}^{N} a_i w_i = 0,$$
$$\max\{\underline{y}x + \underline{x}y - \underline{x}\underline{y}, \overline{y}x + \overline{x}y - \overline{x}\overline{y}\} \leq w \leq \min\{\underline{y}x + \overline{x}y - \overline{x}\underline{y}, \overline{y}x + \underline{x}y - \underline{x}\overline{y}\}\}. \tag{97}$$

Our preliminary experiments have shown that $\mathcal{S}_a^M$ tightly approximates conv$(\mathcal{S}_a)$.

A Cycle Based Relaxation Although we have a linear outer-approximation to 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 Chapter 3 although the construction in that chapter 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 and X^C denote the principal submatrix 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 the submatrix X^C and define the following set

$$\mathcal{Q}_C = \{(c,s) : \exists(\tilde{c},\tilde{s}) : q_k(c,s,\tilde{c},\tilde{s}) = 0,\ k \in \mathcal{K}_C\}, \tag{98}$$

where q_k's are bilinear equations corresponding to minor restrictions from a collection $\mathcal{K}_C$. Here, we denote the variables which correspond to an i,j pair such that $(i,j) \in \mathcal{L}$ as c_{ij}, s_{ij}, and $\tilde{c}_{ij}, \tilde{s}_{ij}$ otherwise. In the latter case, such "artificial" tilde variables

appear for extra edges added to the cycle C after triangulation. See Figure 7 for an illustration on how a 7-cycle can be triangulated.

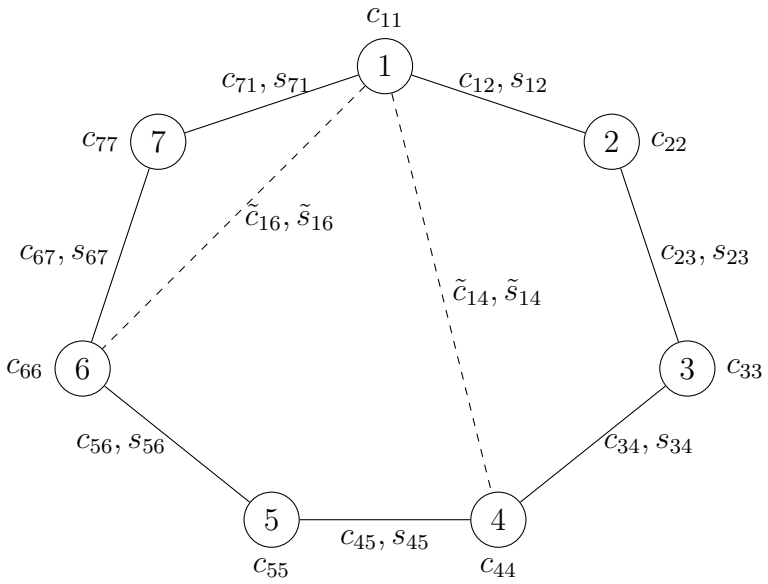


Figure 7: Triangulation of a 7-cycle. Here, we consider three minors corresponding to the "subcycles" $\{1, 2, 3, 4\}$, $\{1, 4, 5, 6\}$ and $\{1, 6, 7\}$. Variables corresponding to each node and edge are also shown.

Finally, we write the McCormick relaxation of 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:

$$\mathcal{M}_C(\underline{c}, \overline{c}, \underline{s}, \overline{s}) = \left\{ (c, s) : \exists (\tilde{c}, \tilde{s}, w) : A \begin{bmatrix} c \\ s \end{bmatrix} + \tilde{A} \begin{bmatrix} \tilde{c} \\ \tilde{s} \end{bmatrix} + Bw \leq c,\ Ew = 0 \right\}. \quad (99)$$

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.

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 [78]).

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:

$$\mathcal{M}_C^D := \text{conv}\left(\bigcup_{d \in D} \mathcal{M}_C(\underline{c}_d, \overline{c}_d, \underline{s}_d, \overline{s}_d)\right) \subseteq \mathcal{M}_C(\underline{c}, \overline{c}, \underline{s}, \overline{s}) \quad \text{and} \quad \text{conv}(\mathcal{Q}_D) \subseteq \mathcal{M}_C^D. \tag{100}$$

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 Appendix C.2 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 7, variables $c_{23}, s_{23}, c_{34}, s_{34}, c_{45}, s_{45}, c_{56}, s_{56}, c_{67}, s_{67}$ are bisected.

4.2.3 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 part of the complex matrix variable X:

$$\theta_j - \theta_i = \text{atan2}(s_{ij}, c_{ij}) \quad i, j \in \mathcal{B} \tag{101}$$

4.2.3.1 Equivalence

First, we claim that Type 1 minor constraint (81) together with arctangent constraint (101) implies Type 2 minor (83) equations.

Proposition 19.

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

Proof. From (101), 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,

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

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 (83). □

A similar proposition about Type 3 minors is also true, that is, Type 1 minor constraint (81) together with arctangent constraint (101) implies Type 3 minor (85) equations.

Proposition 20.

$$\left\{(c,s): \begin{vmatrix} X_{ij} & X_{ik} \\ X_{lj} & X_{lk} \end{vmatrix} = 0\right\} \supseteq \left\{(c,s): \exists \theta: \begin{vmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{vmatrix} = \begin{vmatrix} X_{ii} & X_{ik} \\ X_{ki} & X_{kk} \end{vmatrix} = \begin{vmatrix} X_{jj} & X_{jl} \\ X_{lj} & X_{ll} \end{vmatrix} = 0,$$

$$\begin{vmatrix} X_{kk} & X_{kl} \\ X_{lk} & X_{ll} \end{vmatrix} = 0, \theta_j - \theta_i = \text{atan2}(s_{ij}, c_{ij}), \theta_k - \theta_i = \text{atan2}(s_{ik}, c_{ik}),$$

$$\theta_j - \theta_l = \text{atan2}(s_{lj}, c_{lj}), \theta_l - \theta_l = \text{atan2}(s_{lk}, c_{lk})\Big\}.$$

We omitted the proof of Proposition 20 due to its similarity to the proof of Proposition 19.

4.2.3.2 Convexification

In Section 4.2.2.2, 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 (101). To start with, let us define the following nonconvex set

$$\mathcal{AT}' := \left\{(c,s,\theta) \in \mathbb{R}^3 : \theta = \arctan\left(\frac{s}{c}\right), (c,s,\theta) \in [\underline{c},\overline{c}] \times [\underline{s},\overline{s}] \times [\underline{\theta},\overline{\theta}]\right\}. \tag{102}$$

Note that $\mathcal{AT}'$ is an improved version of $\mathcal{AT}$ defined in (60) with the addition of angle difference bounds.

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

Proposition 21. *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 $\{\zeta^1, \zeta^3, \zeta^4\}$ defined in* (61)*, respectively. Then, two valid inequalities for $\mathcal{AT}'$ can be obtained as*

$$\bar{\gamma}_m + \alpha_m c + \beta_m s \geq \arctan\left(\frac{s}{c}\right) \tag{103}$$

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

$$\Delta\gamma_m = \max\Big\{ \arctan\left(\frac{s}{c}\right) - (\gamma_m + \alpha_m c + \beta_m s) : (c,s) \in [\underline{c},\overline{c}] \times [\underline{s},\overline{s}], \\ c\tan\underline{\theta} \leq s \leq c\tan\overline{\theta}\Big\}, \tag{104}$$

for $m = 1, 2$.

Note that by the construction of (104), 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 (104) can be solved by enumerating all possible Karush-Kuhn-Tucker (KKT) points. These inequalities are improvements over the similar ones in Section 3.3.2 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 22. *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*

$$\bar{\gamma}_n + \alpha_n c + \beta_n s \leq \arctan\left(\frac{s}{c}\right) \tag{105}$$

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

$$\begin{aligned} \Delta\gamma_n = \max\Big\{(\gamma_n + \alpha_n c + \beta_n s) - \arctan\left(\frac{s}{c}\right) : (c, s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}], \\ c\tan\underline{\theta} \leq s \leq c\tan\overline{\theta}\Big\}, \end{aligned} \tag{106}$$

for $n = 3, 4$.

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

$$\text{AT}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) : \ \bar{\gamma}^m_{ij} + \alpha^m_{ij} c_{ij} + \beta^m_{ij} s_{ij} \leq \theta_j - \theta_i \leq \bar{\gamma}^n_{ij} + \alpha^n_{ij} c_{ij} + \beta^n_{ij} s_{ij}, \ m = 1, 2 \ ; \ n = 3, 4 \tag{107}$$

for some line (i, j).

4.2.4 General Principal Submatrices

Up until this point, we have mainly discussed how to incorporate the minor restrictions into our relaxation scheme. In this section, we propose an approach to include

a relaxed version of the positive semidefiniteness restriction 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 (83) and (85).

Let $\mathcal{B}' \subseteq \mathcal{B}$. Let $x \in \mathbb{R}^{2|\mathcal{B}'|}$ be a vector of bus voltages defined as $x = [e; f]$ 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,

$$c_{ij} = e_i e_j + f_i f_j = W_{ij} + W_{i'j'} \qquad i, j \in \mathcal{B}' \tag{108a}$$

$$s_{ij} = e_i f_j - e_j f_i = W_{ij'} - W_{ji'} \qquad i, j \in \mathcal{B}' \tag{108b}$$

$$c_{ii} = e_i^2 + f_i^2 = W_{ii} + W_{i'i'}, \qquad i \in \mathcal{B}' \tag{108c}$$

$$W = xx^T. \tag{108d}$$

Here, we used real matrices instead of complex matrices for convenience. It is proven in [97] that using real matrices is equivalent to complex matrices.

Clearly, the set defined by (108) is nonconvex. A straightforward SDP relaxation can be presented as follows:

$$\begin{aligned} \mathcal{S}_{\mathcal{B}'} := \{(c,s) \in \mathbb{R}^{2|\mathcal{B}'|} : \exists W \in \mathbb{S}^{2|\mathcal{B}'|} : c_{ij} &= W_{ij} + W_{i'j'} \quad i, j \in \mathcal{B}' \\ s_{ij} &= W_{ij'} - W_{ji'} \quad i, j \in \mathcal{B}' \\ c_{ii} &= W_{ii} + W_{i'i'} \quad i \in \mathcal{B}' \ \}. \end{aligned} \tag{109}$$

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 [60] 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

$$\begin{aligned}\mathcal{S}_C := \{(c,s) \in \mathbb{R}^{2|C|} : \exists W \in \mathbb{S}^{2|C|} : & c_{ij} = W_{ij} + W_{i'j'} \quad (i,j) \in C \\ & s_{ij} = W_{ij'} - W_{ji'} \quad (i,j) \in C \\ & c_{ii} = W_{ii} + W_{i'i'} \quad i \in \{k : (k,l) \in C\} \ \},\end{aligned} \tag{110}$$

and then, use the procedure in Appendix C.2 to obtain cutting planes for this set.

4.2.5 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.

4.2.5.1 Optimization-Based Bound Tightening

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

$$\underline{\theta}_{ij} \leq \theta_i - \theta_j \leq \overline{\theta}_{ij}, \tag{111}$$

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

$$\underline{c}_{ij} = \underline{V}_i \underline{V}_j \cos(\underline{\theta}_{ij}), \qquad \overline{c}_{ij} = \overline{V}_i \overline{V}_j, \tag{112a}$$

$$\underline{s}_{ij} = \overline{V}_i \overline{V}_j \sin(\underline{\theta}_{ij}), \qquad \overline{s}_{ij} = \overline{V}_i \overline{V}_j \sin(\overline{\theta}_{ij}). \tag{112b}$$

We claim that these bounds can be further tightened by solving an improved version of the SOCP bounding problems in Section 3.3.4.

Consider the following second-order cone representable set,

$$\mathrm{EC}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{113a}$$

$$\mathrm{AT}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{113b}$$

$$\underline{\theta}_{ij} \leq \theta_i - \theta_j \leq \overline{\theta}_{ij} \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{113c}$$

$$\underline{c}_{ij} \leq c_{ij} \leq \overline{c}_{ij}, \quad \underline{s}_{ij} \leq s_{ij} \leq \overline{s}_{ij} \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{113d}$$

(69),

where EC and AT are defined as in (94) and (107), respectively. Note that this set is an improved version of the one proposed in Section 3.3.4, which does not contain edge cut inequalities (113a) or arctangent envelopes (113b).

Let us now define the following problems:

$$\begin{aligned}
&\underline{P}^c_{kl}(\underline{c}, \overline{c}, \underline{s}, \overline{s}, r): \ \underline{c}^*_{kl} = \min\{c_{kl} : (113), (69)\} \\
&\overline{P}^c_{kl}(\underline{c}, \overline{c}, \underline{s}, \overline{s}, r): \ \overline{c}^*_{kl} = \max\{c_{kl} : (113), (69)\} \\
&\underline{P}^s_{kl}(\underline{c}, \overline{c}, \underline{s}, \overline{s}, r): \ \underline{s}^*_{kl} = \min\{s_{kl} : (113), (69)\} \\
&\overline{P}^s_{kl}(\underline{c}, \overline{c}, \underline{s}, \overline{s}, r): \ \overline{s}^*_{kl} = \max\{s_{kl} : (113), (69)\}
\end{aligned} \tag{114}$$

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 variables to obtain some bounds for the variables defined for the artificial edges by adopting the procedure proposed in Section 3.3.4.

4.2.5.2 Dual-Based Bound Tightening

Let us suppose that the problems (114) 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. 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

$$\Pi^c_{kl} = \underline{c}^*_{kl} - \sum_{(i,j)\in\in\mathcal{L}_{kl}(r)} \left(\underline{c}_{ij}\underline{\pi}^c_{kl}(\underline{c}_{ij}) + \overline{c}_{ij}\underline{\pi}^c_{kl}(\overline{c}_{ij}) + \underline{s}_{ij}\underline{\pi}^c_{kl}(\underline{s}_{ij}) + \overline{s}_{ij}\underline{\pi}^c_{kl}(\overline{s}_{ij})\right). \tag{115}$$

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:

$$\underline{c}^*_{kl} = \max\left\{\underline{c}^*_{kl},\ \Pi^c_{kl} + \sum_{(i,j)\in\in\mathcal{L}_{kl}(r)} \left(\underline{c}^*_{ij}\underline{\pi}^c_{kl}(\underline{c}_{ij}) + \overline{c}^*_{ij}\underline{\pi}^c_{kl}(\overline{c}_{ij}) + \underline{s}^*_{ij}\underline{\pi}^c_{kl}(\underline{s}_{ij}) + \overline{s}^*_{ij}\underline{\pi}^c_{kl}(\overline{s}_{ij})\right)\right\}. \tag{116}$$

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.

4.3 SOCP Based Spatial Branch-and-Cut Method

In Section 4.2, we proposed several convexification techniques for the rank (or equivalently, minor) constrained OPF problem including envelopes and cutting planes. In this section, we will show how they can be used in a relaxation scheme. In Section 4.3.1, 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 4.3.2. Finally, Section 4.3.3 presents the implementation details of the spatial branch-and-cut algorithm.

4.3.1 Root Node Relaxation

Our computational experiments will be based on an SOCP relaxation of the OPF problem. Let 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 $\mathcal{H}$ obtained by solving separation problems. The full model is defined as follows:

$$\text{SOCP}(\underline{c}, \overline{c}, \underline{s}, \overline{s}, \mathcal{H}) :$$

$$\min \sum_{i \in \mathcal{G}} C_i(p_i^g) \tag{117a}$$

$$\text{s.t. } [-G_{ij}c_{ii} + G_{ij}c_{ij} - B_{ij}s_{ij}]^2 + [B_{ij}c_{ii} - B_{ij}c_{ij} - G_{ij}s_{ij}]^2 \leq (S_{ij}^{\max})^2 \quad (i,j) \in \mathcal{L} \tag{117b}$$

$$\text{EC}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) \quad (i,j) \in \mathcal{L} \tag{117c}$$

$$\text{AT}_{ij}(\underline{c}, \overline{c}, \underline{s}, \overline{s}) \quad (i,j) \in \mathcal{L} \tag{117d}$$

$$\underline{\theta}_{ij} \leq \theta_i - \theta_j \leq \overline{\theta}_{ij} \quad (i,j) \in \mathcal{L} \tag{117e}$$

$$\underline{c}_{ij} \leq c_{ij} \leq \overline{c}_{ij}, \quad \underline{s}_{ij} \leq s_{ij} \leq \overline{s}_{ij} \quad (i,j) \in \mathcal{L} \tag{117f}$$

$$\alpha_h^T \begin{bmatrix} c \\ s \end{bmatrix} \geq \beta_h \quad h \in \mathcal{H} \tag{117g}$$

$$(10b) - (10e), (1h) - (1i), (13).$$

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 5.

Algorithm 5 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**
  Enlarge $\mathcal{C}$.
  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

4.3.2 Spatial Branch-and-Cut Algorithm

Algorithm 5 is quite successful in proving strong dual bounds for many instances from the NESTA archive as the numerical experiments in Section 4.4.2.1 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 6.

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:

$$line_L = \max_{(i,j)\in\mathcal{L}} \left| \theta_j - \theta_i - \arctan\left(\frac{s_{ij}}{c_{ij}}\right) \right|. \tag{118}$$

Then, among c_{ij} and s_{ij}, we choose the variable whose smallest distance to the boundary is 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 [95].

(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.

4.3.3 Implementation

In Algorithm 6, 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.

Algorithm 6 Spatial branch-and-cut.

Let LB, UB, $\mathcal{C}$ and $\mathcal{H}$ be computed from Algorithm 5.
Set $list = \{root\}$.
while $|list| > 0$ **do**
  $LB = \min_{l \in list} LB_l$ and $L = \text{argmin}_{l \in list} LB_l$.
  $list = list \setminus \{L\}$.
  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

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.

4.4 Computational Experiments

In this section, we present the results of our extensive computational experiments from NESTA 0.3.0 archive [25]. 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 [73] 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 [74, 75]. 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 7.1 [2] is used to solve LPs, SOCPs and SDPs in our main algorithms. OPF Solver is run with MOSEK and SDPT3.

4.4.1 Methods

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

- McCormick Separation ($\mathsf{SEP}(\mathsf{M})$): We only separate the point from $\mathcal{M}_C^D$ defined in (100).
- SDP Separation ($\mathsf{SEP}(\mathsf{S})$): We only separate the point from $\mathcal{S}_C$ defined in (110).
- SDP + McCormick Separation ($\mathsf{SEP}(\mathsf{M},\mathsf{S})$): We separate the point from $\mathcal{M}_C^D$ and $\mathcal{S}_C$.

We used a fixed cycle basis to generate cutting planes for most instances. For small and difficult instances, we enlarge the set of cycles to obtain more cuts. After initial calibration, we decided to set number of bound tightening rounds T to 5, initial radius r_1 to 2, later radius r_2 to 4, and optimality tolerance ϵ to 10^{-3}. We also employed coefficient rounding for SDP cuts to improve numerical stability.

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

$$\Im(X_{ij}) - \tan\overline{\theta}_{ij}\Re(X_{ij}) \leq 0 \quad \text{and} \quad \Im(X_{ij}) - \tan\underline{\theta}_{ij}\Re(X_{ij}) \geq 0. \tag{119}$$

We run the OPF Solver with two solvers:

- MOSEK
- SDPT3

In the following, we compare our three methods with the SDP relaxation based approaches in both root node relaxation and after branching. We only used the instances from the NESTA archive with at least 1% SOCP optimality gap since we are only interested in the challenging instances.

4.4.2 Comparison to SDP Relaxation

We compared the relaxation values obtained from our approach to SDP relaxation in terms of the optimality gap, which is calculated as follows: $\%\text{gap} = 100 \times \frac{z^{\text{UB}} - z^{\text{LB}}}{z^{\text{UB}}}$. Here, z^{LB} is the optimal objective cost of a relaxation and z^{UB} is the objective cost of a feasible solution obtained by the local solver IPOPT [105].

4.4.2.1 Root Node Relaxation

In this section, we present the computational results in Table 13 for all the instances separately. Also, we provide a scatter plot Figure 8, which visualizes the average % gap and computational times.

Table 13 summarizes the results of our three methods applied only to the root node relaxation. We see that $\mathsf{SEP}(\mathsf{M})$ approach is both the most efficient and the weakest in terms of the optimality gap proven among the three methods whereas $\mathsf{SEP}(\mathsf{M},\mathsf{S})$ approach takes the longest computational times but provides the strongest relaxations overall. However, we should point out that this comes with an issue of numerical difficulties. Cutting planes obtained from the set $\mathcal{S}_C$ are not as reliable as the ones obtained from $\mathcal{M}_C^D$ since an SDP problem has to be solved in the former whereas only an LP is solved in the latter. Therefore, in $\mathsf{SEP}(\mathsf{S})$ and $\mathsf{SEP}(\mathsf{M},\mathsf{S})$ methods, respectively two and one large instances experience numerical issues.

Figure 8 compares our three methods against SDP Relaxation solved using OPF

Table 13: Root node relaxation results (OC: Operating Condition, *: numerical difficulty encountered).

		SEP(M)		SEP(S)		SEP(M, S)		SDP Relaxation	
OC	Instance	% gap	time	% gap	time	% gap	time	% gap	time
TYPICAL	3lmbd	0.09	0.59	0.08	0.42	0.07	0.97	0.39	2.32
	5pjm	5.18	1.20	3.62	1.11	1.84	1.79	5.22	0.97
	30ieee	0.06	13.87	0.01	7.07	0.01	8.02	0.00	1.57
	118ieee	0.42	129.61	0.15	166.53	0.15	169.67	0.07	6.31
	162ieee	2.06	606.08	1.51	606.57	1.50	516.97	1.12	17.86
	300ieee	0.22	243.07	0.09	192.66	0.10	128.66	0.08	16.23
	2383wp	0.86	197.91	0.71	278.97	0.62	286.46	0.37	851.73
CONGESTED	3lmbd	0.77	0.47	0.78	0.39	0.70	0.55	1.26	0.84
	6ww	0.08	1.22	0.00	1.09	0.00	1.39	0.00	0.92
	14ieee	0.12	5.76	0.01	2.93	0.01	4.18	0.00	1.48
	30as	0.40	19.22	0.28	13.39	0.26	18.42	0.00	1.8
	30fsr	10.01	17.44	5.24	17.97	5.22	20.93	11.06	2.02
	39epri	0.06	5.28	0.03	5.77	0.03	5.38	0.00	2.54
	118ieee	13.41	172.98	7.49	171.68	7.47	173.87	31.53	7.11
	162ieee	1.16	607.29	1.03	611.71	1.03	597.39	1.00	21.53
	189edin	0.51	102.75	0.11	96.69	0.11	67.98	0.05	6.41
	300ieee	0.21	239.42	0.09	188.65	0.09	185.37	0.00	14.41
	2383wp	0.78	198.54	0.78*	121.49	0.50	373.36	0.10	853.97
	2736sp	1.15	228.42	0.82	505.76	0.83	497.63	0.07	1430.13
	2737sop	0.89	239.74	0.48	494.38	0.59	497.11	0.01	1203.97
SMALL ANGLE	3lmbd	0.34	0.34	0.31	0.41	0.14	0.45	2.06	1.02
	4gs	0.08	0.36	0.05	0.19	0.05	0.23	0.05	0.88
	5pjm	0.10	0.30	0.08	0.31	0.08	0.34	0.00	0.95
	9wscc	0.05	0.44	0.05	0.44	0.05	0.53	0.00	0.85
	29edin	2.21	136.62	2.17	127.04	1.82	137.81	28.44	2.53
	30as	0.24	12.43	0.14	12.37	0.14	20.45	0.47	1.81
	30ieee	0.09	6.61	0.08	3.57	0.08	5.40	0.00	2.45
	118ieee	4.54	172.12	3.34	172.55	3.34	209.82	7.55	6.11
	162ieee	4.08	618.42	3.75	600.37	3.76	688.69	3.56	20.44
	189edin	1.21	107.44	1.13	77.26	1.06	124.30	1.20	6.27
	300ieee	0.22	246.93	0.10	254.45	0.10	190.36	0.13	15.16
	2383wp	2.61	231.87	2.45	229.66	2.27*	288.14	1.30	845.6
	2736sp	1.84	186.33	1.76	313.54	1.55	381.33	0.69	1364.31
	2737sop	2.14	174.36	2.14*	320.40	2.02	214.41	1.00	1294.13
	2746wp	1.64	195.51	1.31	398.51	1.36	314.85	0.43	1322.23
	Average	1.71	146.31	1.20	171.32	1.11	175.23	2.83	266.54

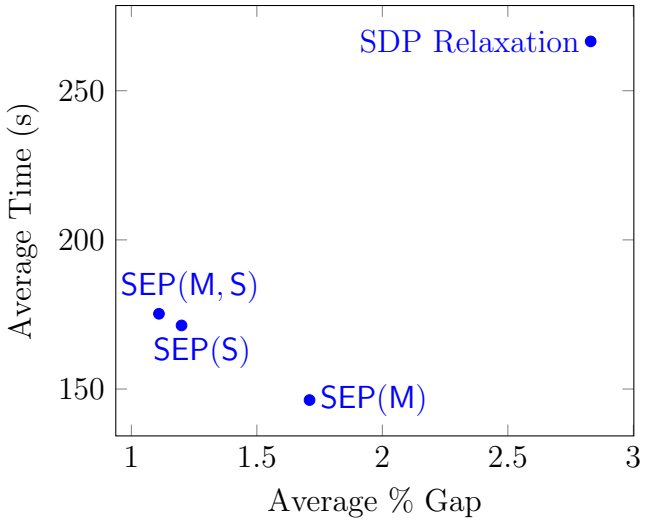


Figure 8: Scatter plot for root node relaxation results.

Solver with SDPT3 chosen as the solver. Although MOSEK is much faster with 143.64 seconds on the average than SDPT3 with 266.54 seconds, SDPT3 provides more accurate solutions with 2.83% optimality gap on the average while the optimality gap for MOSEK is 3.50%. Therefore, we will base our comparisons with SDPT3 results. We can easily see that all our methods dominate OPF Solver. Our methods are about 2-3 times more accurate than OPF Solver in terms of the average optimality gap proven and the computational times are about 50% less. 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 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 much stronger than a purely SDP based approach in about 45% of the computational time. Also, it is the most stable approach in terms of the numerical issues.

4.4.2.2 Effect of Branching

In this section, we present the results in Table 14 for the instances up to 300 buses which are not solved within the optimality threshold at the root node relaxation. We run the branch-and-cut algorithm with a budget of 15 minutes. We have not presented the results for larger instances since it is not possible to process a reasonable number of nodes in the alloted time.

Overall, the branching reduces the average optimality gaps from 1.71, 1.20 and 1.11 at the root node to 1.26, 0.80 and 0.79 after 15 minutes for the methods $\mathsf{SEP}(\mathsf{M})$, $\mathsf{SEP}(\mathsf{S})$ and $\mathsf{SEP}(\mathsf{M},\mathsf{S})$, 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. The average computation times increase from 146.31, 171.32 and 175.23 to 417.69, 322.83 and 323.83 after branching. We should point out that it is possible to process more nodes with $\mathsf{SEP}(\mathsf{M})$ 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 $\mathsf{SEP}(\mathsf{S})$ method is able to provide the same average optimality gap as $\mathsf{SEP}(\mathsf{M},\mathsf{S})$ at the end since it processes more nodes.

Table 14: Branch-and-cut results.

		SEP(M)			SEP(S)			SEP(M, S)		
OC	Instance	% gap	time	#nodes	% gap	time	#nodes	% gap	time	#nodes
TYPICAL	5pjm	0.10	170.86	591	0.10	82.82	329	0.10	74.86	224
	118ieee	0.25	907.58	110	0.09	235.57	11	0.09	240.82	11
	162ieee	1.74	900.40	10	1.35	925.87	10	1.28	926.37	19
CONGESTED	3lmbd	0.05	1.09	5	0.02	1.00	5	0.02	1.20	5
	14ieee	0.10	6.88	3						
	30as	0.15	902.26	412	0.10	96.75	59	0.10	98.98	57
	30fsr	4.87	901.75	379	0.65	900.13	415	0.70	900.34	372
	118ieee	11.02	900.30	85	5.23	904.83	78	5.27	905.89	82
	162ieee	1.14	908.88	10	0.99	932.17	10	0.98	929.08	10
	189edin	0.41	905.74	205	0.10	245.64	49	0.10	200.72	41
	300ieee	0.16	903.13	92						
SMALL ANGLE	3lmbd	0.04	0.69	3	0.04	0.72	3	0.04	0.76	3
	29edin	2.15	909.61	80	2.11	901.39	73	1.75	901.57	70
	30as	0.10	97.08	65	0.09	26.77	11	0.09	35.52	7
	118ieee	4.09	905.63	75	2.12	906.17	72	2.16	900.31	59
	162ieee	4.01	903.53	13	3.67	917.27	10	3.68	928.49	9
	189edin	0.95	902.21	189	0.87	901.58	187	0.87	904.84	177
	300ieee	0.17	906.46	86						

4.5 Conclusions

In this chapter, we solved the Optimal Power Flow Problem using global optimization methods. We first reformulated the OPF problem as a rank/minor constrained SDP problem and then proposed several convexification techniques for nonconvexities involving minor restrictions using only second-order conic and linear relaxations. We improved the resulting SOCP relaxation via cutting planes and convex envelopes by incorporating bound tightening. We proposed three methods with respect to the cutting plane procedure utilized. Our methods are quite successful in proving the global optimal solutions for many challenging OPF instances from the NESTA archive. Compared to SDP relaxation, our approaches provide about 2-3 times smaller optimality gaps with only half of the computation time on the average. 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 0.79% optimality gap in 323 second on the average for some of the most challenging OPF instances publicly available.

As a future work, we would like to apply the methodology developed in this chapter to multi-period Optimal Power Flow 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 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.

CHAPTER V

VALID INEQUALITIES FOR THE DC OPTIMAL TRANSMISSION SWITCHING PROBLEM

5.1 Introduction

Recent works that have first considered the DC OTS problem have found that the resulting MILP model is very challenging and called for a more systematic study on its underlying mathematical structure. For example, the authors of [41] state that

> "When solving the transmission switching problem, ... the techniques for closing the optimality gap, specifically improving the lower bound, are largely ineffective."

The primary focus of this chapter is an attempt to change this state of affairs by developing strong classes of valid inequalities that may be applied to power systems planning problems that involve the addition or removal of transmission elements, in order to solve such problems more efficiently. We summarize the main contributions as follows:

- Using a new formulation of DC OPF based on Kirchoff's Voltage Law, we give a cycle-based linear mixed integer programming (MIP) formulation for DC OTS.
- We formally establish that the DC OTS problem is NP-Hard, even if the interconnection network is a series-parallel graph and there is only one generation/demand pair. (Additional complexity results were recently independently established in [68]).
- Using the cycle-based formulation as inspiration, we derive classes of strong valid inequalities for a cycle–relaxation of the DC OTS. We additionally establish that

inequalities define the convex hull of the cycle-relaxation, and we show how to separate the inequalities over a given cycle in polynomial time.

- We perform computational experiments focusing on improving performance of integer programming-based methods for the OTS using the DC power flow approximation. We show that the new inequalities can help improve solution performance of commercial MIP software.

This chapter is organized as follows: in Section 5.2, we propose a new formulation for the DC OPF problem based on Kirchoff's voltage law. The formulation idea is also applied to the DC OTS problem. In Section 5.3, we prove some complexity results for DC OTS problem. In Section 5.4, we characterize the convex hull of a substructure in the formulation and develop valid inequalities that strengthen the LP relaxation of the proposed switching model. In Section 5.5, we give an efficient implementation of the proposed valid inequalities. We present extensive computational results in Section 5.6. Finally, Section 5.7 concludes this chapter with potential future research directions.

5.2 DC OPF and OTS Formulations Revisited

In Sections 1.1.3 and 1.2.2, we introduced the mathematical programming formulation of DC OPF and OTS problems, respectively. In this section, we first revisit these *angle-based* formulations and then, propose new *cycle-based* formulations.

In this chapter, we assume that the cost of production is a linear function of the production quantities at the generators, i.e. $C_i(p_i^g) := c_i p_i^g$ for all $i \in \mathcal{G}$. This assumption, while standard in the literature surrounding the DC OTS [33, 37], ignores the convex quadratic portion of generation costs. As we discuss in Section 5.4, our formulation and valid inequalities can still be applied when the cost is a nonlinear function.

5.2.1 Formulations for DC OPF

Recall that the standard formulation of DC OPF, which we call the Angle Formulation, is already introduced in 1.1.3. Now, we propose a new formulation, which we call the Cycle Formulation.

The DC power flow model gets its name from the fact that the equations describing the power flow in a network are the same as those that describe current flow in a standard direct current electric network. The constraints (3b) describe Kirchoff's Current Law (KCL) at each bus, and the equations (3c), which define the branch current, follow from Ohm's Law. With this analogy, it is natural to think about an alternative way to represent power flows in a DC circuit using the branch current f_{ij} and Kirchoff's Voltage Law (KVL). Kirchoff's Voltage Law states that around any *directed* cycle C, the voltage differences must sum to zero:

$$\sum_{(i,j)\in C} (\theta_i - \theta_j) = \sum_{(i,j)\in C} \frac{f_{ij}}{B_{ij}} = 0. \tag{120}$$

If the directed cycle C contains arc (i,j), but the transmission line has been given the alternate orientation ($(j,i) \in \mathcal{L}$), we adjust (120) by flipping the sign of the susceptance:

$$\bar{B}_{ij}^{C} = \begin{cases} B_{ij} & \text{if } (i,j) \in C, (i,j) \in \mathcal{L} \\ -B_{ij} & \text{if } (i,j) \in C, (j,i) \in \mathcal{L} \end{cases}$$

In this equivalent representation, the power flow should satisfy the KVL (120) for each cyle. Although the number of cycles in a network can be large, it is sufficient to enforce (120) over any set of cycles that forms a cycle basis $\mathcal{C}_b$ of the network (see Appendix B for a rigorous definition of cycle basis). If angle differences sum up to zero over the cycle basis, they also must sum up to zero over any other cycle (e.g. see [16]). Thus, the KVL-inspired formulation for the DC OPF is the following:

$$\min \ \sum_{i \in \mathcal{G}} c_i p_i^g \tag{121a}$$

$$\text{s.t.} \ \sum_{(i,j) \in C} \frac{f_{ij}}{\bar{B}_{ij}^C} = 0 \qquad C \in \mathcal{C}_b, \tag{121b}$$

$$(3b), (3d), (1h).$$

The voltage angles may be recovered using the equations (3c).

Proposition 23. *Formulations* (3) *and* (121) *are equivalent.*

Proof. In order to prove the equivalence of the formulations, it suffices to show that

$$\sum_{(i,j) \in C} \frac{f_{ij}}{\bar{B}_{ij}^C} = 0, \ C \in \mathcal{C}_b \iff \text{there exists } \theta \text{ such that } f_{ij} = B_{ij}(\theta_i - \theta_j), \ (i,j) \in \mathcal{L}. \tag{122}$$

($\Rightarrow$) First observe that because $\mathcal{C}_b$ is a cycle basis, equations (121b) imply $\sum_{(i,j) \in C} \frac{f_{ij}}{\bar{B}_{ij}^C} = 0$ for any cycle C. Now let $\mathcal{T} = (\mathcal{B}, \mathcal{L}')$ be a spanning tree of G. Clearly, the following system in θ has a solution:

$$B_{ij}(\theta_i - \theta_j) = f_{ij}, \ (i,j) \in \mathcal{L}'. \tag{123}$$

Therefore, it suffices to check if a solution of (123) satisfies $B_{ij}(\theta_i - \theta_j) = f_{ij}$ for $(i,j) \in \mathcal{L} \setminus \mathcal{L}'$. Note that for any $(k,l) \in \mathcal{L} \setminus \mathcal{L}'$, there exists a unique path $\mathcal{P}_{kl}$ from k to l in $\mathcal{T}$ and a cycle $C = \mathcal{P}_{kl} \cup \{(l,k)\}$. Then, we have

$$(\theta_k - \theta_l) = \sum_{(i,j) \in \mathcal{P}_{kl}} (\theta_i - \theta_j) = \sum_{(i,j) \in \mathcal{P}_{kl}} \frac{f_{ij}}{\bar{B}_{ij}^C} = \frac{f_{kl}}{\bar{B}_{kl}^C}, \tag{124}$$

which implies $B_{kl}(\theta_k - \theta_l) = f_{kl}$.

($\Leftarrow$) Suppose there exist θ such that $f_{ij} = B_{ij}(\theta_i - \theta_j)$, for each $(i,j) \in \mathcal{L}$. Then, for each cycle C, we have

$$\sum_{(i,j) \in C} \frac{f_{ij}}{\bar{B}_{ij}^C} = \sum_{(i,j) \in C} \frac{B_{ij}(\theta_i - \theta_j)}{\bar{B}_{ij}^C} = \sum_{(i,j) \in C} (\theta_i - \theta_j) = 0, \tag{125}$$

which concludes the proof. □

5.2.2 Formulations for DC OTS

The angle-based DC OPF formulation (3) can be easily adapted to switching by introducing binary variables x_{ij} that take the value 1 if line $(i,j) \in \mathcal{L}$ is on, and 0 if the line is disconnected. A direct nonlinear formulation of DC OTS is already mentioned in Section 1.2.2, where the nonlinearity stems from constraint (6). However, this constraint can be linearized as follows [33]:

$$B_{ij}(\theta_i - \theta_j) - M_{ij}(1 - x_{ij}) \leq f_{ij} \leq B_{ij}(\theta_i - \theta_j) + M_{ij}(1 - x_{ij}) \quad (i,j) \in \mathcal{L} \quad \text{(126a)}$$

$$-\bar{f}_{ij} x_{ij} \leq f_{ij} \leq \bar{f}_{ij} x_{ij} \quad (i,j) \in \mathcal{L} \quad \text{(126b)}$$

Here, M_{ij} is chosen sufficiently large to make the inequalities (126a) redundant if $x_{ij} = 0$. The constraints (126) ensure both that Ohm's Law is enforced if the line is switched on and that power flow $f_{ij} = 0$ if the line is switched off. After this linearization, the angle-based DC OTS becomes as follows:

$$\min \sum_{i \in \mathcal{G}} c_i p_i^g \quad \text{(127a)}$$

$$\text{s.t. } x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{L}, \quad \text{(127b)}$$

$$\text{(3b)}, \text{(1h)}, \text{(126a)} - \text{(126b)}.$$

Inspired by the cycle formulation (121) for the DC OPF, we can formulate the DC OTS problem without angle variables as well. The full formulation enforces Kirchoff's Voltage Law only if all arcs in a cycle are switched on.

$$\min \sum_{i \in \mathcal{G}} c_i p_i^g \quad \text{(128a)}$$

$$\text{s.t. } -M_C \sum_{(i,j) \in C} (1 - x_{ij}) \leq \sum_{(i,j) \in C} \frac{f_{ij}}{\bar{B}_{ij}^C} \leq M_C \sum_{(i,j) \in C} (1 - x_{ij}) \quad C \in \mathcal{C}, \quad \text{(128b)}$$

$$\text{(3b)}, \text{(1h)}, \text{(126b)} - \text{(127b)}.$$

The value M_C must be selected so that the inequalites (128b) are redundant if $\sum_{(i,j) \in C}(1 - x_{ij}) \geq 1$. In formulation (128), $\mathcal{C}$ is the set of *all cycles* in the graph

$\mathcal{N} = (\mathcal{B}, \mathcal{L})$. The cardinality of $\mathcal{C}$ is in general quite large, so we do not propose using (128) directly. Rather, we use the formulation (128) as the starting point for deriving strong valid inequalities in Section 5.4. Furthermore, the inequalities (128b) could be added as cuts within a branch-and-cut algorithm. These inequalities are required to define the feasible region, so the branch-and-cut procedure would search for a violated inequality from the class (128b) any time it identifies a candidate solution with the x components binary. (Inequalities added as cuts in this way are sometimes referred to as "lazy cuts".)

5.3 Complexity of DC OTS

In this section, we discuss the complexity of the DC optimal transmission switching problem. The input to the problem is a power network as described at the beginning of Section 5.2. In the *feasibility version* of DC OTS, we ask if there *exists* a subset of lines to switch off such that the DC OPF is feasible for the induced topology. The feasibility version of DC OTS with a cardinality constraint has been proven to be strongly NP-Complete in [13] by reduction from the Exact 3-Cover Problem. Recently, many complexity and approximability results on DC-Switching problems were given in [68], including the result that DC OTS is NP-Hard, even if the underlying graph is a cactus. Our results were established independently, and complement the results of [68] by formally establishing that the DC OTS problem is easy if the graph is a tree and NP-Hard even if there is one generation/load pair on series-parallel graphs.

Proposition 24. *In the DC OTS, there exists an optimal solution in which the lines switched on form a connected network.*

Proof. Consider the Cycle Formulation (128) of the DC OTS, and let $\mathcal{L}'$ be the active lines in an optimal solution. Assume that the network corresponding to this solution has k connected components. Since the original network $\mathcal{N} = (\mathcal{B}, \mathcal{L})$ is connected, we can find a set of transmission lines $\mathcal{L}''$ with cardinality $k-1$ such that $\mathcal{N}' = (\mathcal{B}, \mathcal{L}' \cup \mathcal{L}'')$

is connected. Now, let $x_{ij} = 1$ and $f_{ij} = 0$ for all $(i, j) \in \mathcal{L}''$. By construction, no new cycles are created by switching on lines in $\mathcal{L}''$. Further, the balance constraints (3b) and bound constraints (126b) are satisfied. Hence, we have demonstrated a new solution with the same objective value where the network formed by the active lines is connected. □

Corollary 1. *If $\mathcal{N} = (\mathcal{B}, \mathcal{L})$ is a tree, the DC OTS problem is solvable in polynomial time.*

Proof. Due to Proposition 24, there exists an optimal solution which induces a connected network. Since removing any line disconnects the tree, there exists an optimal solution in which all lines are active. But this is exactly the DC OPF problem without switching, which can be solved via linear programming, a problem known to be polynomially solvable. □

Theorem 5 establishes that DC OTS is NP-Complete even if the power network is a series-parallel graph, and there is only one demand-supply pair.

Theorem 5. *The feasibility version of DC OTS is NP-complete even when $\mathcal{N} = (\mathcal{B}, \mathcal{L})$ is a series-parallel graph, there is $|\mathcal{G}| = 1$ generator, and one node $i \in \mathcal{B}$ such that $p_i^d \neq 0$.*

Proof. We prove this result by a reduction from the subset sum problem, which is known to be NP-Complete [38]. Consider an instance of a subset problem as: Given $a_i \in \mathbb{Z}_{++}$ for $i \in \{1, \ldots, n\}$ and $b \in \mathbb{Z}_{++}$, does there exist a subset $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = b$? We construct an instance of switching problem as follows:

(i) There are $n + 3$ buses $\{0, 1, \ldots, n, n + 1, n + 2\}$.

(ii) Following are the lines: $(0, i)$ for all $i \in \{1, \ldots, n\}$; $(i, n+1)$ for all $i \in \{1, \ldots, n\}$; $(n + 1, n + 2)$; $(0, n + 2)$.

(iii) The capacities of the lines are: $\frac{a_i}{b}$ for the line $(0, i)$ and $(i, n+1)$ for all $i \in \{1, \ldots, n\}$; 1 for $(n+1, n+2)$ and $(0, n+2)$.

(iv) The susceptances of the lines are: $2a_i$ for the line $(0, i)$ and $(i, n+1)$ for all $i \in \{1, \ldots, n\}$; 1 for $(n+1, n+2)$; $\frac{b}{b+1}$ for $(0, n+2)$.

(v) There is a generation of 2 at bus 0 and load of 2 at bus $n+2$.

Clearly, the size of the instance of the switching problem is polynomial in the size of the given instance of the subset sum problem. Also note that the graph is a series parallel graph and there is only one demand supply pair.

We now verify that the subset sum problem is feasible if and only if the switching problem is feasible.

($\Rightarrow$): Since the subset sum problem is feasible, let $\sum_{i \in I} a_i = b$ where $I \subseteq \{1, \ldots, n\}$. Then construct a solution to the switching problem as follows: Switch off the lines $(0, i), (i, n+1)$ for $i \in \{1, \ldots, n\} \setminus I$. It is straightforward to establish that a feasible solution to the DC OTS exists. (In the solution, the angle at bus 0 is $1 + \frac{1}{b}$, the angle at bus i is $1 + \frac{1}{2b}$ for all $i \in I$, the angle at bus $n+1$ is 1, and the angle at bus $n+2$ is 0).

($\Leftarrow$): The subset sum problem is infeasible and assume by contradiction that the switching problem is feasible. Then note that the flow in arcs $(0, n+2)$ and $(n+1, n+2)$ are 1 each (and these lines are not switched off). Without loss of generality, let the angle at bus $n+2$ be 0. This implies that the angle at bus 0 is $1 + \frac{1}{b}$ and at bus $n+1$ is 1. Then note that if a pair of lines $(0, i), (i, n+1)$ is not switched off, this implies that the angle at bus i is $1 + \frac{1}{2b}$ and the resulting flow is $\frac{a_i}{b}$ along the path $(0, i), (i, n+1)$. Therefore, as a switching solution exists, we have that there exists some $I \subseteq \{1, \ldots, n\}$ such that the paths $(0, i), (i, n+1)$ for $i \in I$ are switched on (and others are switched off). Then $\sum_{i \in I} \frac{a_i}{b} = 1$ (by flow conservation at bus $n+1$), the required contradiction. $\square$

5.4 Valid Inequalities

In this section, we give two (symmetric) classes of inequalities for DC OTS that are derived by considering a relaxation of the cycle formulation (128). The inequalities are derived by projecting an extended formulation of our chosen relaxation. We additionally show that the inequalities define the convex hull of the relaxation and that each of the inequalities defines a facet of the relaxation. The separation problem for the new class of inequalities is a knapsack problem, but we show in Section 5.4.2 how to exploit the special structure of the knapsack to give a closed-form solution.

We remind the reader that the objective function is assumed to be linear, as opposed to convex quadratic. In general, finding the convex hull of the feasible region may not be as useful algorithmically when the objective function is nonlinear convex as the optimal solution may lie in the interior of the convex hull. However, the generation cost functions are usually convex increasing functions of p_i^g over the interval $[p_i^{\min}, p_i^{\max}]$ and, therefore, the optimal solutions of any convex relaxation will lie on the boundary of the relaxation. Thus, finding the convex hull of the feasible solutions may still be useful in improving bounds when we are working with a convex quadratic increasing objective function instead of a linear objective function. Nevertheless, we also point out that when the optimal solution is not an extreme point, the bound obtained by this approach might be weak. As future work, numerical experiments should be carried out to analyze this case empirically.

5.4.1 Derivation

Consider the constraints (128b), (126b) and (127b) in the cycle-based formulation for DC OTS for one specific cycle $C \in \mathcal{C}$ and define the following relaxation of the feasible region of (128):

$$\mathcal{S}_C = \{(f,x) : -M_C \sum_{(i,j)\in C}(1-x_{ij}) \le \sum_{(i,j)\in C}\frac{f_{ij}}{\bar{B}^C_{ij}} \le M_C \sum_{(i,j)\in C}(1-x_{ij}),$$
$$-\bar{f}_{ij}x_{ij} \le f_{ij} \le \bar{f}_{ij}x_{ij} \quad (i,j)\in C,\ x_{ij}\in\{0,1\} \quad (i,j)\in C\}. \tag{129}$$

In the remainder of this section, we assume that C is a directed cycle and hence, $\bar{B}^C_{ij} = B_{ij}$. Our main result in this section concerns the inequalities

$$\begin{aligned} -\Delta(S)(|C|-1) + \sum_{(i,j)\in S}[\Delta(S)-w_{ij}]x_{ij} + \Delta(S)\sum_{(i,j)\in C\setminus S} x_{ij} \\ \le \sum_{(i,j)\in S}\frac{f_{ij}}{B_{ij}} \le \\ \Delta(S)(|C|-1) - \sum_{(i,j)\in S}[\Delta(S)-w_{ij}]x_{ij} - \Delta(S)\sum_{(i,j)\in C\setminus S} x_{ij} \\ S\subseteq C \ \text{ s.t. } \ \Delta(S)>0, \end{aligned} \tag{130}$$

where $w_{ij} := \frac{\bar{f}_{ij}}{B_{ij}}$, and

$$w(S) := \sum_{(i,j)\in S} w_{ij} \qquad \text{for } S\subseteq C$$
$$\Delta(S) := w(S) - w(C\setminus S) = 2w(S) - w(C) \ \text{ for } S\subseteq C.$$

We show that the inequalities (130) are the only non-trivial inequalities defining $\text{conv}(\mathcal{S}_C)$.

Theorem 6.

$$conv(\mathcal{S}_C) = \{(f,x) : (130), -\bar{f}_{ij}x_{ij} \le f_{ij} \le \bar{f}_{ij}x_{ij},\ x_{ij}\le 1 \quad (i,j)\in C\} \tag{131}$$

In proving this result, for ease of presentation, we assume without loss of generality that $B_{ij} = 1$ for all $(i,j)\in\mathcal{L}$ by appropriately scaling $\bar{f}_{ij}$. The result is proven through a series of propositions using disjunctive arguments. Let us start with the following

linear system

$$-\bar{f}_{ij}x^1_{ij} \le f^1_{ij} \le \bar{f}_{ij}x^1_{ij} \qquad (i,j) \in C \tag{132a}$$

$$\sum_{(i,j)\in C} x^1_{ij} = |C| y_C \tag{132b}$$

$$\sum_{(i,j)\in C} f^1_{ij} = 0 \tag{132c}$$

$$0 \le x^1_{ij} \le y_C \qquad (i,j) \in C \tag{132d}$$

$$-\bar{f}_{ij}x^0_{ij} \le f^0_{ij} \le \bar{f}_{ij}x^0_{ij} \qquad (i,j) \in C \tag{132e}$$

$$\sum_{(i,j)\in C} x^0_{ij} \le (|C|-1)(1-y_C) \tag{132f}$$

$$0 \le x^0_{ij} \le 1 - y_C \qquad (i,j) \in C \tag{132g}$$

$$x_{ij} = x^1_{ij} + x^0_{ij} \qquad (i,j) \in C \tag{132h}$$

$$f_{ij} = f^1_{ij} + f^0_{ij} \qquad (i,j) \in C \tag{132i}$$

$$0 \le y_C \le 1, \tag{132j}$$

and define the polytope

$$\mathcal{E}_C = \{(f, f^1, f^0, x, x^1, x^0, y) : (132)\}.$$

Proposition 25. *System (132) is an extended formulation for conv($\mathcal{S}_C$). Furthermore, polytope $\mathcal{E}_C$ is integral so that we have conv($\mathcal{S}_C$) = proj$_{f,x}\mathcal{E}_C$.*

Proof. Let us first consider the following disjunction for cycle C: Either every line is active or at least one line is disconnected. If all the lines are active, then we have

$$-\bar{f}_{ij}x_{ij} \le f_{ij} \le \bar{f}_{ij}x_{ij} \qquad (i,j) \in C \tag{133a}$$

$$\sum_{(i,j)\in C} x_{ij} = |C| \tag{133b}$$

$$\sum_{(i,j)\in C} f_{ij} = 0 \tag{133c}$$

$$0 \le x_{ij} \le 1 \qquad (i,j) \in C \tag{133d}$$

Define polytope $\mathcal{S}_C^1 = \{(f, x) : (133)\}$. Note that $\mathcal{S}_C^1$ is integral in x since constraint (133b) forces $x_{ij} = 1$ for all $(i, j) \in C$ in a feasible solution.

Otherwise, at least one of the lines is inactive and we have

$$-\bar{f}_{ij}x_{ij} \leq f_{ij} \leq \bar{f}_{ij}x_{ij} \qquad (i,j) \in C \tag{134a}$$

$$\sum_{(i,j)\in C} x_{ij} \leq |C| - 1 \tag{134b}$$

$$0 \leq x_{ij} \leq 1 \qquad (i,j) \in C \tag{134c}$$

Define polytope $\mathcal{S}_C^0 = \{(f, x) : (134)\}$, which is again integral in x.

By construction, we have $\text{conv}(\mathcal{S}_C) = \text{conv}(\mathcal{S}_C^1 \cup \mathcal{S}_C^0)$. Let us duplicate variables (f, x) as (f^1, x^1) and (f^0, x^0) in the descriptions of $\mathcal{S}_C^1$ and $\mathcal{S}_C^0$, respectively. Then, by assigning a binary variable y_C to $\mathcal{S}_C^1$ and $1 - y_C$ to $\mathcal{S}_C^0$, we get system (132). So, it is an extended formulation for $\text{conv}(\mathcal{S}_C)$ [7].

Further, observe that $\mathcal{E}_C$ is the union of the convex hull of two polyhedra that are integral in x. Therefore, $\mathcal{E}_C$ must be integral in x as well. □

By noticing that $x_{ij}^1 = y_C$ and $x_{ij}^0 = x_{ij} - y_C$ for every $(f, f^1, f^0, x, x^1, x^0, y) \in \mathcal{E}_C$, we can simplify the notation by immediately projecting out these variables. Specifically, if we define the linear system

$$-\bar{f}_{ij}y_C \leq f_{ij}^1 \leq \bar{f}_{ij}y_C \qquad (i,j) \in C \tag{135a}$$

$$\sum_{(i,j)\in C} f_{ij}^1 = 0 \tag{135b}$$

$$-\bar{f}_{ij}(x_{ij} - y_C) \leq f_{ij}^0 \leq \bar{f}_{ij}(x_{ij} - y_C) \qquad (i,j) \in C \tag{135c}$$

$$\sum_{(i,j)\in C} x_{ij} - y_C \leq |C| - 1 \tag{135d}$$

$$y_C \leq x_{ij} \leq 1 \qquad (i,j) \in C \tag{135e}$$

$$f_{ij} = f_{ij}^1 + f_{ij}^0 \qquad (i,j) \in C \tag{135f}$$

$$0 \leq y_C \leq 1, \tag{135g}$$

we have that $\mathcal{P}_C := \{(f, f^1, f^0, x, y) : (135)\} = \text{proj}_{f,f^1,f^0,x,y}\mathcal{E}_C$. In Proposition 26 we can further project out the f^1 and f^0 variables by defining

$$-\sum_{(i,j)\in S} w_{ij}x_{ij} + \Delta(S)y_C \leq \sum_{(i,j)\in S} f_{ij} \leq \sum_{(i,j)\in S} w_{ij}x_{ij} - \Delta(S)y_C \quad S \subseteq C \text{ s.t. } \Delta(S) > 0. \tag{136}$$

Proposition 26. *proj*$_{f,x,y}\mathcal{P}_C = \{(f, x, y) : (126b), (136), (135d), (135e), y_C \geq 0\}$.

Proof. We begin by defining $\mathcal{Q} := \{(f, x, y) : (126b), (136), (135d), (135e), y_C \geq 0\}$, and let $(f, f^1, f^0, x, y) \in \mathcal{P}_C$. We claim that $(f, x, y) \in \mathcal{Q}$. For each line $(i, j) \in C$, summing (135a) and (135c) and using (135f) yields (126b). So, it suffices to check constraint (136). We have, for each $S \subseteq C$,

$$0 = \sum_{(i,j)\in C} f^1_{ij} = \sum_{(i,j)\in S} f^1_{ij} + \sum_{(i,j)\in C\setminus S} f^1_{ij}$$

due to (135b). Recall that by scaling $\bar{f}_{ij}$, we have assumed $B_{ij} = 1$, and thus $w_{ij} = \bar{f}_{ij}$ for all $(i, j) \in \mathcal{L}$. Combined with (135f), we have

$$\begin{aligned}
\sum_{(i,j)\in S} f_{ij} &= \sum_{(i,j)\in S} f^0_{ij} - \sum_{(i,j)\in C\setminus S} f^1_{ij} \\
&\leq \sum_{(i,j)\in S} \bar{f}_{ij}(x_{ij} - y_C) + \sum_{(i,j)\in C\setminus S} \bar{f}_{ij}y_C \qquad \text{due to (135c) and (135a)} \\
&= \sum_{(i,j)\in S} w_{ij}x_{ij} - \left(\sum_{(i,j)\in S} w_{ij} - \sum_{(i,j)\in C\setminus S} w_{ij}\right) y_C \\
&= \sum_{(i,j)\in S} w_{ij}x_{ij} - \Delta(S)y_C.
\end{aligned} \tag{137}$$

This is exactly the right inequality of (136). Note that although this inequality is valid for all $S \subseteq C$, the ones with $\Delta(S) \leq 0$ are dominated. In fact, due to (126b) for a subset $\bar{S}$ with $\Delta(\bar{S}) \leq 0$, we have

$$\sum_{(i,j)\in \bar{S}} f_{ij} \leq \sum_{(i,j)\in \bar{S}} w_{ij}x_{ij} \leq \sum_{(i,j)\in \bar{S}} w_{ij}x_{ij} - \Delta(\bar{S})y_C. \tag{138}$$

Using a symmetric argument, we can show the validity of the left inequality similarly. Hence, $\text{proj}_{f,x,y}\mathcal{P}_C \subseteq \mathcal{Q}$.

Next, we prove that any solution $(f, x, y) \in \mathcal{Q}$ can be extended by some (f^1, f^0) such that it satisfies (135). First, we can eliminate f^0 variables by setting $f^0_{ij} = f_{ij} - f^1_{ij}$. Then, for contradiction, assume that there exists $(f, x, y) \in \mathcal{Q}$ such that the following system in f^1 is infeasible:

$$-\bar{f}_{ij}y_C \le f^1_{ij} \le \bar{f}_{ij}y_C \qquad (i,j) \in C \tag{139a}$$

$$-\bar{f}_{ij}(x_{ij} - y_C) \le f_{ij} - f^1_{ij} \le \bar{f}_{ij}(x_{ij} - y_C) \qquad (i,j) \in C \tag{139b}$$

$$\sum_{(i,j)\in C} f^1_{ij} = 0. \tag{139c}$$

This system can be rewritten as

$$f^1_{ij} \le \bar{f}_{ij}y_C \qquad (i,j) \in C \tag{140a}$$

$$-f^1_{ij} \le \bar{f}_{ij}y_C \qquad (i,j) \in C \tag{140b}$$

$$f^1_{ij} \le f_{ij} + \bar{f}_{ij}(x_{ij} - y_C) \qquad (i,j) \in C \tag{140c}$$

$$-f^1_{ij} \le -f_{ij} + \bar{f}_{ij}(x_{ij} - y_C) \qquad (i,j) \in C \tag{140d}$$

$$\sum_{(i,j)\in C} f^1_{ij} = 0. \tag{140e}$$

By Farkas' Lemma, the following system must have a solution:

$$z := \sum_{(i,j)\in C} \left[\bar{f}_{ij}y_C\lambda^+_{ij} + \bar{f}_{ij}y_C\lambda^-_{ij} + (f_{ij} + \bar{f}_{ij}(x_{ij} - y_C))\mu^+_{ij} + (-f_{ij} + \bar{f}_{ij}(x_{ij} - y_C))\mu^-_{ij}\right]$$

$$z < 0 \tag{141a}$$

$$\lambda^+_{ij} - \lambda^-_{ij} + \mu^+_{ij} - \mu^-_{ij} + \gamma = 0 \quad (i,j) \in C \tag{141b}$$

$$\lambda^+_{ij}, \lambda^-_{ij}, \mu^+_{ij}, \mu^-_{ij} \ge 0 \qquad (i,j) \in C. \tag{141c}$$

Now, we consider the subsystem $\mathcal{T}_\gamma$ defined by (141b) and (141c) with respect to γ. Let us find the finite generators of this subsystem. There are three cases:

(i) $\gamma = 0$: In this case, $\mathcal{T}_0$ has a single extreme point at the origin and 4 extreme rays for each $(i,j) \in C$. Their generators with corresponding z values are given as follows:

(a) $(\lambda_{ij}^+, \lambda_{ij}^-, \mu_{ij}^+, \mu_{ij}^-) = (1,1,0,0)$: $z = 2\bar{f}_{ij} y_C \geq 0$

(b) $(\lambda_{ij}^+, \lambda_{ij}^-, \mu_{ij}^+, \mu_{ij}^-) = (1,0,0,1)$: $z = -f_{ij} + \bar{f}_{ij} x_{ij} \geq 0$

(c) $(\lambda_{ij}^+, \lambda_{ij}^-, \mu_{ij}^+, \mu_{ij}^-) = (0,1,1,0)$: $z = f_{ij} + \bar{f}_{ij} x_{ij} \geq 0$

(d) $(\lambda_{ij}^+, \lambda_{ij}^-, \mu_{ij}^+, \mu_{ij}^-) = (0,0,1,1)$: $z = 2\bar{f}_{ij}(x_{ij} - y_C) \geq 0$

So, in all cases, $z \geq 0$ and hence, there is no solution to (141) when $\gamma = 0$.

(ii) $\gamma < 0$: By scaling, we can assume that $\gamma = -1$. In this case, the extreme rays of $\mathcal{T}_{-1}$ are exactly the extreme rays of $\mathcal{T}_0$ while the extreme points of $\mathcal{T}_{-1}$ have the following structure: exactly one of λ_{ij}^+ and μ_{ij}^+ is 1 for each $(i,j) \in C$ and others are zero. Now, let us define $S = \{(i,j) \in C : f_{ij} + \bar{f}_{ij}(x_{ij} - y_C) \leq \bar{f}_{ij} y_C\}$ and calculate the value of z at the extreme point defined by S. Then, we have

$$\begin{aligned} z &\geq \sum_{(i,j)\in S} f_{ij} + \sum_{(i,j)\in S} w_{ij}(x_{ij} - y_C) + \sum_{(i,j)\in C\setminus S} w_{ij} y_C \\ &= \sum_{(i,j)\in S} f_{ij} + \sum_{(i,j)\in S} w_{ij} x_{ij} - \Delta(S) y_C \geq 0 \qquad \text{due to (136)} \end{aligned}$$

Hence, there is no solution to (141) when $\gamma < 0$.

(iii) $\gamma > 0$: By scaling, we can assume that $\gamma = 1$. In this case, the extreme rays of $\mathcal{T}_1$ are exactly the extreme rays of $\mathcal{T}_0$ while the extreme points of $\mathcal{T}_1$ have the following structure: exactly one of λ_{ij}^- and μ_{ij}^- is 1 for each $(i,j) \in C$ and others are zero. Now, let us define $S = \{(i,j) \in C : -f_{ij} + \bar{f}_{ij}(x_{ij} - y_C) \leq \bar{f}_{ij} y_C\}$ and calculate the value of z at the extreme point defined by S. Then, we have

$$\begin{aligned} z &\geq \sum_{(i,j)\in S} -f_{ij} + \sum_{(i,j)\in S} w_{ij}(x_{ij} - y_C) + \sum_{(i,j)\in C\setminus S} w_{ij} y_C \\ &= \sum_{(i,j)\in S} -f_{ij} + \sum_{(i,j)\in S} w_{ij} x_{ij} - \Delta(S) y_C \geq 0 \qquad \text{due to (136)} \end{aligned}$$

Hence, there is no solution to (141) when $\gamma > 0$.

□

We complete the proof of Theorem 6 by projecting out the y_C variable as well.

Proof of Theorem 6. By construction, we have that $\text{conv}(\mathcal{S}_C) = \text{proj}_{f,x}\mathcal{P}_C$. It suffices to show that $\text{conv}(\mathcal{S}_C) = \text{proj}_{f,x}\mathcal{P}_C = \mathcal{R}$ where $\mathcal{R} = \{(f,x) : (126b), (130), x_{ij} \leq 1 \;\; (i,j) \in C\}$.

First, let us rewrite inequalities involving y_C in $\text{proj}_{f,x,y}\mathcal{P}_C$:

$$0 \leq y_C \tag{142a}$$

$$\sum_{(i,j)\in C} x_{ij} - (|C| - 1) \leq y_C \tag{142b}$$

$$y_C \leq x_{ij} \qquad (i,j) \in C \tag{142c}$$

$$y_C \leq \frac{1}{\Delta(S)} \sum_{(i,j)\in S} w_{ij}x_{ij} + \frac{1}{\Delta(S)} \sum_{(i,j)\in S} f_{ij} \quad S \subseteq C, \Delta(S) > 0 \tag{142d}$$

$$y_C \leq \frac{1}{\Delta(S)} \sum_{(i,j)\in S} w_{ij}x_{ij} - \frac{1}{\Delta(S)} \sum_{(i,j)\in S} f_{ij} \quad S \subseteq C, \Delta(S) > 0. \tag{142e}$$

Now, we use Fourier-Motzkin elimination on y_C to show necessity and sufficiency of the convex hull description. Note that equation (142c) together with (142a) and (142b) give redundant inequalities dominated by $0 \leq x_{ij} \leq 1$. Similarly, (142a) with (142d) and (142e) exactly give equation (126b).

Next, we look at (142b) and (142e) together, which give

$$\begin{aligned}
&\sum_{(i,j)\in C} x_{ij} - (|C|-1) \leq \frac{1}{\Delta(S)} \sum_{(i,j)\in S} w_{ij}x_{ij} - \frac{1}{\Delta(S)} \sum_{(i,j)\in S} f_{ij} \\
\iff\ & \Delta(S) \sum_{(i,j)\in C} x_{ij} - \Delta(S)(|C|-1) \leq \sum_{(i,j)\in S} w_{ij}x_{ij} - \sum_{(i,j)\in S} f_{ij} \\
\iff\ & \sum_{(i,j)\in S} f_{ij} \leq \Delta(S)(|C|-1) + \sum_{(i,j)\in S} w_{ij}x_{ij} - \Delta(S) \sum_{(i,j)\in S} x_{ij} - \Delta(S) \sum_{(i,j)\in C\setminus S} x_{ij} \\
\iff\ & \sum_{(i,j)\in S} f_{ij} \leq \Delta(S)(|C|-1) - \sum_{(i,j)\in S} [\Delta(S) - w_{ij}]x_{ij} - \Delta(S) \sum_{(i,j)\in C\setminus S} x_{ij},
\end{aligned} \tag{143}$$

for $S \subseteq C, \Delta(S) > 0$. But, this is the right inequality in (130). Finally, using a similar argument, if we look at (142b) and (142d) together, we get exactly left inequality in (130) for $S \subseteq C, \Delta(S) > 0$, which concludes the proof. □

Theorem 6 shows that the inequalities (130) are sufficient to define the convex hull of $\mathcal{S}_C$. In Theorem 7 we show that each inequality is also necessary by showing that they always define facets.

Theorem 7. *The inequalities* (130) *are facet-defining for the set conv*$(\mathcal{S}_C)$.

Proof. See [62]. □

5.4.2 Separation

We now investigate the separation problem over constraints (130). Given a fractional point $(\hat{f}, \hat{x})$, let us first define $K_C = 1 - \sum_{(i,j)\in C}(1 - \hat{x}_{ij})$. We focus on the right inequality in (130); the left is analyzed similarly. For the given cycle C and $S \subseteq C$ with $\Delta(S) > 0$ the inequality takes the form:

$$\sum_{(i,j)\in S} f_{ij} + \sum_{(i,j)\in S} [\Delta(S) - w_{ij}]x_{ij} + \Delta(S) \sum_{(i,j)\in C} x_{ij} \leq \Delta(S)(|C|-1).$$

For convenience, we rearrange the inequality as follows:

$$\sum_{(i,j)\in S} (f_{ij} - w_{ij}x_{ij}) + \Delta(S)\Big(1 - \sum_{(i,j)\in C} (1 - x_{ij})\Big) \leq 0. \tag{144}$$

Our aim is to determine if there exists $S \subseteq C$ with $\Delta(S) > 0$ such that

$$\text{viol}(S) := \sum_{(i,j)\in S} (\hat{f}_{ij} - w_{ij}\hat{x}_{ij}) + \Delta(S)K_C > 0. \tag{145}$$

Recalling that $\Delta(S) = w(S) - w(C \setminus S) = 2w(S) - w(C)$, this can be determined by solving

$$\max_{S\subseteq C}\Big\{ \sum_{(i,j)\in S} (\hat{f}_{ij} - w_{ij}\hat{x}_{ij}) + \Delta(S)K_C : \Delta(S) > 0\Big\}. \tag{146}$$

A violated inequality exists if and only if the optimal value of (146) is positive. Introducing binary variables z_{ij} for $(i,j) \in C$ to indicate whether or not line $(i,j) \in S$, (146) can be reformulated as follows:

$$\begin{aligned}
&\max_{z\in\{0,1\}^{|C|}}\Big\{ \sum_{(i,j)\in C} (\hat{f}_{ij} - w_{ij}\hat{x}_{ij})z_{ij} + \sum_{(i,j)\in C} w_{ij}K_C z_{ij} - \sum_{(i,j)\in C} w_{ij}K_C(1 - z_{ij}) : \\
&\qquad\qquad \sum_{(i,j)\in C} w_{ij}z_{ij} - \sum_{(i,j)\in C} w_{ij}(1 - z_{ij}) > 0 \Big\} \\
&= -w(C)K_C + \max_{z\in\{0,1\}^{|C|}}\Big\{ \sum_{(i,j)\in C} \Big(\hat{f}_{ij} - w_{ij}\hat{x}_{ij} + 2w_{ij}K_C\Big) z_{ij} : \sum_{(i,j)\in C} w_{ij}z_{ij} > \frac{1}{2}w(C)\Big\}.
\end{aligned} \tag{147}$$

Note that here we do need the condition that $\Delta(S) > 0$, which yields a knapsack problem. A similar minimization problem can be posed to separate left inequalities.

There is a necessary condition for a cycle C to have a violating inequality of form (130) given in Proposition 27.

Proposition 27. *Given $(\hat{f}, \hat{x})$ for a cycle C, if $K_C = 1 - \sum_{(i,j)\in C}(1 - \hat{x}_{ij}) \leq 0$, then inequalities (130) are not violated.*

Proof. Consider right inequalities first. If $K_C < 0$, then given an optimal solution z to problem (147), we have $\sum_{(i,j)\in C} \left(\hat{f}_{ij} - w_{ij}\hat{x}_{ij} + 2w_{ij}K_C\right) z_{ij} < w(C)K_C$ since $\sum_{(i,j)\in C} w_{ij}z_{ij} > \frac{1}{2}w(C)$ and $f_{ij} \leq w_{ij}x_{ij}$. Therefore, no violating inequality exists. If $K_C = 0$, then given an optimal solution z to problem (147), we have $\sum_{(i,j)\in C} \left(\hat{f}_{ij} - w_{ij}\hat{x}_{ij} + 2w_{ij}K_C\right) z_{ij} \leq 0 = w(C)K_C$ since $\hat{f}_{ij} \leq w_{ij}\hat{x}_{ij}$. Therefore,

no violating inequality exists. A similar argument can be used to show that the requirement $K_C > 0$ is necessary for a left inequality to be violating. □

Although (147) formulates the separation problem of inequalities (130) as a knapsack problem, the special structure of this knapsack problem enables it to be solved efficiently. In fact, we could solve a linear program derived from the extended formulation (135) to solve the separation problem over $\text{conv}(\mathcal{S}_C)$. However, we next show how separation of the cycle inequalities can be accomplished efficiently in closed form. Define

$$S_C^* = \{(i,j) \in C : \hat{f}_{ij} - w_{ij}\hat{x}_{ij} + 2w_{ij}K_C > 0\}.$$

The following proposition shows that S_C^* is the only subset that needs to be considered when solving the separation problem for cycle C.

Proposition 28. *Assume $K_C > 0$. If there is any $S \subseteq C$ with $\Delta(S) > 0$ and $\text{viol}(S) > 0$, then the separation problem (146) is solved by S_C^*.*

Proof. See [62]. □

Recall from Proposition 27 that $K_C > 0$ is a necessary condition for a violated inequality to exist from cycle C. Thus, for a given cycle C with $K_C > 0$, a violated inequality exists for this cycle if and only if:

$$\sum_{(i,j)\in C} \Big((\hat{f}_{ij} - w_{ij}\hat{x}_{ij} + 2w_{ij}K_C)_+ - w_{ij}K_C \Big) > 0. \tag{148}$$

where $(\cdot)_+ = \max\{\cdot, 0\}$. The separation problem then reduces to a search for a cycle C having $K_C > 0$ and that satisfies (148).

5.5 Algorithms

In this section, we describe our algorithmic framework for solving the DC switching problem, beginning with section 5.5.1 where we describe the overall algorithm. Section 5.5.2 describes algorithms we implemented for separating over cycle inequalities (130)

for a fixed cycle C, and section 5.5.3 describes our procedure for generating a set of cycles over which we will perform the separation.

5.5.1 Overall Algorithm

The overall structure of the proposed algorithm is shown in Algorithm 7. The preprocessing phase of the algorithm aims to add cycle inequalities to strengthen the LP relaxation of the Angle Formulation (127). In particular, we first generate a set of cycles over the original power network, and solve the LP relaxation of (127). Then, a separation algorithm finds all the cycle inequalities (130) that are violated by the LP solution over the generated set of cycles. These violated inequalities are added to the LP relaxation as cuts. This procedure is iterated for a number of times to strengthen the LP relaxation, which is then fed to the MIP solver (we use CPLEX). We prefer to implement our separation in this "cut-and-branch" manner in order to investigate the utility of the cutting planes when combined with all advanced features of modern MIP software. An alternative approach would be to the use User Cut callback facility of CPLEX. However, this procedure disables many advanced features of the solver such as dynamic search. We implemented a version of our cutting planes using the CPLEX callback features and found that on average the performance was around 30% worse than with our cut-and-branch Algorithm 7.

Algorithm 7 Overall Algorithm for DC Switching

(i) Preprocessing:

  (a) Generate a set Γ of cycles (Cycle basis generation Algorithm 12).

  (b) Strengthen LP relaxation of Angle Formulation (127) by adding violated cycle inequalities (130) from each cycle $C \in \Gamma$ (Separation Algorithms 8 or 9).

(ii) Solve the Angle Formulation with added cuts using CPLEX.

5.5.2 Separation Algorithms

For a given LP relaxation solution, the separation algorithm implements the ideas presented in Section 5.4.2 to identify all violated cycle constraints of the form (130) for a predetermined set of cycles. The procedure is summarized in Algorithm 8.

Algorithm 8 Separation Algorithm

Given a set Γ of cycles and a LP relaxation solution $(\hat{f}, \hat{x})$.
for every cycle $C \in \Gamma$ **do**
  Compute $K_C = 1 - \sum_{a \in C}(1 - \hat{x}_a)$.
  if $K_C > 0$ **then**
    For each $a \in C$, compute $z_a = \begin{cases} 1 & \text{if } \hat{f}_a - w_a \hat{x}_a + 2 w_a K_C \geq 0 \\ 0 & \text{otherwise} \end{cases}$
    if $\sum_{a \in C} w_a z_a > \frac{1}{2} w(C)$ **and** $\sum_{a \in C} \Big((\hat{f}_a - w_a \hat{x}_a + 2 w_a K_C)_+ - w_a K_C \Big) > 0$ **then**
      A violated cycle inequality for C is found.
    end if
  end if
end for

Algorithm 8 generates a *single* violating inequality for each cycle, if such a violated inequality exists. However, the method can be extended to find all violating inequalities for a cycle. This procedure is summarized in Algorithm 9, which uses a recursive subroutine described in Algorithm 10.

Algorithm 9 Finding all valid inequalities.

Given a cycle C, define $v_{ij} = \hat{f}_a - w_a \hat{x}_a + 2 w_a K_C$ for $a \in C$
Set $S = \{a \in C : v_a \geq 0\}$ and denote $C \setminus S = \{a_1, \ldots, a_n\}$
Calculate $v(S) = \sum_{a \in S} v_a$ and $w(S) = \sum_{a \in S} w_a$
Recursion(S, 0)

5.5.3 Cycle Generation Algorithm

This section discusses how to generate a set of cycles for the preprocessing phase of Algorithm 7. The number of cycles in a graph $G = (V, E)$ grows exponentially in $|V|$, so in computations, finding all cycles is not efficient. Instead, we find a cycle basis for the original power network and use the cycle basis to generate cycles for

Algorithm 10 Recursion(S, k)

if $v(S) \leq w(C)K_C$ or $k = n$ **then**
  Stop.
end if
if $v(S) > w(C)K_C$ and $w(S) > \frac{1}{2}w(C)$ **then**
  A violating inequality is found.
end if
for $l = k+1, \ldots, n$ **do**
  Recursion($S \cup \{a_l\}$, l)
end for

separation. There are many algorithms for finding a cycle basis [56]. We use a simple algorithm based on the LU decomposition of the incidence matrix of the graph G. See Appendix B for details.

Given an initial set of cycles Γ coming from the cycle basis, the following procedure is used to generate additional cycles from which we may apply the separation procedures 8 and 9. Any pair of cycles in $\Gamma := \mathcal{C}^0$ that share at least one common edge can be combined to form a new cycle by removing the common edges. Denote $\bar{\mathcal{C}}^0$ as the set of all the new cycles thus generated from $\mathcal{C}^0$. Then, the set $\mathcal{C}^1 := \mathcal{C}^0 \cup \bar{\mathcal{C}}^0$ has more cycles than the cycle basis $\mathcal{C}_b$. This process can be repeated to generate sets $\mathcal{C}^{k+1} := \mathcal{C}^k \cup \bar{\mathcal{C}}^k$ for $k \geq 1$.

Given a set of cycles Γ, we can use Algorithms 8 or 9 to identify and add all violated cycle inequalities for that set of cycles to the LP relaxation of the Angle Formulation. We solve this strengthened LP relaxation again and add further violated cycle inequalities. This procedure can be carried out in several iterations (five times for our experiments) to produce a strengthened LP relaxation that is eventually passed to the MIP solver.

5.6 Computational Experiments

In this section, we present extensive computational studies that demonstrate the effectiveness of our proposed cut-and-branch algorithms on the DC OTS problem.

Section 5.6.1 explains how the test instances are generated. Section 5.6.2 compares the default branch-and-cut algorithm of CPLEX with two algorithms that employ the cycle inequalities (130). The first algorithm generates inequalities from cycles in one fixed cycle basis, and the other generates inequalities from a larger set of cycles. The results show that the proposed algorithm with cutting planes separated from more cycles consistently outperforms the default algorithm in terms of the size of the branch-and-bound tree, the total computation time, and the number of instances solved within the time limit.

For all experiments, we use a single thread in a 64-bit computer with Intel Core i5 CPU 3.33GHz processor and 4 GB RAM. Codes are written in the `C#` language. Considering that the transmission switching problem is usually solved under a limited time budget, the relative optimality gap is set to a moderate amount of 0.1% for all MIPs solved using CPLEX 12.4 [1]. We set a time limit of one hour in all experiments.

5.6.1 Instance Generation

Our computational experiments focus on instances where the solution of the DC OTS is significantly different than the solution of the DC OPF. In addition to selecting instances where switching made an appreciable instance, we selected instances whose network size was large enough so that the instances were not trivial for existing algorithms, but small enough to not be intractable. The 118-bus instance case118B generated in [15] turns out to be suitable for our purposes, and we also modified the 300-bus instance case300 so that transmission switching produces meaningfully different solutions from the OPF problem without switching. Furthermore, in order to extensively test the effectiveness of the proposed cuts and separation algorithms, we generate the following five sets of instances based on case118B and case300:

- Set 118_15: We generate 35 instances by modifying case118B, where the load at each bus of the original case118B is increased by a discrete random variable

following a uniform distribution on $[0, 15]$.

- Set 118_15_6: To each of the instances in Set 118_15, we randomly add 5 new lines to the power network, each creating a 6-cycle. The transmission limits for the lines in the cycle are set to 30% of the smallest capacity $\bar{f}_{ij}$ in the network, and B_{ij} is chosen randomly from one of the lines which is already in the original network.

- Set 118_15_16: The instances are constructed the same as in Set 118_15_6, except that a 16-cycle is created by adding 5 new transmission lines.

- Set 118_9G: We generate 35 different instances from case118B, where the original load at each bus is increased by a discrete random variable following a uniform distribution on $[0, 9]$. Furthermore, the generation topology of the network is changed. In particular, a generator located at bus i is moved to one of its neighboring buses or stays at its current location with equal probability.

- Set 300_5: We generate 35 different instances from case300, where the original load at each node is incremented by a discrete random variable following a uniform distribution on $[-5, 5]$. Also, eight generators are turned off and the cost coefficients of remaining generators are updated to be similar to the objective coefficients in [15]. Finally, more restrictive transmission line limits are imposed.

These instances can be downloaded from `https://sites.google.com/site/burakkocuk/research`.

5.6.2 DC Transmission Switching

We now investigate the computational impact of using the proposed valid inequalities within the proposed cut-and-branch procedure. We compare the following three solution procedures:

(i) The angle formulation (127) solved with CPLEX, abbreviated by Default.

(ii) The angle formulation (127) with valid inequalities (130) found using cycles coming from a single cycle basis, abbreviated by BasicCycles.

(iii) The angle formulation (127) with valid inequalities (130) coming from more cycles than a cycle basis, abbreviated by MoreCycles. The procedure for generating additional cycles for separation is discussed in Section 5.5.3. We use the set of cycles $\mathcal{C}^2$ for the 118-bus networks, where $|\mathcal{C}^2| \approx 3500$. For the 300-bus networks, $\mathcal{C}^2$ has more than $37,000$ cycles, which makes the separation procedure quite computationally expensive. For the 300-bus networks, we select 10% of the cycles in $\mathcal{C}^2$ randomly for separation.

We conducted preliminary experiments comparing Algorithms 8 and 9 when using the valid inequalities (130), and found Algorithm 9 yielded consistently better performance. Therefore, we use Algorithm 9 as the separation algorithm for inequalities (130) in both BasicCycles and MoreCycles.

Tables 15-19 show the computational results for the five sets of test instances described in Section 5.6.1. To measure the impact of the cuts on closing the integrality gap, we use the following objective values:

- z_{LP}: the objective value of the LP relaxation at the root node without inequalities (130) and without CPLEX cuts;
- z_{LP}^{cuts}: the objective value of the LP relaxation at the root node with inequalities (130) and without CPLEX cuts;
- z_{LP}^{root}: the objective value of the LP relaxation at the root node with inequalities (130) and with CPLEX cuts;
- z_{IP}: the objective value of the final integer solution of the switching problem.

The integrality gap measures reported in the tables are defined as

- "Initial LP Gap (%)" := $100\% \times (z_{IP} - z_{LP})/z_{IP}$;
- "Gap Closed by Cuts (%)" := $100\% \times (z_{LP}^{cuts} - z_{LP})/(z_{IP} - z_{LP})$;
- "Root Gap Closed (%)" := $100\% \times (z_{LP}^{root} - z_{LP})/(z_{IP} - z_{LP})$.

Other performance metrics reported in the tables are the average number of valid inequalities generated by the proposed algorithm (row "# Cuts"); the average preprocessing time for generating valid inequalities ("Preprocessing Time"), which includes the time for solving five rounds of the LP relaxation of the switching problem and the associated separation problems; the average total computation time including the preprocessing time ("Total Time"); the number of Branch-and-Bound nodes ("B&B Nodes"); the number of unsolved instances within a time limit of one hour ("# Unsolved"); and the average final optimality gap for unsolved instances ("Unsolved Opt Gap (%)"). For each metric, we report both the arithmetic mean (the first number) and the geometric mean (the second number).

From these tables, we can see that the proposed algorithm MoreCycles consistently outperforms the default algorithm in terms of the percentage of optimality gap closed at the root note, the size of the Branch-and-Bound tree, the total computation time, and the number of instances solved.

Figures 9-13 show the performance profiles of the three algorithms on the five sets of test instances. In particular, each curve in a performance profile is the cumulative distribution function for the ratio of one algorithm's runtime to the best runtime among the three [29].

Set 118_15 is a relatively easy test set. Figure 9 shows that for 42.9% of the instances, the Default algorithm is the fastest algorithm, the BasicCycles algorithm is fastest on 37.1%, and the MoreCycles algorithm is fastest on 20%. However, if we choose being within a factor of two of the fastest algorithm as the comparison criterion, both BasicCycles and MoreCycles surpass Default. BasicCycles solves all the instances and has the dominating performance for this set of instances.

Table 15: Summary of results for Set 118_15. Initial LP Gap (%): 26.87/26.57

	Default	BasicCycles	MoreCycles
# Cuts	-	32.91/31.64	218.66/190.10
Preprocessing Time (s)	-	0.05/0.04	0.80/0.44
Gap Closed by Cuts (%)	-	1.50/0	2.90/0
Root Gap Closed (%)	4.41/0	7.84/7.32	18.43/17.33
Total Time (s)	437.75/35.39	26.12/15.71	234.43/30.51
B&B Nodes	6.1E+5/3.6E+4	3.6E+4/1.5E+4	2.7E+5/2.2E+4
# Unsolved	2	0	1
Unsolved Opt Gap (%)	0.30/0.29	0/0	0.11/0.11

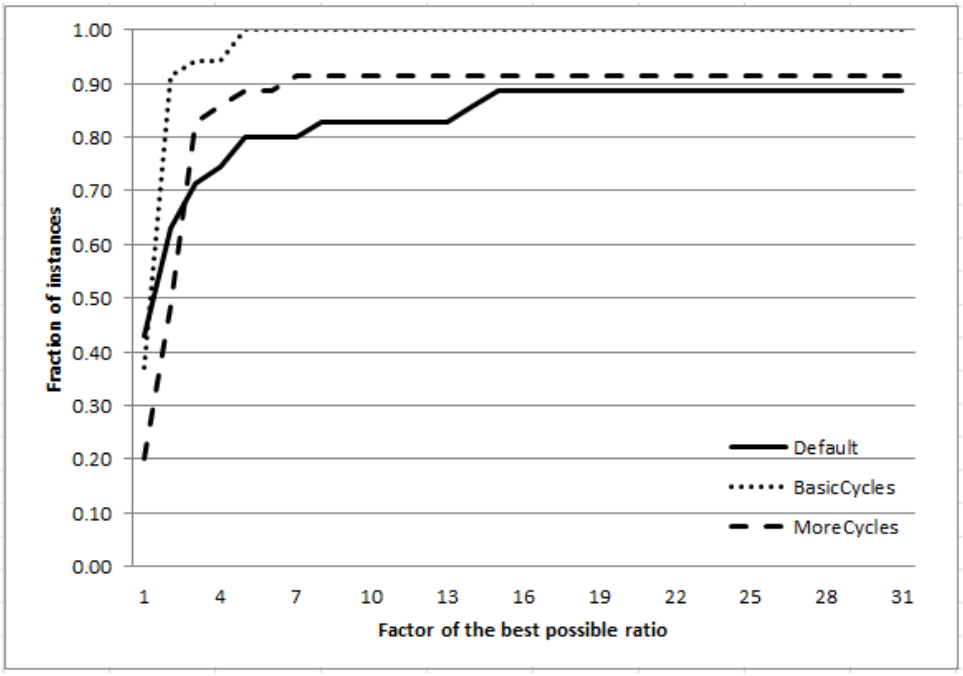


Figure 9: Performance profile for Set 118_15.

Figure 10 shows the results for Set 118_9G. BasicCycles is the fastest algorithm in 40% of the instances; MoreCycles and Default have the success rate of 20% of being the fastest. If we choose being within a factor of four of the fastest algorithm as the interest of comparison, then MoreCycles starts to outperform BasicCycles. Also, MoreCycles solves 74.3% of the instances, which is the highest among the three.

Table 16: Summary of results for Set 118_9G. Initial LP Gap (%): 19.12/13.37

	Default	BasicCycles	MoreCycles
# Cuts	-	28.37/27.63	150.31/139.02
Preprocessing Time (s)	-	0.05/0.04	1.14/0.32
Gap Closed by Cuts (%)	-	5.43/0	10.83/0
Root Gap Closed (%)	13.44/0	22.26/0	27.24/0
Total Time (s)	1126.54/148.65	1170.22/129.52	951.47/121.90
B&B Nodes	1.9E+6/2.1E+5	1.9E+6/1.8E+5	1.3E+6/1.5E+5
# Unsolved	10	10	7
Unsolved Opt Gap (%)	0.71/0.45	0.79/0.36	0.50/0.33

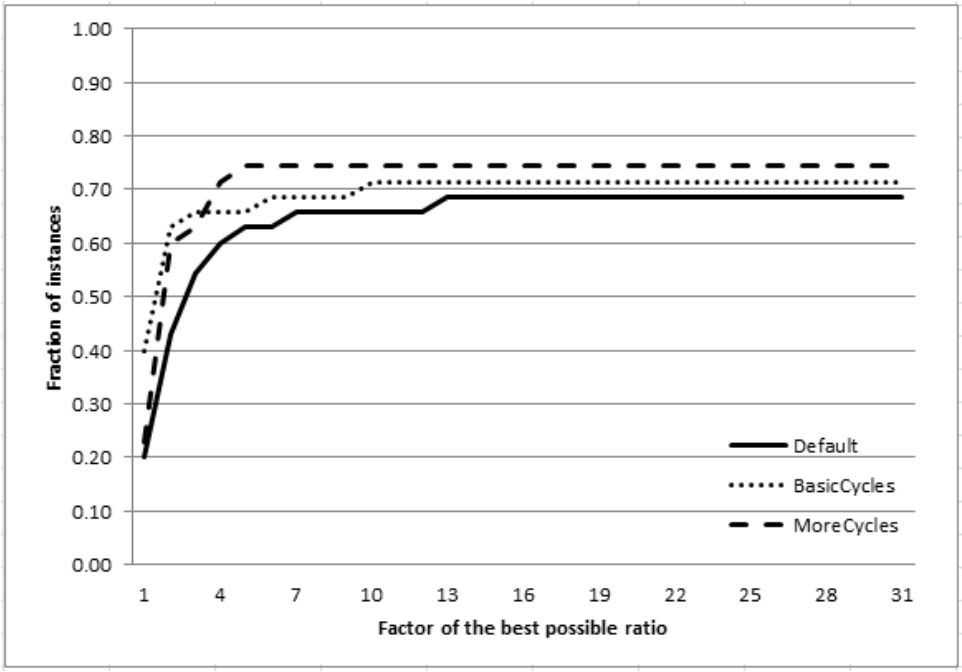


Figure 10: Performance profile for Set 118_9G.

For instance sets 118_15_6 and 118_15_16, Figures 11-12 show that BasicCycles is the fastest algorithm in the highest percentage of instances. For the ratio factor higher

than one, BasicCycles and MoreCycles clearly dominate Default, and both solve significantly more instances than Default. MoreCycles is the most robust algorithm in the sense that it solves the most instances.

Table 17: Summary of results for Set 118_15_6. Initial LP Gap (%): 26.76/26.48

	Default	BasicCycles	MoreCycles
# Cuts	-	29.34/28.91	145.20/141.54
Preprocessing Time (s)	-	0.11/0.03	1.30/0.44
Gap Closed by Cuts (%)	-	1.50/0	2.92/0
Root Gap Closed (%)	5.42/0	7.80/7.32	18.19/17.00
Total Time (s)	901.31/124.72	506.40/55.70	515.05/72.37
B&B Nodes	1.3E+6/1.4E+5	4.5E+5/4.9E+4	6.1E+5/6.6E+4
# Unsolved	5	3	1
Unsolved Opt Gap (%)	1.31/0.76	1.87/1.68	0.13/0.13

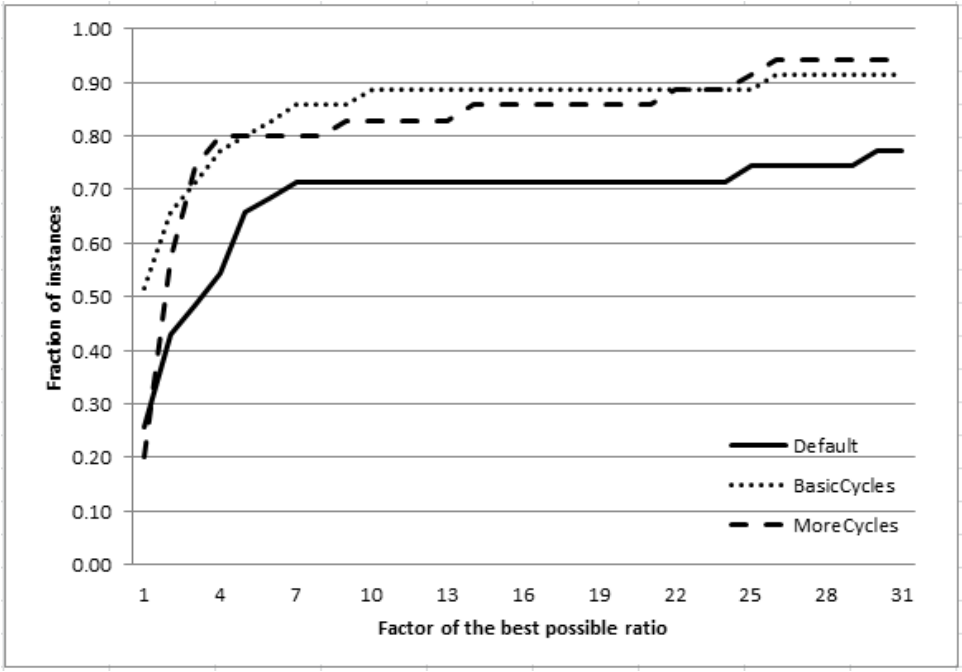


Figure 11: Performance profile for Set 118_15_6.

Table 18: Summary of results for Set 118_15_16. Initial LP Gap (%): 3.25/2.81

	Default	BasicCycles	MoreCycles
# Cuts	-	26.54/25.85	86.23/84.23
Preprocessing Time (s)	-	0.11/0.05	0.54/0.31
Gap Closed by Cuts (%)	-	0.05/0	0.31/0
Root Gap Closed (%)	0.47/0	4.38/0	11.65/0
Total Time (s)	2243.71/1750.82	1473.52/924.64	1581.60/1170.37
B&B Nodes	2.0E+6/1.5E+6	1.2E+6/8.1E+5	1.2E+6/9.0E+5
# Unsolved	13	4	3
Unsolved Opt Gap (%)	0.54/0.41	0.93/0.74	1.22/0.73

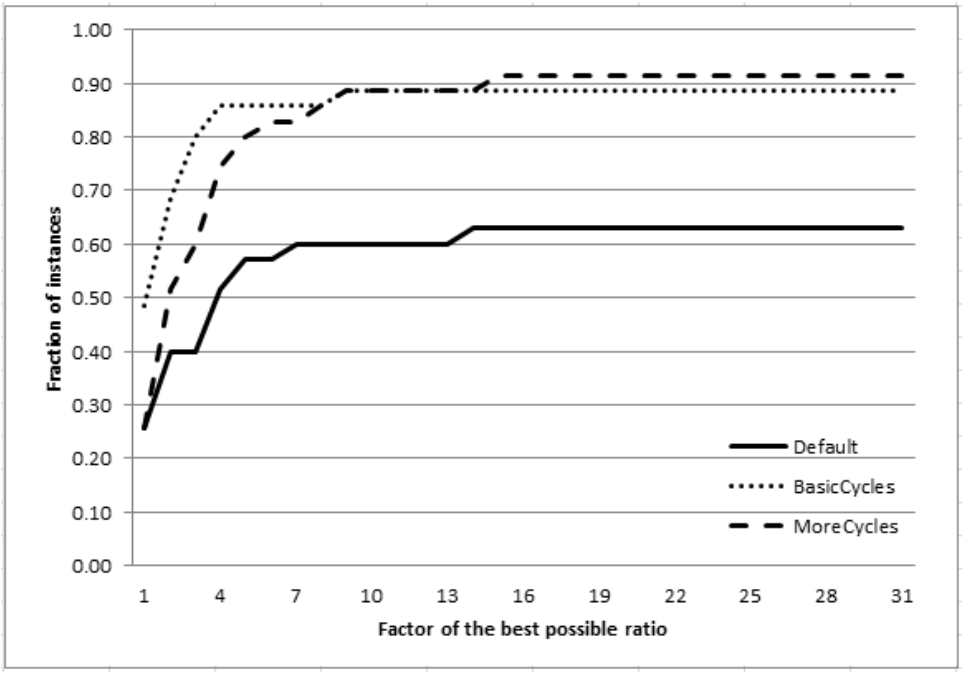


Figure 12: Performance profile for Set 118_15_16.

On the 300_5 instance set, Figure 13 shows that MoreCycles is the fastest in the largest fraction of instances and it also solves the most instances. BasicCycles is dominated by the other two methods for this set of instances.

Table 19: Summary of results for Set 300_5. Initial LP Gap (%): 4.88/4.88

	Default	BasicCycles	MoreCycles
# Cuts	-	15.66/15.26	34.83/33.56
Preprocessing Time (s)	-	0.09/0.06	0.48/0.43
Gap Closed by Cuts (%)	-	7.26/7.25	7.26/7.25
Root Gap Closed (%)	7.11/4.17	48.37/48.28	48.39/48.30
Total Time (s)	1685.39/634.75	1940.16/841.88	1524.14/514.76
B&B Nodes	6.6E+5/2.3E+5	7.8E+5/3.1E+5	6.2E+5/1.9E+5
# Unsolved	13	16	10
Unsolved Opt Gap (%)	0.21/0.19	0.22/0.21	0.40/0.23

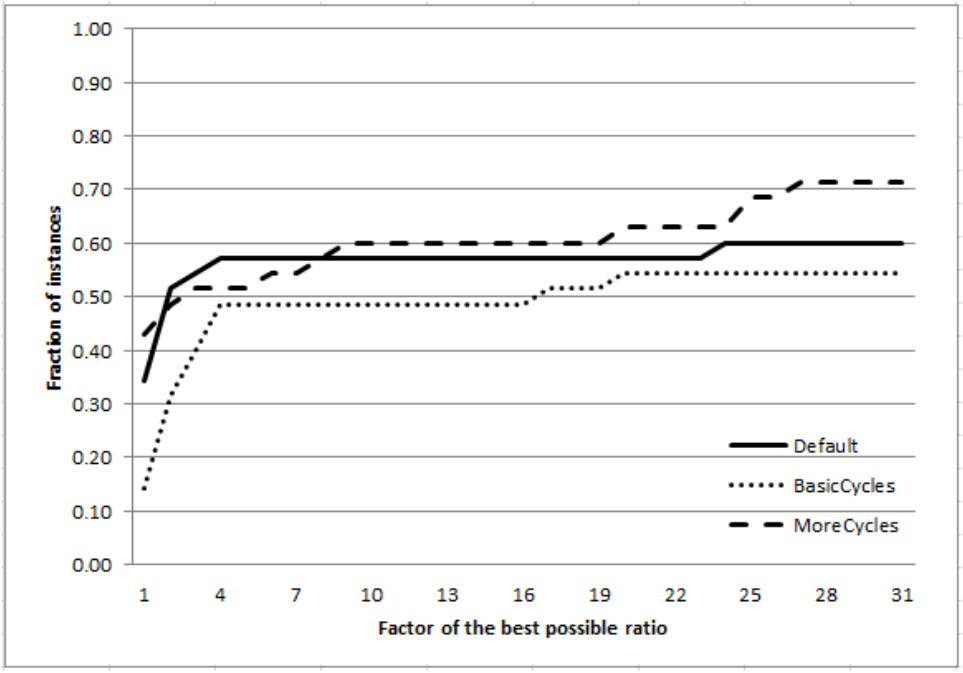


Figure 13: Performance profile for Set 300_5.

Figure 14 shows the performance profiles of the three algorithms over all the five test sets. It shows that, BasicCycles is the fastest algorithm in 38.3% of the instances, whereas Default is the fastest in 29.7% of the instances and MoreCycles is the fastest in 25.7% of the instances. If we look at the algorithm that can solve 75% of all the instances with the highest efficiency, then BasicCycles and MoreCycles have almost identical performance, and both significantly dominate Default. MoreCycles solves slightly more instances than BasicCycles within the time limit. In summary, the performance profiles show that BasicCycles has the highest probability of being the fastest algorithm and MoreCycles solves the most instances. These experiments demonstrate that the cycle inequalities (130) can be quite useful in improving the performance of state-of-the-art MIP software for solving the DC OTS.

Table 20: Summary of all the instances. Initial LP Gap (%): 16.17/10.53

	Default	BasicCycles	MoreCycles
# Cuts	-	26.57/25.10	127.05/101.12
Preprocessing Time (s)	-	0.08/0.05	0.85/0.38
Gap Closed by Cuts (%)	-	3.15/0	4.84/0
Root Gap Closed (%)	6.17/0	18.13/0	24.78/0
Total Time (s)	1278.94/235.82	1023.29/154.57	961.34/174.58
B&B Nodes	1.3E+6/2.1E+5	8.8E+5/1.3E+5	7.9E+5/1.3E+5
# Unsolved	43	33	22
Unsolved Opt Gap (%)	0.56/0.35	0.63/0.34	0.52/0.28

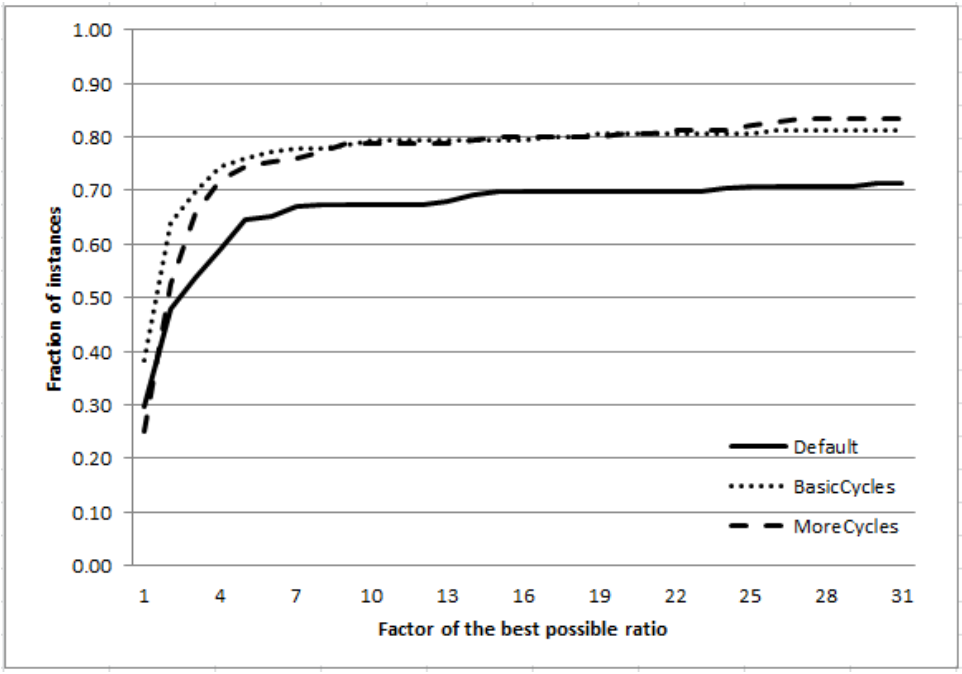


Figure 14: Performance profile for all the instances.

5.6.3 Sensitivity Analysis

Previous literature on the DC OTS demonstrates that significant cost savings can be achieved by switching off only a few lines [33, 107]. However, the optimal solutions obtained by the integer program turn off a significantly larger number of lines than suggested by previous studies. Specifically, the average number of lines turned off in the optimal solutions to the 118-bus instances is 42, and the maximum number turned off is 57. For the 300-bus instances, an average of 85 lines are turned off in the optimal solutions, with a maximum of 107. This surprising result is a consequence of our observation that there are *many* optimal or near-optimal topologies for the DC OTS. To demonstrate the impact of switching off fewer lines than suggested by the optimal solution to the integer program, we performed a sensitivity analysis of the cost versus the number of lines N that are allowed to be switched off.

In this analysis, we chose one instance from each of the five sets whose optimal solution had a large number of lines switched off (41, 38, 41, 48, 91, respectively -

refered to as instances (a), (b), (c), (d), (e) henceforth) and solved a number of DC OTS instances with the cardinality constraint:

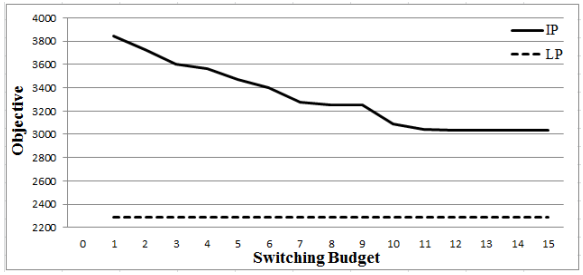
$$\sum_{(i,j)\in\mathcal{L}} x_{ij} \geq |\mathcal{L}| - N \tag{149}$$

added to the formulation. Here, N is the *switching budget*, that is, the number of lines allowed to be switched off (note that $N = 0$ corresponds to DC OPF). We experimented with different N values and the results are given in Figures 15(a)-15(e). We make the following observations:

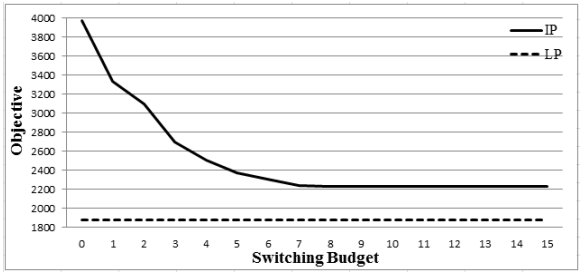
- DC OPF versions of instances (a), (c), (d) and (e) are infeasible.
- Once a particular instance becomes feasible, increasing the switching budget has a significant effect on the objective value for the first few lines (especially, for instances (b), (d) and (e)).
- Nevertheless, the full cost benefit can only be realized by switching off several lines.
- Switching off 11 lines is enough for 118-bus instances (a), (b), (c), (d) to achieve the maximum cost benefit (just seven lines are needed for instance (b)).
- For the 300-bus instance, switching off 15 lines yields nearly the maximum cost benefit.
- The LP relaxation value is not affected by the switching budget.

Our results support the observation that most, although not all, of the cost benefits in transmission switching can be realized by switching off only a few lines. This has a positive impact on the robustness of the network. In our experience, the MIP instances with cardinality constraints were more time consuming to solve that without the cardinality constraint. For example the two instances which had the largest

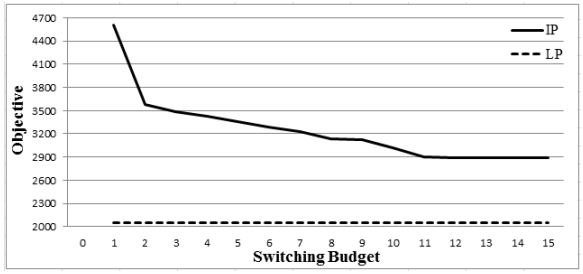
number of switched off lines in the optimal solutions we found (the 118-bus instance with 57 lines switched off and the 300-bus instance with 107 lines switched off) could not be solved in one hour for most switching budgets. Our observation that the LP relaxation value is not affected by the switching budget may help explain this. Since the optimal IP value is larger for smaller switching budgets, the LP relaxation gap is larger for these instances.



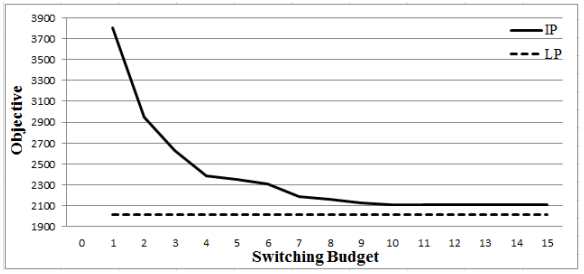
(a) An instance from Set 118_15.



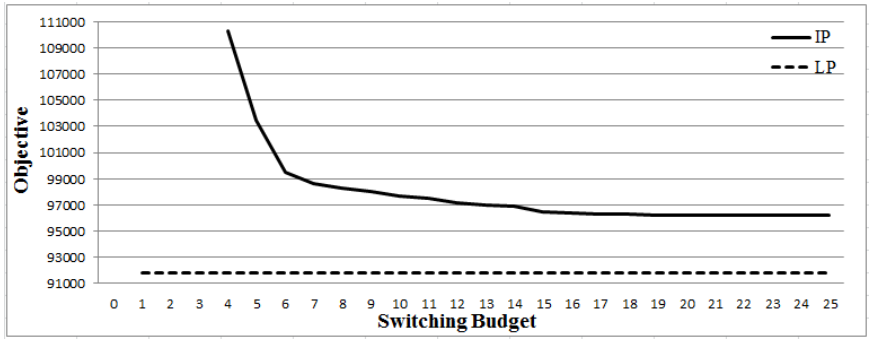
(b) An instance from Set 118_9G.



(c) An instance from Set 118_15_6.



(d) An instance from Set 118_15_16.



(e) An instance from Set 300_5.

Figure 15: Evolution of objective function (IP) and linear programming relaxation (LP) with respect to different switching budgets for five instances.

5.7 Conclusions

In this chapter, we propose new cycle-based formulations for the optimal power flow problem and the optimal transmission switching problems that use the DC approximation to the power flow equations. We characterize the convex hull of a cycle-induced substructure in the new formulation, which provides strong valid inequalities that we may add to improve the new formulation. We demonstrate that separating the new inequalities may be done in linear time for a fixed cycle. We conduct extensive experiments to show that the valid inequalities are very useful in reducing the size of the search tree and the computation time for the DC optimal transmission switching problem.

The inequalities we derive may be gainfully employed for *any* power systems problem that involves the addition or removal of transmission lines and for which the DC approximation to power flow is sufficient for engineering purposes. We will pursue the application of these inequalities to other important power systems planning and operations problems as a future line of research. Other future lines of research include the investigation of more complicated substructure of the new formulation, and engineering the cutting plane procedure to effectively solve larger-scale networks. As an example of studying more complicated substructures, stronger relaxations could be obtained by separating cutting-planes using an extended formulation similar to (132) that includes flow balance constraints.

CHAPTER VI

STRONG MISOCP RELAXATIONS FOR THE AC OPTIMAL TRANSMISSION SWITCHING PROBLEM

6.1 Introduction

In Chapter 5, we used the DC approximation of the power flow model to avoid the mathematical complexity induced by the non-convexity of AC power flow equations and proposed a class of valid inequalities to exactly solve the MILP problem. However, even if this problem can be solved quickly, it has been recognized that the optimal topology obtained by solving DC transmission switching is not guaranteed to be AC feasible, also it may over-estimate cost improvements and overlook stability issues [45]. In this chapter, we aim to push the control scheme for transmission switching closer to the real-world power system operation by proposing a new exact formulation and an efficient algorithm for the AC OTS problem.

In Chapter 3, we proposed several strong second-order cone programming relaxations for AC OPF, which produce extremely high quality feasible AC solutions (not dominated by the SDP relaxations) in a time that is an order of magnitude faster than solving the SDP relaxations. In this chapter, we extend these new techniques to the more challenging AC OTS problem. In particular, we first formulate the AC OTS problem as an MINLP problem. Then, we propose a mixed-integer second-order cone programming relaxation, which relaxes the non-convex AC power flow constraints to a set of convex quadratic constraints, represented in the form of SOCP constraints. The chapter then provides several techniques to strengthen this MISOCP relaxation by adding several types of valid inequalities. Some of these valid inequalities have demonstrated to have excellent performance for the AC OPF in Chapter 3, and some

others are specifically developed for the AC OTS problem. Finally, we also propose practical algorithms that utilize the solutions from the MISOCP relaxation to obtain high quality feasible solutions for the AC OTS problem.

The rest of this chapter is organized as follows: In Section 6.2, we present AC OTS as an MINLP problem and discuss its MISOCP relaxation. Then, we propose several valid inequalities in Section 6.3 and develop a practical algorithm to solve AC OTS in Section 6.4. We present the results of our extensive computational experiments in Section 6.5. Finally, some concluding remarks are given in Section 6.6.

6.2 AC OTS Formulation Revisited

In Section 1.2.2, we introduced the formulation of the AC OTS problem as an MINLP. In this section, we propose an Alternative Formulation of AC OTS based on the Alternative Formulation of the AC OPF problem introduced in Section 2.2.2.

6.2.1 Alternative Formulation

Mathematical programming formulation of OTS can be stated with the same variables as used in OPF with the addition of a set of binary variables, denoted by x_{ij}, for each line. The variable x_{ij} takes the value one if the corresponding line (i, j) is switched on, and zero otherwise. Then, OTS is formulated as the following MINLP problem:

$$\min \sum_{i \in \mathcal{G}} C_i(p_i^g) \tag{150a}$$

$$\text{s.t. } p_{ij} = (-G_{ij}c_{ii} + G_{ij}c_{ij} - B_{ij}s_{ij})x_{ij} \qquad (i,j) \in \mathcal{L} \tag{150b}$$

$$q_{ij} = \quad (B_{ij}c_{ii} - B_{ij}c_{ij} - G_{ij}s_{ij})x_{ij} \qquad (i,j) \in \mathcal{L} \tag{150c}$$

$$(c_{ij}^2 + s_{ij}^2 - c_{ii}c_{jj})x_{ij} = 0 \qquad (i,j) \in \mathcal{L} \tag{150d}$$

$$(\theta_j - \theta_i - \text{atan2}(s_{ij}, c_{ij}))x_{ij} = 0 \qquad (i,j) \in \mathcal{L} \tag{150e}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in \mathcal{L}, \tag{150f}$$

$$\text{(1g)-(1i)}, \text{(10b)-(10c)}, \text{(10d)-(10e)}.$$

Here, constraints (150b) and (150c) guarantee that real and reactive flow on every line takes the associated values if the line is switched on and zero otherwise. Similarly, constraints (150d) and (150e) are active only when the corresponding binary variable takes the value one.

We also note that the model (150) can be appropriately modified to include circuit breakers between bus bars [101].

6.2.2 MISOCP Relaxation of Alternative Formulation

Now, we propose an MISOCP relaxation of OTS (150). For notational convenience, let $\underline{c}_{ii} = \underline{V}_i^2$ and $\overline{c}_{ii} = \overline{V}_i^2$. Here, we extend the definition of variables c_{ij} and s_{ij}, which now take the values as before when the corresponding line is switched on and zero otherwise. We also denote lower and upper bounds of c_{ij} (resp. s_{ij}) as $\underline{c}_{ij}$ (resp. $\underline{s}_{ij}$) and $\overline{c}_{ij}$ (resp. $\overline{s}_{ij}$), respectively, when the line is switched on. Next, we define new variables $c_{ii}^j := c_{ii}x_{ij}$. Using this notation, we present an MISOCP relaxation as follows:

$$\min \ \sum_{i\in\mathcal{G}} C_i(p_i^g) \tag{151a}$$

$$\text{s.t. } p_{ij} = -G_{ij}c_{ii}^j + G_{ij}c_{ij} - B_{ij}s_{ij} \qquad (i,j)\in\mathcal{L} \tag{151b}$$

$$q_{ij} = \ \ B_{ij}c_{ii}^j - B_{ij}c_{ij} - G_{ij}s_{ij} \qquad (i,j)\in\mathcal{L} \tag{151c}$$

$$\underline{c}_{ij}x_{ij} \le c_{ij} \le \overline{c}_{ij}x_{ij} \qquad (i,j)\in\mathcal{L} \tag{151d}$$

$$\underline{s}_{ij}x_{ij} \le s_{ij} \le \overline{s}_{ij}x_{ij} \qquad (i,j)\in\mathcal{L} \tag{151e}$$

$$\underline{c}_{ii}x_{ij} \le c_{ii}^j \le \overline{c}_{ii}x_{ij} \qquad (i,j)\in\mathcal{L} \tag{151f}$$

$$c_{ii} - \overline{c}_{ii}(1-x_{ij}) \le c_{ii}^j \qquad (i,j)\in\mathcal{L} \tag{151g}$$

$$c_{ii}^j \le c_{ii} - \underline{c}_{ii}(1-x_{ij}) \qquad (i,j)\in\mathcal{L} \tag{151h}$$

$$c_{ij}^2 + s_{ij}^2 \le c_{ii}^j c_{jj}^i \qquad (i,j)\in\mathcal{L}, \tag{151i}$$

$$\text{(1g)-(1i)}, \text{(10b)-(10c)}, \text{(10d)-(10e)}, \text{(150f)}.$$

Here, constraints (151b) and (151c) again guarantee that flow variables takes the correct value when the line is switched on and zero otherwise, due to constraints (151d)-(151f). On the other hand, (151g)-(151h) restrict that c_{ii}^j takes value c_{ii} when line in switched on. We note that constraints (151f)-(151h) are precisely the McCormicks envelopes [77] applied to $c_{ii}^j = c_{ii}x_{ij}$. Finally, (151i) is the SOCP relaxation of (150d).

We note that the non-convex constraint (150e) is dropped altogether to obtain the MISOCP relaxation (151). In the next section, we propose three ways to incorporate the constraint (150e) back into the MISOCP relaxation.

6.3 Valid Inequalities

In this section, we propose three methods to strengthen the MISOCP relaxation (151). They are based on the strengthening methods we proposed for the SOCP relaxation of the AC OPF problem in Section 3.3, which are combined with integer programming techniques. In Section 6.3.1, we construct a polyhedral envelope for the arctangent constraint (150e) in 3-dimension. In Section 6.3.2, we propose a disjunctive cut generation scheme that separates a given SOCP solution from the SDP cones. In Section 6.3.3, we propose another disjunctive cut generation scheme that separates a given SOCP solution from the cycle based McCormick relaxation of the OPF problem. Finally, in Section 6.3.4, we propose variable bounding techniques that provide tight variable bounds, which is essential for the success of the proposed approach.

6.3.1 Arctangent Envelopes

First, we propose a convex outer-approximation of the angle condition (150e) to the MISOCP relaxation. Our construction uses four linear inequalities to approximate the convex envelope for the following set defined by the arctangent constraint (150e).

Let us first focus on the upper envelopes. Proposition 29 is adapted from Proposition 10 to the case of OTS:

Proposition 29. *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 $\{z^1, z^2, z^3\}$, and $\{z^1, z^3, z^4\}$, respectively. Then, for $k = 1, 2$, we have*

$$\gamma'_k + \alpha_k c + \beta_k s + (2\pi - \gamma'_k)(1 - x) \geq \arctan\left(\frac{s}{c}\right) \tag{152}$$

for all $(c, s) \in [\underline{c}, \overline{c}] \times [\underline{s}, \overline{s}]$ with $\gamma'_k = \gamma_k + \Delta\gamma_k$, where $\Delta\gamma_k$ is calculated as in (63).

A similar argument can be used to construct lower envelopes as well. See Section 3.3.2 for details.

6.3.2 SDP Disjunction

In the second method to strengthen the MISOCP relaxation (151), we propose a cutting plane approach to separate a given SOCP relaxation solution from the feasible region of the SDP relaxation of cycles. To start with, let us consider a cycle with the set of lines C and the set of buses $\mathcal{B}_C$. Let $v \in \mathbb{R}^{2|C|}$ be a vector of bus voltages defined as $v = [e; f]$ such that $v_i = e_i$ for $i \in \mathcal{B}$ and $v_{i'} = f_i$ for $i' = i + |C|$. Observe that if we have a set of c, s variables satisfying the definitions in (8) and a matrix variable $W = vv^T$, then the following relationship holds between c, s, x and W,

$$c_{ij} = (W_{ij} + W_{i'j'})x_{ij} \qquad (i,j) \in C \tag{153a}$$

$$s_{ij} = (W_{ij'} - W_{ji'})x_{ij} \qquad (i,j) \in C \tag{153b}$$

$$c_{ii} = W_{ii} + W_{i'i'} \qquad i \in \mathcal{B}_C \tag{153c}$$

$$\underline{c}_{ij} x_{ij} \leq c_{ij} \leq \overline{c}_{ij} x_{ij} \qquad (i,j) \in C \tag{153d}$$

$$\underline{s}_{ij} x_{ij} \leq s_{ij} \leq \overline{s}_{ij} x_{ij} \qquad (i,j) \in C \tag{153e}$$

$$\underline{c}_{ii} \leq c_{ii} \leq \overline{c}_{ii} \qquad i \in \mathcal{B}_C \tag{153f}$$

$$c^j_{ii} = c_{ii} x_{ij} \qquad (i,j) \in C \tag{153g}$$

$$x_{ij} \in \{0, 1\} \qquad (i,j) \in C \tag{153h}$$

$$W \succeq 0. \tag{153i}$$

Let us define $\mathcal{S} := \{(c, s, x) : \exists W : (153)\}$. Clearly, any feasible solution to the OTS formulation (150) must also satisfy (153). Therefore, any valid inequality for $\mathcal{S}$ is also valid for the formulation (150).

Note that $\mathcal{S}$ is a mixed-integer set. Ideally, one would be interested in finding $\text{conv}(\mathcal{S})$ to generate strong valid inequalities. However, this is a quite computationally challenging task, no easier than solving the original MINLP. Instead, we outer-approximate $\text{conv}(\mathcal{S})$ and obtain cutting planes by utilizing a simple disjunction for a cycle C: Either every line is active, that is $\sum_{(i,j)\in C} x_{ij} = |C|$, or at least one line is disconnected, that is $\sum_{(i,j)\in C} x_{ij} \leq |C| - 1$. Below, we approximate these two disjunctions.

Disjunction 1: In the first disjunction, we have $x_{ij} = 1$ for all $(i, j) \in C$. Let us consider the following constraints

$$c_{ij} = W_{ij} + W_{i'j'} \qquad (i,j) \in C \tag{154a}$$

$$s_{ij} = W_{ij'} - W_{ji'} \qquad (i,j) \in C \tag{154b}$$

$$c_{ii} = c^j_{ii} \qquad (i,j) \in C \tag{154c}$$

$$x_{ij} = 1 \qquad (i,j) \in C, \tag{154d}$$

and define $\mathcal{S}_1 := \{(c, s, x) : \exists W : (154), (153c) - (153f), (153i)\}$.

Disjunction 0: In the second disjunction, $x_{ij} = 0$ for some $(i, j) \in C$. Let us consider the following constraints

$$c^2_{ij} + s^2_{ij} \leq c^j_{ii} c^i_{jj} \qquad (i,j) \in C \tag{155a}$$

$$\underline{c}_{ii} x_{ij} \leq c^j_{ii} \leq \overline{c}_{ii} x_{ij} \qquad (i,j) \in C \tag{155b}$$

$$c_{ii} - \overline{c}_{ii}(1 - x_{ij}) \leq c^j_{ii} \qquad (i,j) \in C \tag{155c}$$

$$c^j_{ii} \leq c_{ii} - \underline{c}_{ii}(1 - x_{ij}) \qquad (i,j) \in C \tag{155d}$$

$$0 \leq x_{ij} \leq 1 \qquad (i,j) \in C \tag{155e}$$

$$\sum_{(i,j)\in C} x_{ij} \leq |C| - 1, \tag{155f}$$

and define $\mathcal{S}_0 := \{(c, s, x) : (155), (153d)\text{-}(153f)\}$.

We note that both $\mathcal{S}_1$ and $\mathcal{S}_0$ are conic representable. In particular, these bounded sets are respectively semidefinite and second-order cone representable. Therefore, $\text{conv}(\mathcal{S}_1 \cup \mathcal{S}_0)$ is also conic representable (see Appendix C.1 on how to obtain a representation as an extended formulation), and by construction, contains $\mathcal{S}$.

Now, suppose a point (c^*, s^*, x^*) is given. We want to decide whether this point belongs to $\text{conv}(\mathcal{S}_1 \cup \mathcal{S}_0)$ or otherwise, find a separating hyperplane. Given that we have an extended semidefinite representation for $\text{conv}(\mathcal{S}_1 \cup \mathcal{S}_0)$, we can solve an SDP separation problem to achieve this. See Appendix C.2 for details.

6.3.3 McCormick Disjunction

The last method to strengthen the MISOCP relaxation (151) is based on the cycle-based relaxation proposed in Section 3.3.1. The key observation is as follows: instead of satisfying the angle condition (150e) for each $(i, j) \in \mathcal{L}$, we guarantee that angle differences sum up to 0 modulo 2π over every cycle C in the power network if all the lines of the cycle C are switched on, i.e.

$$\Big(\sum_{(i,j)\in C} \theta_{ij} - 2\pi k\Big) \prod_{(i,j)\in C} x_{ij} = 0, \qquad \text{for some } k \in \mathbb{Z}, \tag{156}$$

where $\theta_{ij} := \theta_j - \theta_i$.

Next, we consider

$$\Big[\cos\Big(\sum_{(i,j)\in C} \theta_{ij}\Big) - 1\Big] \prod_{(i,j)\in C} x_{ij} = 0 \tag{157a}$$

$$c_{ij} = \sqrt{c_{ii}c_{jj}} \cos\theta_{ij} x_{ij} \qquad (i, j) \in C \tag{157b}$$

$$s_{ij} = \sqrt{c_{ii}c_{jj}} \sin\theta_{ij} x_{ij} \qquad (i, j) \in C, \tag{157c}$$

$$(153d) - (153h).$$

Here, (157a) is equivalent to (156) and (157b)-(157c) follow from the definition of c, s variables. Let us define $\mathcal{M} := \{(c, s, x) : \exists\theta : (157), (153d) - (153h)\}$. Again,

observe that any feasible solution to the OTS formulation (150) must also satisfy (157). Therefore, any valid inequality for $\mathcal{M}$ is also valid for the formulation (150).

We again follow a similar procedure to the previous section and consider two disjunctions for a cycle C.

Disjunction 1: In the first disjunction, we have $x_{ij} = 1$ for all $(i,j) \in C$. Note that (157a) reduces to

$$\cos\Big(\sum_{(i,j)\in C} \theta_{ij}\Big) = 1. \tag{158}$$

Now, we can apply the proposed decomposition scheme in Section 3.3.1.3 to obtain a set of bilinear equations $q_k(c, s, \tilde{c}, \tilde{s}) = 0,\ k \in \mathcal{K}_C$, for a given cycle C, which ensure that phase angle differences sum up to 0 modulo 2π (recall that the additional $\tilde{c}, \tilde{s}$ variables are introduced in the cycle decomposition and represent the values for the artificial lines in the triangulation). Finally, we use McCormick envelopes for each bilinear constraint to linearize the system of polynomials as in Section 3.3.1.4 to obtain an extended polyhedral relaxation written compactly in (58). Let us define the set $\mathcal{M}_1 := \{(c, s, x) : \exists(\tilde{c}, \tilde{s}) : (58), (153\text{d})\text{-}(153\text{f}), (154\text{c})\text{-}(154\text{d})\}$.

Disjunction 0: In the second disjunction, $x_{ij} = 0$ for some $(i,j) \in C$. We take $\mathcal{M}_0 := \mathcal{S}_0$.

We note that both $\mathcal{M}_1$ and $\mathcal{M}_0$ are conic representable. In particular, these bounded sets are respectively polyhedral and second-order cone representable. Therefore, $\text{conv}(\mathcal{M}_1 \cup \mathcal{M}_0)$ is also conic representable, and by construction, contains $\mathcal{M}$.

Now, suppose a point (c^*, s^*, x^*) is given. We want to decide whether this point belongs to $\text{conv}(\mathcal{M}_1 \cup \mathcal{M}_0)$ or otherwise, find a separating hyperplane. Given that we have an extended second-order cone representation for $\text{conv}(\mathcal{M}_1 \cup \mathcal{M}_0)$, we can solve an SOCP separation problem.

In our computations, we observed that stronger cuts are obtained by combining SDP and McCormick Disjunction. In particular, we separate cutting planes from $\text{conv}((\mathcal{S}_1 \cap \mathcal{M}_1) \cup \mathcal{S}_0)$ by solving SDP separation problems.

6.3.4 Obtaining Variable Bounds

Note that the arctangent envelopes and the McCormick relaxations are more effective when tight variable upper/lower bounds are available for the c and s variables. Now, we explain how we obtain good bounds for these variables, which is the key ingredient in the success of our proposed methods.

Recall that c_{ij} and s_{ij} do not have explicit variable bounds except the implied bounds in (24). However, these bounds may be quite loose, especially when the phase angle differences are small, implying $c_{ij} \approx 1$ and $s_{ij} \approx 0$ when the corresponding line is switched on. Therefore, one should try to improve these bounds.

We adapt the procedure proposed in Section 3.3.4 (which dealt only with OPF) to the case of OTS in order to obtain variable bounds, that is, we solve a reduced version of the full MISOCP relaxation to efficiently compute bounds. In particular, consider (159), the continuous relaxation of MISOCP relaxation applied to the part of the power network within r steps of the buses k and l. c_{kl} and s_{kl} can be minimized and maximized subject to (159) for each edge (k, l) to obtain lower and upper bounds, respectively. These SOCPs can be solved in parallel, since they are independent of each other. Recall that a good trade-off between accuracy and speed is to select $r = 2$ in Chapter 3. Constraint (159n) may seem to restrict the feasible region, however, the way we defined c_{kl} and s_{kl} variables, they are the values for cosine and sine components when $x_{kl} = 1$ (otherwise, they are 0). Therefore, it is enough for the bounds to be valid for $x_{kl} = 1$ only.

$$p_i^g - p_i^d = g_{ii}c_{ii} + \sum_{j\in\delta(i)} p_{ij} \qquad i \in \mathcal{B}_{kl}(r) \tag{159a}$$

$$q_i^g - q_i^d = -b_{ii}c_{ii} + \sum_{j\in\delta(i)} q_{ij} \qquad i \in \mathcal{B}_{kl}(r) \tag{159b}$$

$$p_{ij} = -G_{ij}c_{ii}^j + G_{ij}c_{ij} - B_{ij}s_{ij} \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159c}$$

$$q_{ij} = \quad B_{ij}c_{ii}^j - B_{ij}c_{ij} - G_{ij}s_{ij} \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159d}$$

$$p_{ij}^2 + q_{ij}^2 \le (S_{ij}^{\max})^2 \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159e}$$

$$\underline{c}_{ij}x_{ij} \le c_{ij} \le \overline{c}_{ij}x_{ij} \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159f}$$

$$\underline{s}_{ij}x_{ij} \le s_{ij} \le \overline{s}_{ij}x_{ij} \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159g}$$

$$\underline{c}_{ii}x_{ij} \le c_{ii}^j \le \overline{c}_{ii}x_{ij} \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159h}$$

$$c_{ii} - \overline{c}_{ii}(1 - x_{ij}) \le c_{ii}^j \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159i}$$

$$c_{ii}^j \le c_{ii} - \underline{c}_{ii}(1 - x_{ij}) \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159j}$$

$$c_{ij} = c_{ji}, \quad s_{ij} = -s_{ji} \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159k}$$

$$c_{ij}^2 + s_{ij}^2 \le c_{ii}^j c_{jj}^i \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159l}$$

$$0 \le x_{ij} \le 1 \qquad (i,j) \in \mathcal{L}_{kl}(r) \tag{159m}$$

$$x_{kl} = 1, \tag{159n}$$

$$\text{(69c)-(69e)}. \tag{159o}$$

Bounds on an artificial edge (i,j) used in the construction of McCormick envelopes are chosen as follows:

$$\overline{c}_{ij} = -\underline{c}_{ij} = \overline{s}_{ij} = -\underline{s}_{ij} = \overline{V}_i\overline{V}_j. \tag{160}$$

A similar idea can be used to fix some of the binary variables as well. In particular, we can minimize x_{kl} over (69) after omitting (159n). If the optimal value turns out to be strictly positive, then x_{kl} can be fixed to one.

6.4 Algorithm

In this section, we propose an algorithm to solve OTS. The algorithm has two phases. The first phase involves solving a sequence of SOCPs obtained by relaxing integrality restriction of the binary variables in MISOCP (151), and incorporates cycle inequalities generated from the extended SDP and McCormick relaxations in Section 6.3.2 and 6.3.3. In this phase, the aim is to strengthen the lower bound on the MISOCP relaxation. The second phase involves solving a sequence of MISOCP relaxations strengthened by cycle inequalities. The aim in this phase is to obtain high quality feasible solutions for OTS. In particular, this is achieved by solving OPF subproblems with fixed topologies obtained from the integral solutions found during the branch-and-cut process of solving the MISOCP (151). This procedure is repeated by "forbidding" the topologies already considered in order to obtain different network configurations in the subsequent iterations.

Now we formally define the ingredients of the algorithm. First, let $SOCP(\mathcal{V})$ be the continuous relaxation of MISOCP (151) with a set of valid inequalities $\mathcal{V}$ obtained from cycle inequalities using extended SDP and McCormick relaxations. The set $\mathcal{V}$ is dynamically updated T_1 times. Similarly, we define $MISOCP(\mathcal{V}, \mathcal{F})$ as the MISOCP relaxation of OTS with a set of valid cycle inequalities $\mathcal{V}$ and forbidden topologies $\mathcal{F}$. Here, we forbid a topology $x^* \in \mathcal{F}$ by adding the following "no-good" cut (see [4] for generalizations) to the formulation:

$$\sum_{(i,j):x^*_{ij}=1} (1 - x_{ij}) + \sum_{(i,j):x^*_{ij}=0} x_{ij} \geq 1. \tag{161}$$

We denote by LB_t as the optimal value of $MISOCP(\mathcal{V}, \mathcal{F})$ and $\mathcal{P}_t$ as the set of all integral solutions found by the solver at the t-th iteration. For instance, CPLEX offers this option called *solution pool*. In a practical implementation, this part is repeated T_2 times.

Let $OPF(x)$ denote the value of a feasible solution to OPF problem (10) for the

fixed topology induced by the integral vector x. Finally, UB is the best upper bound on OTS. Now, we present Algorithm 11.

Algorithm 11 OTS algorithm.

Input: T_1, T_2, ϵ.
Phase I: Set $\mathcal{V} \leftarrow \emptyset$, $\mathcal{F} \leftarrow \emptyset$, $UB \leftarrow \infty$.
for $\tau = 1, \ldots, T_1$ **do**
  Solve $SOCP(\mathcal{V})$.
  Separate cycle inequalities for each cycle in a cycle basis to obtain a set of valid inequalities $\mathcal{V}_\tau$.
  Update $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{V}_\tau$.
end for
Phase II: Set $t \leftarrow 0$.
repeat
  $t \leftarrow t + 1$
  Solve $MISOCP(\mathcal{V}, \mathcal{F})$ to obtain a pool of integral solutions $\mathcal{P}_t$ and record the optimal cost as LB_t.
  for all $x \in \mathcal{P}_t$ **do**
    if $OPF(x) < UB$ **then**
      $UB \leftarrow OPF(x)$
    end if
  end for
  Update $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{P}_t$.
until $LB_t \geq (1 - \epsilon)UB$ or $t \geq T_2$

Observation 1. *If $OPF(x)$ returns globally optimal solution for every topology x, $\epsilon = 0$ and $T_2 = \infty$, then Algorithm 11 converges to the optimal solution of OTS in finitely many iterations.*

Observation 1 follows from the fact that there are finitely many topologies and by the hypothesis that $OPF(x)$ can be solved globally, which is possible for some IEEE instances using moment/sum-of-squares relaxations [55]. Although Observation 1 states that Algorithm 11 can be used to solve OTS to global optimality in finitely many iterations, the requirement of solving $OPF(x)$ to global optimality may not be satisfied always. In practice, we can solve OPF subproblems using local solver, in which case we have Observation 2.

Observation 2. *If $OPF(x)$ is solved by a local solution method, then we have $LB_1 \leq z^* \leq UB$ upon termination of Algorithm 11, where z^* is the optimal value of OTS.*

6.5 Computational Experiments

In this section, we present the results of our extensive computational experiments on standard IEEE instances available from MATPOWER [111] and instances from NESTA 0.3.0 archive with congested operating conditions [25]. The code is written in the C# language with Visual Studio 2010 as the compiler. 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. We use three different solvers:

- MIP solver CPLEX 12.6 [3] to solve MISOCPs.
- Conic interior point solver MOSEK 7.1 [2] to solve SDP separation problems.
- Nonlinear interior point solver IPOPT [105] to find local optimal solutions to $OPF(x)$.

We use a Gaussian elimination based approach to construct a cycle basis given in Appendix B and use this set of cycles in the separation phase.

6.5.1 Methods

We report the results of three algorithmic settings:

- SOCP: MISOCP formulation (151) in Phase II without Phase I (i.e. $T_1 = 0$).
- SOCPA: SOCP strengthened by the arctangent envelopes introduced in Section 6.3.1.
- SOCPA_Disj: SOCPA strengthened further by dynamically generating linear valid inequalities obtained from separating an SOCP feasible solution from the SDP and McCormick relaxation over cycles using a disjunctive argument T_1

times. In particular, a separation oracle is used to separate a given point from $\text{conv}((\mathcal{S}_1 \cap \mathcal{M}_1) \cup \mathcal{S}_0)$.

The following four performance measures are used to assess the accuracy and the efficiency of the proposed methods:

- "%OG" is the percentage optimality gap proven by our algorithm calculated as $100 \times (1 - \widetilde{LB}_1/UB)$. Here, $\widetilde{LB}_1$ is the lower bound proven, which may be strictly smaller than LB_1 due to optimality gap tolerance and time limit.

- "%CB" is the percentage cost benefit obtained by line switching calculated as $100 \times (1 - UB/OPF(e))$, where e is the vector of ones so that $OPF(e)$ corresponds to the OPF solution with the initial topology.

- "#off" is the number of lines switched off in the topology which gives UB.

- "TT" is the total time in seconds, including preprocessing (bound tightening), solution of $T_1 = 5$ rounds of SOCPs to improve lower bound and separation problems to generate cutting planes (in the case of SOCPA_Disj), solution of T_2 rounds of MISOCPs and several calls to local solver IPOPT with given topologies. MISOCPs are solved under 720 seconds time limit so that 5 iterations take about 1 hour (optimality gap for integer programs is 0.01%). Preprocessing and separation subproblems are parallelized.

We choose parameter $T_2 = 5$ and pre-terminate Algorithm 11 if 0.1% optimality gap is proven.

6.5.2 Results

The results of our computational experiments are presented in Tables 21 and 22 for standard IEEE and NESTA instances, respectively. We considered instances up to 300-bus since Phase II of the Algorithm 11 does not scale up well for larger instances.

Let us start with the former: IEEE instances are a relatively easy set since transmission line limits are generally not binding. Therefore, cost benefits obtained by switching are also limited. The largest cost reduction is obtained for case30Q with 2.24%. Among the three methods, the most successful one is SOCPA_Disj, on average proving 0.05% optimality gap and providing 0.31% cost savings. In terms of computational time, SOCP is the fastest, however, its performance is not as good as the other two. Quite interestingly, SOCPA_Disj is faster than SOCPA, on average, for this set of instances. In terms of comparison with other methods, unfortunately, there is limited literature for this purpose. In [45], nine of these instances (except for cases 9Q and 30Q) are considered and a quadratic convex (QC) relaxation based approach is used. On average, their approach proves 0.14% optimality gap, which is worse than any of our methods over the same nine instances. The only instance QC approach is better is 118ieee with 0.11% optimality gap, while it is worse than our methods for case300 with a 0.47% optimality gap.

Table 21: Results summary for standard IEEE Instances.

	SOCP				SOCPA				SOCPA_Disj			
case	%OG	%CB	#off	TT(s)	%OG	%CB	#off	TT(s)	%OG	%CB	#off	TT(s)
6ww	0.16	0.48	2	1.29	0.02	0.48	2	0.67	0.01	0.48	2	1.28
9	0.00	0.00	0	0.26	0.00	0.00	0	0.22	0.00	0.00	0	0.55
9Q	0.04	0.00	0	0.42	0.04	0.00	0	0.33	0.04	0.00	0	0.97
14	0.08	0.00	0	0.66	0.09	0.00	0	0.70	0.01	0.00	1	1.81
ieee30	0.05	0.00	1	1.95	0.05	0.00	0	1.67	0.02	0.00	1	3.84
30	0.07	0.52	1	4.60	0.06	0.52	2	5.01	0.03	0.51	2	9.39
30Q	0.44	2.05	2	24.43	0.43	2.03	5	25.80	0.13	2.24	5	44.16
39	0.03	0.00	0	2.53	0.01	0.02	1	3.17	0.01	0.02	1	4.48
57	0.07	0.02	4	6.18	0.07	0.02	4	8.72	0.08	0.01	1	13.59
118	0.19	0.08	4	3065.64	0.15	0.12	10	2553.59	0.17	0.08	16	3174.01
300	0.16	0.02	9	2318.89	0.15	0.03	12	3624.12	0.10	0.05	15	2803.31
Avg.	**0.12**	**0.29**	**2.1**	**493.35**	**0.10**	**0.29**	**3.3**	**565.82**	**0.05**	**0.31**	**4.0**	**550.67**

Now let us consider NESTA instances with congested operating conditions. This set is particularly suited for line switching as more stringent transmission line limits are imposed. In fact, large cost improvements are observed for some test cases. For instance, about 45% and 39% cost reductions are possible for cases 30fsr and 118ieee, respectively. Other instances with sizable cost reductions include cases 6ww and 30as. SOCPA_Disj is again the most successful method if we look at averages of optimality gap (1.16%) and cost savings (6.21%). It certifies that the best topology is within 1.17% of the optimal for all the cases except for 118ieee and 189edin. In terms of computational time, SOCP is again the fastest, however, its performance is significantly worse than the other two. We also note that SOCPA improves quite a bit over SOCP in terms of optimality and cost benefits with 70% increase in computational time. SOCPA_Disj takes about only 10% more time than SOCPA. As we go from SOCP to SOCPA_Disj, problems get more complicated and sometimes, MISOCPs are not solved to optimality within time limit. Consequently, for cases 189edin and 300ieee, the optimality gaps proven and cost benefits obtained by SOCPA_Disj can be slightly worse.

Finally, we note that that optimality gaps can be explained by two non-convexities: 1) integrality, 2) power flow equations. For instance, in case 3lmbd, the optimality gap can only be explained by the non-convexity of power flow equation since all the relevant topologies are considered. Similarly, at least some portion of the relatively large optimality gaps for cases 118ieee and 189edin may be attributed to non-convexity of power flow equations. Consequently, any future improvements on strengthening the convex relaxations of OPF problem can be useful in closing more gaps in OTS as well.

Table 22: Results summary for NESTA Instances from Congested Operating Conditions.

	SOCP				SOCPA				SOCPA_Disj			
case	%OG	%CB	#off	TT(s)	%OG	%CB	#off	TT(s)	%OG	%CB	#off	TT(s)
3lmbd	3.30	0.00	0	0.14	2.00	0.00	0	0.14	1.17	0.00	0	0.30
4gs	0.65	0.00	0	0.11	0.16	0.00	0	0.13	0.00	0.00	0	0.27
5pjm	0.18	0.27	1	0.61	0.01	0.27	1	0.41	0.02	0.27	1	0.89
6ww	6.06	7.74	1	1.23	1.34	7.74	1	1.64	1.05	7.74	1	1.97
9wscc	0.00	0.00	0	0.19	0.00	0.00	0	0.20	0.00	0.00	0	0.30
14ieee	1.02	0.33	1	2.86	0.89	0.45	2	3.48	0.41	0.45	2	4.49
29edin	0.43	0.00	2	12.79	0.24	0.18	13	299.82	0.33	0.08	21	181.74
30as	1.81	3.13	2	14.82	0.35	3.30	5	19.52	0.34	3.30	5	24.93
30fsr	3.24	44.20	2	9.72	0.05	44.98	2	4.76	0.03	44.98	3	6.97
30ieee	0.54	0.46	1	12.28	0.40	0.48	2	10.61	0.15	0.48	2	13.37
39epri	1.92	1.10	1	11.56	0.80	1.41	2	13.20	0.70	1.52	2	12.65
57ieee	0.12	0.10	3	41.48	0.12	0.10	2	58.97	0.09	0.10	3	29.86
118ieee	41.67	4.33	3	225.57	21.51	27.98	30	3838.62	7.50	39.09	21	3856.76
162ieee	0.57	1.05	9	3675.75	0.63	1.00	15	3861.29	0.60	1.00	15	3855.50
189edin	5.31	1.10	3	540.02	4.81	0.13	2	2194.80	5.58	0.00	0	3634.02
300ieee	1.00	0.10	12	3655.10	0.65	0.37	21	3640.14	0.61	0.35	21	3651.95
Avg.	**4.24**	**3.99**	**2.6**	**512.76**	**2.12**	**5.52**	**6.1**	**871.73**	**1.16**	**6.21**	**6.1**	**954.75**

6.5.3 Discussion

In this section, we take a closer look at some of the instances with large cost benefits and try to gain some insight as to 1) how these large savings are obtained, and 2) how simple heuristics may fail to produce comparable results. Firstly, using a small example, we illustrate how large cost savings can be obtained. Secondly, we compare the results of our algorithm with a commonly used heuristic based on switching the best line and demonstrate how different the solution quality can be.

To address the first issue, let us concentrate on a small instance, namely case6ww from NESTA archive. This instance has the same topology and line characteristics as the standard IEEE test case but load and generation parameters are slightly different. In particular, $p_i^d = 78.24$, $q_i^d = 70$, $\underline{V}_i = 0.95$ and $\overline{V}_i = 1.05$ for the load buses $i = 4, 5, 6$ while the data for generation buses 1, 2 and 3 is summarized in Table 23. With this topology, the local optimal solution obtained using IPOPT with objective

Table 23: Generator data for NESTA case6ww test case.

	$p_i^{\min}$	$p_i^{\max}$	$q_i^{\min}$	$q_i^{\max}$	$\underline{V}_i = \overline{V}_i$	cost
1	25	200	−100	100	1.05	1.276311
2	18.75	106	−100	100	1.05	0.586272
3	22.5	93	−100	100	1.07	1.29111

value of 273.76 is given in Figure 16. We note that the lines $(1, 5)$, $(2, 4)$ and $(3, 6)$ are congested in this configuration. On the other hand, if the line $(1, 2)$ is switched off, then the objective value reduces to 252.57, corresponding to a 7.74% cost saving over the initial topology. The difference is due to the fact that the outputs of generators 1 and 2 are now changed to (85.56, 32.74) and (84.25, 63.26), respectively. Notice that with the new topology, the cheaper generator 2 is used more, which results in the cost reduction. In the initial topology, this is not possible since the voltage magnitudes of the generators are fixed, and lines $(1, 5)$ and $(2, 4)$ are congested.

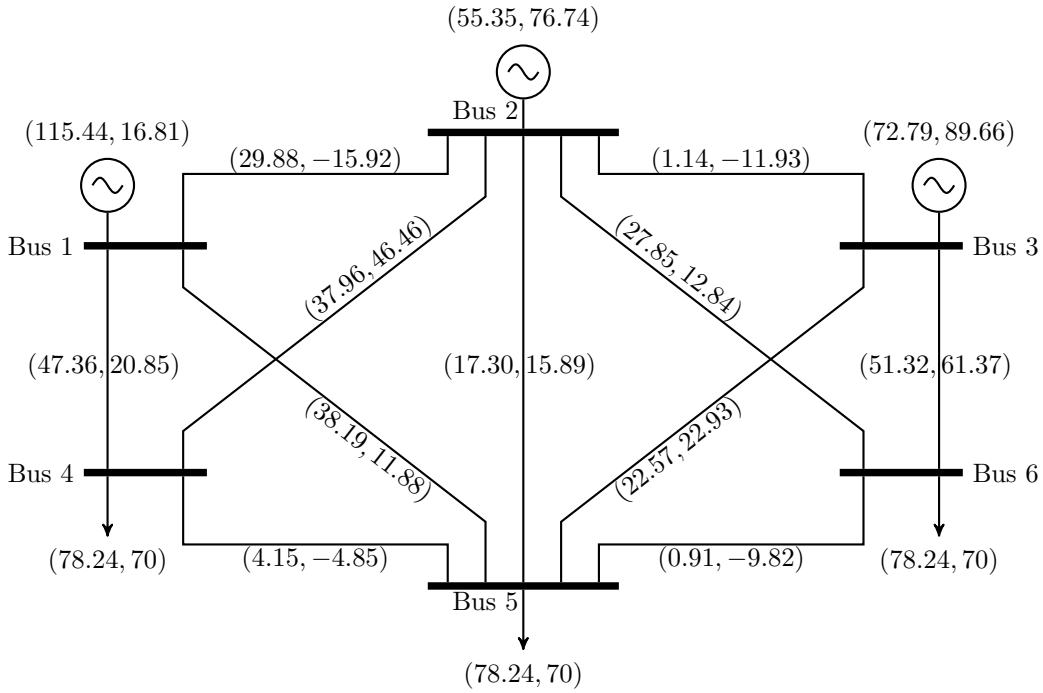


Figure 16: Flow diagram for the solution of NESTA case6ww without any line switching. The numbers above each generator node respectively represent the active and reactive power output. Similarly, the numbers near each edge respectively represent the active and reactive power flow of the line in the direction from the small indexed bus to the large indexed one. The figure is generated by modifying [70].

Now, let us consider the second issue. Due to the combinatorial nature of OTS problem, heuristics are frequently used to obtain suboptimal solutions. A commonly used one is to switch off a single line to obtain cost benefits [90, 64]. Although this heuristic idea is easy to implement and works well in some instances, there are no guarantees on its accuracy. For example, in case6ww, the best line to switch off is, in fact, $(1, 2)$ suggested by both the best line heuristic and our algorithm. However, for other problems with large cost benefits, this is not always the case. For instance, in case30as and case30fsr, the best line heuristic reduces the overall cost to 2.99% and 44.02% respectively, compared to 3.30% and 44.98% obtained from our algorithm. For case118ieee, the cost reduction is dramatically different. The best line heuristic

reduces the cost only by 19.52% while our algorithm provides a topology with 39.09% saving. Moreover, the best line heuristic does not provide any guarantee on how good the solution is while our algorithm gives optimality guarantees by construction. Therefore, adapting Algorithm 11 in real-time operations can yield significant savings over simple heuristics.

6.6 Conclusions

In this chapter, we proposed a systematic approach to solve the AC OTS problem. In particular, we presented an alternative formulation for OTS and constructed a MISOCP relaxation. We improved the strength of this relaxation by the addition of arctangent envelopes and cutting planes obtained using disjunctive techniques. The use of these disjunctive cuts help in closing gap significantly. Our experiments on standard and congested instances suggest that the proposed methods are effective in obtaining strong lower bounds and producing provably good feasible solutions.

We remind the reader that AC OTS is a challenging problem since it embodies two types of non-convexities due to AC power flow constraints and integrality of variables. We hope that the methodology developed in this chapter can eventually be further improved to solve AC OTS problem in real life operations. As a future work, we would like to pursue finding ways to improve the solution time of MISOCPs as this step is the bottleneck in Algorithm 11. Also, decomposition methods can be sought to solve large-scale problems more efficiently, which could make the proposed approach adaptable to real life instances.

APPENDIX A

PROOF OF THEOREM 4

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

Proposition 30. *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 2b: $\frac{a_i y_i}{a_j x_j} < 0$

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 30 implies the following corollary.

Corollary 2. *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 30 implies that

$$(x_i = \hat{x}_i \vee y_j = \hat{y}_j) \wedge (x_i = \hat{x}_i \vee x_j = \hat{x}_j) \wedge (y_i = \hat{y}_i \vee y_j = \hat{y}_j) \wedge (y_i = \hat{y}_i \vee x_j = \hat{x}_j)$$
$$= (x_i = \hat{x}_i \wedge y_i = \hat{y}_i) \vee (x_j = \hat{x}_j \wedge y_j = \hat{y}_j), \tag{162}$$

which is the desired conclusion. □

An immediate consequence of Corollary 2 is the following characterization of extreme points of $\mathcal{S}_a$:

Corollary 3. *All the extreme points of $\mathcal{S}_a$ are in one of the following sets:*

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

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

$$T_\alpha = \{(x,y) \in \mathbb{R}^2 : xy = \alpha,\quad x \in [\underline{x}, \overline{x}],\ y \in [\underline{y}, \overline{y}]\} \tag{163}$$

for some constant α.

Proposition 31. *Set conv(T_α) is second-order cone representable for any value of α.*

There are several cases based on parameter values. In the worst 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 4. 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. □

APPENDIX B

CYCLE BASIS

Consider a directed graph $G = (V, E)$ with vertex set V and arc set A. Let $|V| = n$ and $|E| = m$. We define edge-node incidence matrix A as

$$A_{(i,j),k} = \begin{cases} 1 & \text{if } i = k \\ -1 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \tag{164}$$

Definition 2. *[40] Let $v \in \{0, \pm 1\}^{|E|}$ be an incidence vector for a cycle C in graph $G = (V, E)$, where*

$$v_{ij} = \begin{cases} 1 & \textit{if } (i,j) \textit{ is traversed in the right direction by } C \\ -1 & \textit{if } (i,j) \textit{ is traversed in the opposite direction by } C \\ 0 & \textit{if } (i,j) \textit{ is not in } C. \end{cases}$$

The cycle space of G is the vector space that is spanned by the incidence vectors of its cycles. A set of cycles is called a cycle basis if it forms a basis for this vector space.

A cycle basis of G is then a minimal set of cycles of G with the property that all cycles of G are linear combinations of the cycles in the basis. Assuming that G is connected, Algorithm 12 can be used to find a cycle basis.

Algorithm 12 Cycle basis generation.

Define edge-node incidence matrix A of directed graph G as given in (164).
Carry out LU decomposition of A with partial pivoting to compute $PA = LU$ with a unit lower triangular matrix L.
Last $m - n + 1$ rows of $L^{-1}P$, denoted by C_b, gives a cycle basis.

Proposition 32. *Algorithm 12 works correctly.*

Proof. Without loss of generality, assume that the first $n-1$ rows of A are selected such that no row permutation is necessary during LU decomposition. In the remaining of the proof, we will replace PA with A for brevity.

The LU decomposition of A can be obtained by a sequence of Gaussian eliminations on A as

$$\tilde{A}_1 = \tilde{L}_1 \cdot A, \;\; \tilde{A}_2 = \tilde{L}_2 \cdot \tilde{A}_1, \ldots, \;\; U = \tilde{A}_{n-1} = \tilde{L}_{n-1} \cdot \tilde{A}_{n-1},$$

where each matrix $\tilde{L}_i$ is an elementary row operation that adds or subtracts multiples of the i-th row of $\tilde{A}_{i-1}$ to other rows to make the i-th column of $\tilde{A}_{i-1}$ the i-th unit vector. Consider a nonzero entry a_{i1} of A. Since $a_{11}, a_{i1} \in \{+1, -1\}$, the row operation only adds $+1$ or -1 copy of row 1 to row i, that is, the first column of $\tilde{L}_1$ only contains $0, \pm 1$. Also, after eliminating a_{i1}, row i of A_1 will either be all zero, or contain exactly one 1 and one -1. In other words, A_1 is an arc-node incidence matrix for a new digraph $G_1 = (V, E_1)$. Since $\text{rank}(A) = n-1$, we have $\text{rank}(\tilde{A}_1) = n-1$, which implies the new digraph G_1 is connected. Repeating this argument for each subsequent round of Gaussian elimination, we have that U is an incidence matrix of the connected digraph G_{n-1} with $n-1$ arcs, which implies G_{n-1} is a spanning tree of the node set V. Denote the first $n-1$ rows of U as U_1. The last $m-n+1$ rows of U are zeros.

Denote $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ where A_1 is the first $n-1$ rows of A and represents a spanning tree T in the original graph G. Note that the rows of A_1 are linearly independent. Let us first carry out the LU decomposition of A_1 to get $A_1 = L_1 U_1$. In fact, U_1 represents a spanning tree, say T', on a new graph $G' = (V, E')$. Note that the entries of L_1 are precisely the negative of the pivots in Gaussian elimination and hence, they are ± 1. Moreover, we can interpret the rows of L_1 indexed by the edges in T and columns indexed by the edges in T'. In particular, the elements of row $(i, j) \in T$ represent the unique path in T' going from i to j.

Claim 1. *A path in T can be mapped to a path in T' by post-multiplication of L_1 and this transformation is unique.*

Proof. Let us consider a path p in T as a row vector where +1 (-1) means an arc is traversed in forward (backward) direction and 0 means that arc is not part of the path. Define $p' = pL_1$. We claim that the row vector p' is a path in T'. Let us traverse the path p in terms of the edges in T'. In particular, we weight the rows of $\mathcal{T}$ corresponding to (i, j) with the value of that edge in the path p. In other words, for each arc (i, j) in the path, we traverse the path from i to j in $\mathcal{T}'$. But, this gives a path in $\mathcal{T}'$. Finally, this transformation is unique since the path joining two nodes in a tree is unique. □

Now, consider $A' = \begin{bmatrix} U_1 \\ A_2 \end{bmatrix}$. We continue LU decomposition on A' to obtain $A' = \begin{bmatrix} U_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ L_2 & I \end{bmatrix} \begin{bmatrix} U_1 \\ 0 \end{bmatrix}$. In particular, we have $A_2 = L_2U_1$. Since the rows of U_1 are linearly independent and U_1 defines a tree, the elements of A_2 can be traced via a unique path in T'. In fact, the paths are exactly L_2 in the new network. If the paths in L_2 are traced backwards, we obtain cycles in G'. Hence, $\begin{bmatrix} -L_2 & I \end{bmatrix}$ is a cycle basis in G'.

At this point, we can write $A = LU$ where

$$L = \begin{bmatrix} L_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ L_2 & I \end{bmatrix} \text{ and } L^{-1} = \begin{bmatrix} L_1^{-1} & 0 \\ -L_2L_1^{-1} & I \end{bmatrix}$$

Finally, we claim that $C_b = \begin{bmatrix} -L_2L_1^{-1} & I \end{bmatrix}$ is a cycle basis in G. Let us first focus on the system $L_2 = ML_1$. Recall that the rows of L_2 are paths in G'. We claim that the rows of M are the corresponding paths in G. Using Claim 1, we know that post-multiplication of a path in G by L_1 gives a path in G'. But, since L_1 is invertible, $M = L_2L_1^{-1}$ is the unique solution and therefore, the rows of M should represent

paths in G. Then, by tracing the paths in M backwards, we obtain cycles in G. Therefore, $\begin{bmatrix} -L_2 L_1^{-1} & I \end{bmatrix}$ is a cycle basis in G. □

Note that we do not need to explicitly invert L to obtain L^{-1}. In fact, LU decomposition produces L^{-1}. Hence, it is computationally efficient to find cycle basis using Algorithm 12.

APPENDIX C

SOME OPERATIONS WITH CONIC REPRESENTABLE SETS

C.1 Convex Hull of Union of Two Conic Representable Sets

Let S_1 and S_2 be two bounded, conic representable sets

$$S_i = \{x : \exists u^i : A_i x + B_i u^i \succeq_{K_i} b_i\} \quad i = 1, 2. \tag{165}$$

Here, K_i's are regular (closed, convex, pointed with non-empty interior) cones. Then, a conic representation for $\text{conv}(S_1 \cup S_2)$ is given as follows [12]:

$$x = x^1 + x^2,\ \lambda_1 + \lambda_2 = 1,\ \lambda_1, \lambda_2 \geq 0 \tag{166a}$$

$$A_i x^i + B_i u^i \succeq_{K_i} b_i \lambda_i \quad i = 1, 2. \tag{166b}$$

C.2 Separation from an Extended Conic Representable Set

Let S be a conic representable set $S = \{x : \exists u : Ax + Bu \succeq_K b\}$. Here, K is a regular cone. Suppose we want to decide if a given point x^* belongs to S and find a separating hyperplane $\alpha^T x \geq \beta$ if $x^* \notin S$. This problem can be formulated as $\max_{\alpha,\beta} \left\{\beta - \alpha^T x^* : \alpha^T x \geq \beta\ \forall x \in S\right\}$, where the constraint can be further dualized as

$$\begin{aligned} \text{SEP}(S, x^*) : Z^* := \max_{\alpha,\beta,\mu} \{\beta - \alpha^T x^* \ &: b^T \mu \geq \beta, A^T \mu = \alpha, B^T \mu = 0, \\ &\mu \in K^*, -e \leq \alpha \leq e, -1 \leq \beta \leq 1\}, \end{aligned} \tag{167}$$

where K^* is the dual cone of K. If $Z^* \leq 0$, then $x^* \in S$, otherwise, the optimal α, β from the above program gives the desired separating hyperplane.

REFERENCES

[1] *User's Manual for CPLEX Version 12.4*. 2011.

[2] *MOSEK Modeling Manual*. 2013.

[3] *User's Manual for CPLEX Version 12.6*. 2014.

[4] Angulo, G., Ahmed, S., Dey, S. S., and Kaibel, V., "Forbidden vertices," *Mathematics of Operations Research*, vol. 40, no. 2, pp. 350–360, 2014.

[5] Bai, X. and Wei, H., "Semi-definite programming-based method for security-constrained unit commitment with operational and optimal power flow constraints," *IET Generation Transmission & Distribution*, vol. 3, no. 2, pp. 182–197, 2009.

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

[7] Balas, E., "Disjunctive programming," *Annals of Discrete Mathematics*, vol. 5, pp. 3–51, 1979.

[8] Baran, M. E. and Wu, F. F., "Optimal capacitor placement on radial distribution systems," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 725–734, 1989.

[9] Baran, M. E. and Wu, F. F., "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 735–743, 1989.

[10] Barrows, C., Blumsack, S., and Bent, R., "Computationally efficient optimal transmission switching: Solution space reduction," in *Power and Energy Society General Meeting, 2012 IEEE*, pp. 1–8, July 2012.

[11] Barrows, C., Blumsack, S., and Hines, P., "Correcting optimal transmission switching for AC power flows," in *47th Hawaii International Conference on System Sciences (HICSS)*, pp. 2374–2379, 2014.

[12] Ben-Tal, A. and Nemirovski, A., "On polyhedral approximations of the second-order cone," *Math. Oper. Res.*, vol. 26, pp. 193–205, May 2001.

[13] Bienstock, D. and Mattia, S., "Using mixed-integer programming to solve power grid blackout problems," *Discrete Optimization*, vol. 4, pp. 115–141, Mar. 2007.

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

[15] Blumsack, S., *Network Topologies and Transmission Investment Under Electric-Industry Restructuring*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 2006.

[16] Bollobás, B., *Modern Graph Theory*. Springer, 2002.

[17] Bose, S., Gayme, D. F., Chandy, K. M., and Low, S. H., "Quadratically constrained quadratic programs on acyclic graphs with application to power flow," *arXiv preprint arXiv:1203.5599*, 2012.

[18] Bose, S., Gayme, D. F., Low, S., and Chandy, K. M., "Optimal power flow over tree networks," in *49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1342–1348, 2011.

[19] Bukhsh, W. A., Grothey, A., McKinnon, K., and Trodden, P., "Local solutions of optimal power flow," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4780–4788, 2013.

[20] Cain, M. B., O'Neill, R. P., and Castillo, A., "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.

[21] Carpentier, J., "Contributions to the economic dispatch problem," *Bulletin Society Francaise Electriciens*, vol. 8, no. 3, pp. 431–447, 1962.

[22] Chen, C., Atamtürk, A., and Oren, S. S., "Bound tightening for the alternating current optimal power flow problem," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1–8, 2015.

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

[24] Christie, R., "Power system test archive," 1999. `http://www.ee.washington.edu/research/pstca`.

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

[26] Coffrin, C., Hijazi, H., Hentenryck, P. V., and Lehmann, K., "Primal and dual bounds for optimal transmission switching," in *Power Systems Computation Conference (PSCC)*, (Poland), 2014.

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

[28] Coffrin, C. and Van Hentenryck, P., "A linear-programming approximation of AC power flows," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, 2014.

[29] Dolan, E. and Moré, J., "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, pp. 201–213, 2002.

[30] Expósito, A. G. and Ramos, E. R., "Reliable load flow technique for radial distribution networks," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1063–1069, 1999.

[31] FERC, "Recent ISO Software Enhancements and Future Software and Modeling Plans." http://www.ferc.gov/industries/electric/indus-act/rto/rto-iso-soft-2011.pdf, 2011.

[32] Ferreira, R., Borges, C., and Pereira, M., "A flexible mixed-integer linear programming approach to the ac optimal power flow in distribution systems," *IEEE Transactions on Power Systems*, vol. 29, pp. 2447–2459, Sept 2014.

[33] Fisher, E., O'Neill, R., and Ferris, M., "Optimal transmission switching," *IEEE Transactions on Power Systems*, pp. 1346–1355, 2008.

[34] Frank, S., Steponavice, I., and Rebennack, S., "Optimal power flow: A bibliographic survey I – formulations and deterministic methods," *Energy Systems*, vol. 3, no. 3, pp. 221–258, 2012.

[35] Frank, S., Steponavice, I., and Rebennack, S., "Optimal power flow: A bibliographic survey II – nondeterministic and hybrid methods," *Energy Systems*, vol. 3, no. 3, pp. 259–289, 2012.

[36] Fukuda, M., Kojima, M., Murota, K., and Nakata, K., "Exploiting sparsity in semidefinite programming via matrix completion I: General framework," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 647–674, 2001.

[37] Fuller, J., Ramasra, R., and Cha, A., "Fast heuristics for transmission-line switching," *IEEE Transactions on Power Systems*, vol. 27, pp. 1377–1386, Aug 2012.

[38] Garey, M. and Johnson, D., *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[39] Han, J. and Papavasiliou, A., "The impacts of transmission topology control on the European electricity network," *IEEE Transactions on Power Systems*, vol. 31, pp. 496–507, Jan 2016.

[40] Hariharan, R., Kavitha, T., and Mehlhorn, K., "Faster algorithms for minimum cycle basis in directed graphs," *SIAM Journal on Computing*, vol. 38, no. 4, pp. 1430–1447, 2008.

[41] Hedman, K., Ferris, M., O'Neill, R., Fisher, E., and Oren, S., "Co-optimization of generation unit commitment and transmission switching with N-1 reliability," *IEEE Transactions on Power Systems*, vol. 25, pp. 1052–1063, May 2010.

[42] Hedman, K., O'Neill, R., Fisher, E., and Oren, S., "Optimal transmission switching with contingency analysis," *IEEE Transactions on Power Systems*, vol. 24, pp. 1577–1586, Aug 2009.

[43] Hedman, K., Oren, S., and O'Neill, R., "A review of transmission switching and network topology optimization," in *Power and Energy Society General Meeting, 2011 IEEE*, pp. 1–7, July 2011.

[44] Hijazi, H., Coffrin, C., and Hentenryck, P. V., "Polynomial sdp cuts for optimal power flow," *arXiv preprint arXiv:1510.08107*, 2015.

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

[46] Hillestad, R. J. and Jacobsen, S. E., "Linear programs with an additional reverse convex constraint," *Applied Mathematics and Optimization*, vol. 6, no. 1, pp. 257–269, 1980.

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

[48] Jabr, R. A., "Exploiting sparsity in SDP relaxations of the OPF problem," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 1138–1139, 2012.

[49] Jabr, R. A., Coonick, A. H., and Cory, B. J., "A primal-dual interior point method for optimal power flow dispatching," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 654–662, 2002.

[50] Jabr, R., "Optimization of AC transmission system planning," *IEEE Transactions on Power Systems*, vol. 28, pp. 2779–2787, Aug 2013.

[51] Jabr, R., Singh, R., and Pal, B., "Minimum loss network reconfiguration using mixed-integer convex programming," *IEEE Transactions on Power Systems*, vol. 27, pp. 1106–1115, May 2012.

[52] Jabr, R. A., "Radial distribution load flow using conic programming," *IEEE Transactions on Power Systems*, vol. 21, no. 3, pp. 1458–1459, 2006.

[53] Jabr, R. A., "A conic quadratic format for the load flow equations of meshed networks," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2285–2286, 2007.

[54] Jabr, R. A., "Optimal power flow using an extended conic quadratic formulation," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1000–1008, 2008.

[55] Josz, C., Maeght, J., Panciatici, P., and Gilbert, J. C., "Application of the moment-sos approach to global optimization of the OPF problem," *IEEE Transactions on Power Systems*, 2014.

[56] Kavitha, T., Liebchen, C., Mehlhorn, K., Michail, D., Rizzi, R., Ueckerdt, T., and Zweig, K., "Cycle bases in graphs: Characterization, algorithms, complexity, and applications," *Computer Science Review*, vol. 3, pp. 199–243, 2009.

[57] Khanabadi, M., Ghasemi, H., and Doostizadeh, M., "Optimal transmission switching considering voltage security and N-1 contingency analysis," *IEEE Transactions on Power Systems*, vol. 28, pp. 542–550, Feb 2013.

[58] Khodaei, A., Shahidehpour, M., and Kamalinia, S., "Transmission switching in expansion planning," *IEEE Transactions on Power Systems*, vol. 25, pp. 1722–1733, Aug 2010.

[59] Kocuk, B., Dey, S. S., and Sun, X. A., "New formulation and strong MISOCP relaxations for AC optimal transmission switching problem," *arXiv preprint arXiv:1411.0359*, 2015.

[60] Kocuk, B., Dey, S. S., and Sun, X. A., "Strong SOCP relaxations for the optimal power flow problem," *to appear in Operations Research*, 2016.

[61] Kocuk, B., Dey, S., and Sun, X., "Inexactness of SDP relaxation and valid inequalities for optimal power flow," *IEEE Transactions on Power Systems*, vol. 31, pp. 642–651, Jan 2016.

[62] Kocuk, B., Jeon, H., Dey, S. S., Linderoth, J., Luedtke, J., and Sun, X. A., "A cycle-based formulation and valid inequalities for DC power transmission problems with switching," *to appear in Operations Research*, 2016.

[63] Korad, A. and Hedman, K., "Robust corrective topology control for system reliability," *IEEE Transactions on Power Systems*, vol. 28, pp. 4042–4051, Nov 2013.

[64] Korad, A. and Hedman, K., "Enhancement of do-not-exceed limits with robust corrective topology control," *to appear in IEEE Transactions on Power Systems*, 2015.

[65] Lasserre, J. B., "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[66] Lavaei, J., Tse, D., and Zhang, B., "Geometry of power flows and optimization in distribution networks," *IEEE Transactions on Power Systems*, vol. 29, pp. 572–583, March 2014.

[67] Lavaei, J. and Low, S. H., "Zero duality gap in optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 92–107, 2012.

[68] Lehmann, K., Grastien, A., and Hentenryck, P. V., "The complexity of DC-Switching problems," technical report, NICTA, 2014.

[69] Lesieutre, B. C., Molzhan, D. K., Borden, A. R., and DeMarco, C. L., "Examining the limits of the application of semidefinite programming to power flow problems," in *Forty-Nine Annual Allerton Conference*, pp. 1492–1499, 2011.

[70] Lincoln, R., "Learning to trade power." `https://github.com/rwl/thesis/blob/master/tikz/case6ww.tex`, 2010.

[71] Low, S. H., "Convex relaxation of optimal power flow, part I: Formulations and equivalence," *IEEE Transactions on Control of Network Systems*, vol. 1, pp. 15–27, March 2014.

[72] Low, S. H., "Convex relaxation of optimal power flow, part II: Exactness," *IEEE Transactions on Control of Network Systems*, vol. 1, pp. 177–189, June 2014.

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

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

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

[76] Madani, R., Sojoudi, S., and Lavaei, J., "Convex relaxation for optimal power flow problem: Mesh networks," in *ACSSC*, pp. 1375–1382, 2013.

[77] McCormick, G. P., "Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems," *Mathematical Programming*, vol. 10, no. 1, pp. 147–175, 1976.

[78] Misener, R., Thompson, J. P., and Floudas, C. A., "Apogee: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes," *Computers & Chemical Engineering*, vol. 35, no. 5, pp. 876 – 892, 2011.

[79] Molzahn, D. K., Holzer, J. T., Lesieutre, B. C., and DeMarco, C. L., "Implementation of a large-scale optimal power flow solver based on semidefinite programming," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3987–3998, 2013.

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

[81] Momoh, J. A., El-Hawary, M. E., and Adapa, R., "A review of selected optimal power flow literature to 1993 part I: Nonlinear and quadratic programming approaches," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 96–104, 1999.

[82] Momoh, J. A., El-Hawary, M. E., and Adapa, R., "A review of selected optimal power flow literature to 1993 part II: Newton, linear programming and interior point methods," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 105–111, 1999.

[83] Nakata, K., Fujisawa, K., Fukuda, M., Kojima, M., and Murota, K., "Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results," *Mathematical Programming*, vol. 95, no. 2, pp. 303–327, 2003.

[84] O'Neill, R., Baldick, R., Helman, U., Rothkopf, M., and Stewart, J., "Dispatchable transmission in RTO markets," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 171–179, 2005.

[85] Ostrowski, J., Wang, J., and Liu, C., "Transmission switching with connectivity-ensuring constraints," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1–7, 2014.

[86] Parrilo, P., "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, no. 2, pp. 293–320, 2003.

[87] Phan, D. T., "Lagrangian duality and branch-and-bound algorithms for optimal power flow," *Operations Research*, vol. 60, no. 2, pp. 275–285, 2012.

[88] Ruiz, P., Forster, J. M., Rudkevich, A., and Caramanis, M. C., "On fast transmission topology control heuristics," *IEEE Power and Energy Society General Meeting*, pp. 1–8, 2011.

[89] Ruiz, P., Forster, J. M., Rudkevich, A., and Caramanis, M. C., "Tractable transmission topology control using sensitivity analysis," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1550–1559, 2012.

[90] Sahraei-Ardakani, M., Korad, A., Hedman, K., Lipka, P., and Oren, S., "Performance of AC and DC based transmission switching heuristics on a

large-scale polish system," in *PES General Meeting — Conference Exposition, 2014 IEEE*, pp. 1–5, July 2014.

[91] Sojoudi, S. and Lavaei, J., "Physics of power networks makes hard optimization problems easy to solve," in *IEEE Power and Energy Society General Meeting*, pp. 1–8, 2012.

[92] Sojoudi, S. and Lavaei, J., "On the exactness of semidefinite relaxation for nonlinear optimization over graphs: Part II," in *IEEE 52nd Annual Conference on Decision and Control (CDC)*, pp. 1051–1057, Dec 2013.

[93] Soroush, M. and Fuller, J., "Accuracies of optimal transmission switching heuristics based on DCOPF and ACOPF," *IEEE Transactions on Power Systems*, vol. 29, pp. 924–932, March 2014.

[94] Sun, A. and Phan, D. T., *Wiley Encyclopedia of Operations Research and Management Science*, ch. Some Optimization Models and Techniques for Electric Power System Short-term Operations. John Wiley & Sons, Inc., 2013.

[95] Tawarmalani, M. and Sahinidis, N. V., "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming*, vol. 103, no. 2, pp. 225–249, 2005.

[96] Tawarmalani, M. and Sahinidis, N. V., *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*, vol. 65. Springer, 2002.

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

[98] Thompson, C., McIntyre, K., Nuthalapati, S., Garcia, A., and Villanueva, E., "Real-time contingency analysis methods to mitigate congestion in the ERCOT region," in *Power Energy Society General Meeting, 2009. PES '09. IEEE*, pp. 1–7, July 2009.

[99] Torres, G. L. and Quintana, V. H., "An interior-point method for nonlinear optimal power flow using voltage rectangular coordinates," *IEEE Transactions on Power Systems*, vol. 13, no. 4, pp. 1211–1218, 1998.

[100] Trodden, P., Bukhsh, W., Grothey, A., and McKinnon, K., "Optimization-based islanding of power networks using piecewise linear AC power flow," *IEEE Transactions on Power Systems*, vol. 29, pp. 1212–1220, May 2014.

[101] Van Acker, T., Van Hertem, D., Bekaert, D., Karoui, K., and Merckx, C., "Implementation of bus bar switching and short circuit constraints in optimal power flow problems," in *PowerTech, 2015 IEEE Eindhoven*, pp. 1–6, June 2015.

[102] Verma, A., *Power grid security analysis: An optimization approach*. PhD thesis, Columbia University, 2009.

[103] Villumsen, J. C. and Philpott, A. B., "Investment in electricity networks with transmission switching," *European Journal of Operational Research*, vol. 222, pp. 377–385, 2012.

[104] Villumsen, J., Bronmo, G., and Philpott, A., "Line capacity expansion and transmission switching in power systems with large-scale wind power," *IEEE Transactions on Power Systems*, vol. 28, pp. 731–739, May 2013.

[105] Wächter, A. and Biegler, L. T., "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[106] Wang, H., Murillo-Sánchez, C. E., Zimmerman, R. D., and Thomas, R. J., "On computational issues of market based optimal power flow," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1185–1193, 2007.

[107] Wu, J. and Cheung, K., "On selection of transmission line candidates for optimal transmission switching in large power networks," in *Power and Energy Society General Meeting (PES), 2013 IEEE*, pp. 1–5, July 2013.

[108] Wu, Y., Debs, A. S., and Marsten, R. E., "A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows," *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 876–883, 1994.

[109] Zhang, B. and Tse, D., "Geometry of feasible injection region of power networks," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 788–797, 2013.

[110] Zhang, B. and Tse, D., "Geometry of feasible injection region of power networks," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pp. 1508–1515, Sept 2011.

[111] Zimmerman, R., Murillo-Sanchez, C., and Thomas, R., "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, pp. 12–19, Feb 2011.