Tightening Quadratic Convex Relaxations for the AC Optimal Transmission Switching Problem

Cheng Guo
School of Mathematical and Statistical Sciences, Clemson University, Clemson, SC 29634 USA, cguo2@clemson.edu

Harsha Nagarajan
Applied Mathematics and Plasma Physics (T-5), Los Alamos National Laboratory, Los Alamos, NM 87545 USA, harsha@lanl.gov

Merve Bodur
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S 3G8 Canada, bodur@mie.utoronto.ca

The Alternating Current Optimal Transmission Switching (ACOTS) problem incorporates line switching decisions into the fundamental AC optimal power flow (ACOPF) problem. The advantages of the ACOTS problem are well-known in terms of reducing the operational cost and improving system reliability. ACOTS optimization models contain discrete variables and nonlinear, non-convex constraints, which make it difficult to solve. In this work, we develop strengthened quadratic convex (QC) relaxations for ACOTS, where we tighten the relaxation with several new valid inequalities, including a novel kind of on/off cycle-based polynomial constraints by taking advantage of the network structure. We linearize the sum of on/off trilinear terms that appears in the relaxation with extreme-point representation and theoretically show its tightness. Also, we efficiently incorporate on/off cycle-based polynomial constraints using disjunctive programming based cutting planes. Combined with the optimization-based bound tightening algorithm, we obtain the tightest QC-based ACOTS relaxation to the best of our knowledge. Our extensive numerical experiments on medium-scale PGLib instances show significant improvement on relaxation bounds for many instances.

Key words: Optimal transmission line switching, On/off quadratic convex relaxation, Convex hull, Bound tightening.

1. Introduction

Transmission switching, with its inception in the 1980s (Glavitsch 1985), has gained considerable attention in both industry and academia in recent years (Hedman et al. 2008, Kocuk et al. 2017). The optimal transmission switching (OTS) problem studies how to switch on or off certain transmission lines to modify the network topology in the real-time operation of transmission power grids. Solving the OTS problem brings several benefits that the traditional optimal power flow (OPF) problem solution cannot offer, such as reducing the total operational cost, mitigating transmission congestion, clearing contingencies, and improving engineering limits (Hedman et al. 2011). Thus, as the modern transmission

control and relay technologies evolve, transmission line switching has become an important option in power system operators' toolkits.

Previous literature on OTS has mainly relied on the DC approximation of the power flow model to avoid the mathematical complexity of the non-convex AC power flow equations. The first formal mathematical model for the OTS problem, proposed by Fisher et al. (2008), is based on the DC approximation of the power flow equations. Kocuk et al. (2016b) derive a cycle-induced relaxation for a DCOTS model, and characterize the convex hull of this relaxation.

The downside of DCOTS approximation is that the optimal decisions may not necessarily represent accurate power flows or even be infeasible in the AC setting (Coffrin et al. 2014). Those drawbacks motivate the adoption of ACOTS. The ACOTS problem can be formulated as a non-convex mixed-integer nonlinear program (MINLP), which is challenging to solve. Moreover, even in the DC setting, the OTS problem is known to be NP-hard (Lehmann et al. 2014). Several heuristics are proposed for solving the ACOTS problem, e.g., by Barrows et al. (2014) and Goldis et al. (2013). Separately, convex relaxations for ACOTS have gained significant attention in recent years.

The convex relaxations of the ACOTS problem, of which the literature is quite scarce, are built upon the rich literature on ACOPF convex relaxations, such as the second-order cone (SOC) relaxation (Jabr 2006), the quadratic convex (QC) relaxation (Coffrin et al. 2016), and the semi-definite programming (SDP) relaxation (Bai et al. 2008). For the ACOPF problem, the standard SDP and QC relaxations are at least as strong as the SOC relaxation, while the strength of the SDP and QC relaxations are not comparable. Computationally, the SOC and QC relaxations are faster and more reliable than the SDP relaxation (Coffrin et al. 2016). In the ACOTS setting, Hijazi et al. (2017) and Bestuzheva et al. (2020) propose a QC relaxation that incorporates on/off decision variables, which provides a tight lower bound to the generation-cost minimization objective. We further tighten this on/off version of the QC relaxation with a much stronger linearization and valid inequalities, with some of the latter being novel.

In particular, one type of valid inequalities we incorporate is cycle-based polynomial constraints ("lifted cycle constraints" for short). The lifted cycle constraints were first proposed by Kocuk et al. (2016a) as a relaxation to the arc-tangent constraints in ACOPF. Their work provides a reformulated SOC relaxation for ACOPF, strengthened by the

McCormick relaxation of the lifted cycle constraints, which is shown to be incomparable to the SDP relaxation. Further, these cycle-based valid inequalities were generalized for the ACOTS problem by Kocuk et al. (2017), which was derived in the rectangular co-ordinates. However, in our work, we develop a new type of lifted cycle constraints based on the QC relaxation of the ACOPF problem, written in polar co-ordinates. Our method takes advantage of the direct access to auxiliary variables representing the trigonometric functions in the QC relaxation. Then we reformulate the on/off version of those lifted cycle constraints for the ACOTS problem, which is new in the literature. In our strengthened formulation, we combine those new constraints and the lifted cycle constraints from (Kocuk et al. 2017). Further, we linearize these on/off polynomial constraints with the tightest extreme-point representation, which captures the convex hull of the on/off lifted cycle constraints for a given cycle. Since adding all constraints at the same time can make the problem challenging to solve, we derive novel cutting planes and incorporate lifted cycle constraints via a branch-and-cut scheme.

We further improve the bounds using an optimization-based bound tightening (OBBT) technique. OBBT is often used in MINLPs to tighten relaxation bounds (Nagarajan et al. 2019a). It has been shown to be an effective bound-tightening method in nonlinear AC power flow models. Sundar et al. (2018) use OBBT to tighten an ACOPF-QC relaxation, whereas Bestuzheva et al. (2020) use OBBT for an ACOTS-QC relaxation with nonlinear terms linearized using weaker recursive-McCormick relaxations. In our work, we incorporate all the proposed tightening valid inequalities and the ones from the literature within OBBT, to achieve a very tight lower bound for the ACOTS problem.

Our main contributions can be summarized as follows:

(1) We strengthen the ACOTS-QC relaxation with several techniques. First, we linearize the summation of two on/off trilinear terms with the tight extreme-point representation and prove such a linearization captures its convex hull. To the best of our knowledge, we are the first to develop this linearization method for ACOTS. We also reformulate several ACOPF-QC strengthening constraints for the OTS setting, some of which are novel. Compared with the state-of-the-art on/off QC relaxation formulation (Coffrin et al. 2018), our strengthened relaxation is shown to provide better lower bounds for many test cases, and some of those improvements are substantial.

(2) We derive a new lifted cycle constraint that strengthens the QC relaxations of both ACOPF and ACOTS. We linearize those constraints with the extreme-point representation, which is always tighter than the recursive-McCormick relaxation in the literature. Also, to separate lifted cycle constraints with discrete decisions, we develop novel cutting planes by reformulating Benders cuts via disjunctive programming. We incorporate those cutting planes in a branch-and-cut framework, which leads to significant solution time reductions.

(3) We combine strengthening methods listed in (1) and (2) with OBBT and obtain the tightest QC-based ACOTS relaxation in the literature, to the best of our knowledge.

2. The ACOTS Problem

Before describing the model for the ACOTS problem, we first introduce some notation. Throughout, constants are typeset in boldface to make it easier to distinguish between decision variables and parameters. In the AC power flow equations, upper case letters represent complex quantities. $\mathfrak{R}(\cdot)$ and $\mathfrak{T}(\cdot)$ respectively denote the real and imaginary parts of a complex number. Given any two complex numbers (variables/constants) z_1 and z_2, $z_1 \geqslant z_2$ implies $\mathfrak{R}(z_1) \geqslant \mathfrak{R}(z_2)$ and $\mathfrak{T}(z_1) \geqslant \mathfrak{T}(z_2)$. $|\cdot|$ and $(\cdot)^*$ represent the magnitude and Hermitian conjugate of a complex number, respectively. When applied on a real-valued number, $|\cdot|$ represents its absolute value. $\langle\cdot\rangle^R$ represents the convex envelope of a function. The notation for sets, parameter and variables is summarized in Table 1.

Table 1: Notation

Sets and parameters:	
$\mathcal{N}$	Set of buses (nodes).
$\mathcal{N}^L$	Set of leaf buses with no loads.
$\mathcal{N}^{\text{ref}}$	Set of reference buses.
$\mathcal{G}$	Set of generators.
$\mathcal{G}_i$	Set of generators at bus i.
$\mathcal{A}$	Set of lines (arcs).
$\mathcal{A}^R$	Set of arcs in reversed direction.
$\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{c}_2$	Generation cost coefficients.
$\mathbf{j}$	Unit imaginary number.
$\boldsymbol{Y}_{ij} = \boldsymbol{g}_{ij} + \mathbf{j}\boldsymbol{b}_{ij}$	Admittance on line (i,j).
$\boldsymbol{Y}^c_{ij} = \boldsymbol{g}^c_{ij} + \mathbf{j}\boldsymbol{b}^c_{ij}$	Charging admittance on line (i,j).
$\boldsymbol{Y}^s_{ij} = \boldsymbol{g}^s_{ij} + \mathbf{j}\boldsymbol{b}^s_{ij}$	Shunt admittance on line (i,j).
$\boldsymbol{T}_{ij} = \boldsymbol{t}^R_{ij} + \mathbf{j}\boldsymbol{t}^I_{ij}$	Branch complex transformation ratio (tap ratio) on line (i,j).
$\boldsymbol{S}^{\boldsymbol{d}}_i = \boldsymbol{p}^{\boldsymbol{d}}_i + \mathbf{j}\boldsymbol{q}^{\boldsymbol{d}}_i$	AC power demand at bus i.
$\overline{\boldsymbol{s}}_{ij}$	Apparent power bound on line (i,j).
$\underline{\boldsymbol{\theta}}_{ij}, \overline{\boldsymbol{\theta}}_{ij}$	Phase angle difference bounds on line (i,j).
$\underline{\boldsymbol{v}}_i, \overline{\boldsymbol{v}}_i$	Voltage magnitude bounds at bus i.
$\underline{\boldsymbol{S}}^{\boldsymbol{g}}_i, \overline{\boldsymbol{S}}^{\boldsymbol{g}}_i$	Power generation bounds at bus i.

$\bar{\boldsymbol{l}}_{ij}$	Current magnitude squared upper limit on line (i,j).
Variables:	
v_i	Voltage magnitude at bus i.
θ_i	Voltage angle at bus i.
$V_i = v_i \boldsymbol{e}^{\mathbf{j}\theta_i}$	AC complex voltage at bus i.
θ_{ij}	Phase angle difference on line (i,j).
w_i	Squared voltage magnitude at bus i.
$W_{ij} = w_{ij}^R + \mathbf{j} w_{ij}^I$	AC voltage product on line (i,j).
$S_{ij} = p_{ij} + \mathbf{j} q_{ij}$	AC power flow on line (i,j).
$S_k^g = p_k^g + \mathbf{j} q_k^g$	AC power generation of generator k.
l_{ij}	Current magnitude squared on line (i,j).
z_{ij}	1 if line (i,j) is switched on, 0 otherwise.

The power network can be represented with the graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ corresponds to the set of buses, while the set of arcs $\mathcal{A}$ corresponds to the set of lines. Note that we assume lines in the network are directed with designated from/to buses, as indicated by the data. This assumption is conventional, and it is necessary because the data contains asymmetric shunt conductance and transformers.

The ACOTS problem minimizes the total production cost of generators such that all the demands at the buses, the physical constraints (e.g., Ohm's and Kirchoff's law), and engineering limit constraints (e.g., transmission line flow limits) are satisfied. In this work, for the purpose of the QC relaxation (see Section 3), we model the AC power flow equations in *polar* co-ordinates (Taylor 2015), thus the following ACOTS formulation:

$$\min \sum_{k\in\mathcal{G}} \left(\boldsymbol{c}_{2k}(p_k^g)^2 + \boldsymbol{c}_{1k} p_k^g\right) + \sum_{i\in\mathcal{N}\setminus\mathcal{N}^L} \sum_{k\in\mathcal{G}_i} \boldsymbol{c}_{0k} + \sum_{i\in\mathcal{N}^L} \sum_{k\in\mathcal{G}_i} \boldsymbol{c}_{0k} z_{\{ft|(f,t)\in\mathcal{A}, f=i \text{ or } t=i\}} \tag{1a}$$

$$\text{s.t.} \sum_{k\in\mathcal{G}_i} S_k^g - \boldsymbol{S}_i^{\boldsymbol{d}} - \boldsymbol{Y}_{ij}^{s*} w_i = \sum_{(i,j)\in\mathcal{A}\cup\mathcal{A}^R} S_{ij} \qquad \forall i \in \mathcal{N} \tag{1b}$$

$$S_{ij} = (\boldsymbol{Y}_{ij} + \boldsymbol{Y}_{ij}^c)^* \frac{w_i}{|\boldsymbol{T}_{ij}|^2} z_{ij} - \boldsymbol{Y}_{ij}^* \frac{W_{ij}}{\boldsymbol{T}_{ij}} \qquad \forall (i,j) \in \mathcal{A} \tag{1c}$$

$$S_{ji} = (\boldsymbol{Y}_{ij} + \boldsymbol{Y}_{ji}^c)^* w_j z_{ij} - \boldsymbol{Y}_{ij}^* \frac{W_{ij}^*}{\boldsymbol{T}_{ij}^*} \qquad \forall (i,j) \in \mathcal{A} \tag{1d}$$

$$w_i = v_i^2 \qquad \forall i \in \mathcal{N} \tag{1e}$$

$$W_{ij} = V_i V_j^* z_{ij} \qquad \forall (i,j) \in \mathcal{A} \tag{1f}$$

$$\theta_{ij} = \theta_i - \theta_j \qquad \forall (i,j) \in \mathcal{A} \tag{1g}$$

$$\underline{\boldsymbol{\theta}}_{ij} z_{ij} - \boldsymbol{\theta}^{\mathbf{M}}(1 - z_{ij}) \leqslant \theta_{ij} \leqslant \overline{\boldsymbol{\theta}}_{ij} z_{ij} + \boldsymbol{\theta}^{\mathbf{M}}(1 - z_{ij}) \qquad \forall (i,j) \in \mathcal{A} \tag{1h}$$

$$\theta_i = 0 \qquad \forall i \in \mathcal{N}^{\text{ref}} \tag{1i}$$

$$\underline{\boldsymbol{S}}_i^{\boldsymbol{g}} \leqslant S_k^g \leqslant \overline{\boldsymbol{S}}_k^{\boldsymbol{g}} \qquad \forall k \in \mathcal{G} \tag{1j}$$

$$|S_{ij}|^2 \leqslant \overline{\boldsymbol{s}}_{ij}^2 z_{ij}^2, \quad |S_{ji}|^2 \leqslant \overline{\boldsymbol{s}}_{ij}^2 z_{ij}^2 \qquad \forall (i,j) \in \mathcal{A} \tag{1k}$$

$$\underline{\boldsymbol{v}}_i \leqslant v_i \leqslant \overline{\boldsymbol{v}}_i \qquad \forall i \in \mathcal{N} \tag{1l}$$

$$z_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}, \tag{1m}$$

where $z_{\{ft|(f,t)\in\mathcal{A},f=i \text{ or } t=i\}}$ is the switch on/off variable for a line having either end connected to a leaf node with 0 load. When such a line is switched off, the generators on the leaf node are disconnected from the network, and thus we do not need to pay the fixed cost c_{0k}.

The convex quadratic objective (1a) minimizes total generator dispatch cost. Constraints (1b) correspond to the power balance at each bus, i.e., Kirchoff's current law. Constraints (1c) to (1f) model the power flow on each line. Note that constraints (1c) and (1d) ensure that the power flow over line (i,j) is zero if the line is switched off; constraints (1f) ensure that $W_{ij} = 0$ when line (i,j) is switched off. Constraints (1g) connect voltage angle and voltage difference variables.

Constraints (1h) limit the phase angle difference on each line. We define $\boldsymbol{\theta}_{ij}^{\mathbf{u}} = \max(|\underline{\boldsymbol{\theta}}_{ij}|, |\overline{\boldsymbol{\theta}}_{ij}|)$. Let $\boldsymbol{\theta}_{ij,k}^{\boldsymbol{u},\max}$ be the kth largest value in $\{\boldsymbol{\theta}_{ij}^{\mathbf{u}} \mid (i,j) \in \mathcal{A}\}$, and $\boldsymbol{\theta}^{\mathbf{M}} = \sum_{k=1}^{|\mathcal{N}|-1} \boldsymbol{\theta}_{ij,k}^{\boldsymbol{u},\max}$ be a big-M constant for phase angle difference. This big-M constant enables us to provide proper bounds for θ_{ij} in constraints (1h).

Constraints (1i) set the voltage angles of reference buses to 0. Constraints (1j) restrict the apparent power output of each generator. Constraints (1k) are thermal limit constraints that restrict the total electric power transmitted on each line. Note that in these constraints a squared form of z_{ij} is used instead of the linear form, as it produces a tighter formulation (Hijazi et al. 2017). Constraints (1l) limit the voltage magnitude at each bus.

This ACOTS model is a non-convex MINLP, which contains nonlinear constraints (1c) through (1f). An easy method to linearize constraints (1c) and (1d) is to introduce a lifted variable w_{ij}^z per line, which equals w_i when $z_{ij} = 1$ and 0 otherwise, then constraints (1c) and (1d) can be replaced with the following linear constraints for every line $(i,j) \in \mathcal{A}$:

$$S_{ij} = (\boldsymbol{Y}_{ij} + \boldsymbol{Y}_{ij}^c)^* \frac{w_{ij}^z}{|\boldsymbol{T}_{ij}|^2} - \boldsymbol{Y}_{ij}^* \frac{W_{ij}}{\boldsymbol{T}_{ij}} \tag{2a}$$

$$S_{ji} = (\boldsymbol{Y}_{ij} + \boldsymbol{Y}_{ji}^c)^* w_{ji}^z - \boldsymbol{Y}_{ij}^* \frac{W_{ij}^*}{\boldsymbol{T}_{ij}^*} \tag{2b}$$

$$w_i - (1 - z_{ij})\overline{\boldsymbol{v}}_i^2 \leqslant w_{ij}^z \leqslant w_i - (1 - z_{ij})\underline{\boldsymbol{v}}_i^2 \tag{2c}$$

$$w_j - (1 - z_{ij})\overline{\boldsymbol{v}}_j^2 \leqslant w_{ji}^z \leqslant w_j - (1 - z_{ij})\underline{\boldsymbol{v}}_j^2 \tag{2d}$$

$$\underline{\boldsymbol{v}}_i^2 z_{ij} \leqslant w_{ij}^z \leqslant \overline{\boldsymbol{v}}_i^2 z_{ij} \tag{2e}$$

$$\underline{\boldsymbol{v}}_j^2 z_{ij} \leqslant w_{ji}^z \leqslant \overline{\boldsymbol{v}}_j^2 z_{ij}. \tag{2f}$$

Constraints (2a) - (2d) are from (Hijazi et al. 2017), while constraints (2e) and (2f) are new and provide tighter bounds for w_{ij}^z and w_{ji}^z. Tractable convex relaxations for constraints (1e) and (1f) are described in the next section.

3. On/Off Quadratic Convex Relaxation

The on/off version of the QC relaxation for the ACOTS model relaxes nonlinear constraints (1e) and (1f). Let $w_{ij}^R := \Re(W_{ij})$ and $w_{ij}^I := \Im(W_{ij})$, then (1f) can be equivalently written as follows ($\forall (i,j) \in \mathcal{A}$):

$$w_{ij}^R = z_{ij}\left(v_i v_j \cos(\theta_{ij})\right) \tag{3a}$$

$$w_{ij}^I = z_{ij}\left(v_i v_j \sin(\theta_{ij})\right). \tag{3b}$$

A key feature of the QC relaxation is the use of polar co-ordinates, which has direct access to voltage magnitude v_i and voltage angle θ_i variables. This enables stronger links between voltage variables. In what follows, we list the QC relaxation constraints, some of which are based on previous works (Coffrin et al. 2016, Hijazi et al. 2017), while others are newly derived, which we will specify.

(1) *Quadratic function relaxation ($\forall i \in \mathcal{N}$):* We can formulate the convex-hull envelope for constraints (1e) by relaxing every equality in to a quadratic inequality constraint (4a), and providing upper bounds via linear McCormick relaxation constraints in (4b):

$$w_i \geqslant v_i^2 \tag{4a}$$

$$w_i \leqslant (\underline{\boldsymbol{v}}_i + \overline{\boldsymbol{v}}_i)v_i - \underline{\boldsymbol{v}}_i\overline{\boldsymbol{v}}_i \tag{4b}$$

(2) *Cosine and sine function relaxations* ($\forall (i,j) \in \mathcal{A}$)*:* The trigonometric term $\cos(\theta_{ij})$ in (3a) is non-convex. We define a lifted variable c_{ij} that is in the convex envelope of $\cos(\theta_{ij})$, i.e., $c_{ij} \in \langle \cos(\theta_{ij}) \rangle^R$. We assume $\boldsymbol{\theta}_{ij}^{\mathbf{u}} \leqslant \pi/2$, which is reasonable as the absolute value of the phase angle differences across the lines is usually under 15 degrees in practice

(Purchala et al. 2005). We define the following constants for every line $(i,j) \in \mathcal{A}$ which are necessary for the relaxation constraints:

$$\boldsymbol{\theta}^{\boldsymbol{c}}_{ij} = \frac{\cos(\overline{\boldsymbol{\theta}}_{ij}) - \cos(\underline{\boldsymbol{\theta}}_{ij})}{\overline{\boldsymbol{\theta}}_{ij} - \underline{\boldsymbol{\theta}}_{ij}}, \quad \boldsymbol{\theta}^{\boldsymbol{s}}_{ij} = \frac{\sin(\overline{\boldsymbol{\theta}}_{ij}) - \sin(\underline{\boldsymbol{\theta}}_{ij})}{\overline{\boldsymbol{\theta}}_{ij} - \underline{\boldsymbol{\theta}}_{ij}}$$

Following the disjunctive programming method in (Hijazi et al. 2017), an on/off version of the convex relaxation of the cosine function is as follows:

$$-c_{ij} + \boldsymbol{\theta}^{\boldsymbol{c}}_{ij}\theta_{ij} \leqslant \left(\boldsymbol{\theta}^{\boldsymbol{c}}_{ij}\underline{\boldsymbol{\theta}}_{ij} - \cos(\underline{\boldsymbol{\theta}}_{ij})\right) z_{ij} + \left|\boldsymbol{\theta}^{\boldsymbol{c}}_{ij}\right| \boldsymbol{\theta}^{\mathbf{M}}(1 - z_{ij}) \tag{5a}$$

$$c_{ij} \leqslant z_{ij} - \frac{1 - \cos(\boldsymbol{\theta}^{\mathbf{u}}_{ij})}{(\boldsymbol{\theta}^{\mathbf{u}}_{ij})^2}\theta^2_{ij} + \frac{1 - \cos(\boldsymbol{\theta}^{\mathbf{u}}_{ij})}{(\boldsymbol{\theta}^{\mathbf{u}}_{ij})^2}(\boldsymbol{\theta}^{\mathbf{M}})^2(1 - z_{ij}) \tag{5b}$$

$$\underline{\boldsymbol{c}}_{ij} z_{ij} \leqslant c_{ij} \leqslant \overline{\boldsymbol{c}}_{ij} z_{ij}. \tag{5c}$$

Constraints (5a) and (5b) are big-M constraints. When $z_{ij} = 1$, they represent quadratic convex relaxations of $cos(\theta_{ij})$ derived from trigonometric identities and properties of quadratic functions, and those convex relaxations are not valid when the line (i,j) is switched off, as we have $c_{ij} = 0$. Therefore, big-M parameter $\boldsymbol{\theta}^{\mathbf{M}}$ is used to ensure that when $z_{ij} = 0$, those constraints are valid for $c_{ij} = 0$ and $\theta_{ij} \in [\underline{\boldsymbol{\theta}}_{ij}, \overline{\boldsymbol{\theta}}_{ij}]$. Constraint (5c) provides bounds for c_{ij}, and ensure that $c_{ij} = 0$ when the line (i,j) is switched off. Note that constraint (5a) is a new constraint that is not in (Hijazi et al. 2017) or (Bestuzheva et al. 2020).

Similarly, we define $s_{ij} \in \langle \sin(\theta_{ij}) \rangle^R$. When $\boldsymbol{\theta}^{\mathbf{u}}_{ij} \leqslant \pi/2$, a disjunctive relaxation of the sine function is as follows:

$$s_{ij} - \cos\left(\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right)\theta_{ij} \leqslant \left(\sin\left(\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right) - \cos\left(\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right)\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right) z_{ij} + \cos\left(\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right)\boldsymbol{\theta}^{\mathbf{M}}(1 - z_{ij}), \text{ if } \overline{\boldsymbol{\theta}}_{ij} \geqslant 0 \tag{6a}$$

$$-s_{ij} + \cos\left(\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right)\theta_{ij} \leqslant \left(\sin\left(\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right) - \cos\left(\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right)\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right) z_{ij} + \cos\left(\frac{\boldsymbol{\theta}^{\mathbf{u}}_{ij}}{2}\right)\boldsymbol{\theta}^{\mathbf{M}}(1 - z_{ij}), \text{ if } \underline{\boldsymbol{\theta}}_{ij} \leqslant 0 \tag{6b}$$

$$s_{ij} - \boldsymbol{\theta}^{\boldsymbol{s}}_{ij}\theta_{ij} \leqslant \left(-\boldsymbol{\theta}^{\boldsymbol{s}}_{ij}\underline{\boldsymbol{\theta}}_{ij} + \sin(\underline{\boldsymbol{\theta}}_{ij})\right) z_{ij} + \boldsymbol{\theta}^{\boldsymbol{s}}_{ij}\boldsymbol{\theta}^{\mathbf{M}}(1 - z_{ij}), \text{ if } \overline{\boldsymbol{\theta}}_{ij} \leqslant 0 \tag{6c}$$

$$-s_{ij} + \boldsymbol{\theta}^{\boldsymbol{s}}_{ij}\theta_{ij} \leqslant \left(\boldsymbol{\theta}^{\boldsymbol{s}}_{ij}\underline{\boldsymbol{\theta}}_{ij} - \sin(\underline{\boldsymbol{\theta}}_{ij})\right) z_{ij} + \boldsymbol{\theta}^{\boldsymbol{s}}_{ij}\boldsymbol{\theta}^{\mathbf{M}}(1 - z_{ij}), \text{ if } \underline{\boldsymbol{\theta}}_{ij} \geqslant 0 \tag{6d}$$

$$\underline{\boldsymbol{s}}_{ij} z_{ij} \leqslant s_{ij} \leqslant \overline{\boldsymbol{s}}_{ij} z_{ij}, \tag{6e}$$

where constraints (6a) - (6d) are derived from linear outer approximation of the $\sin(\theta_{ij})$ function (Hijazi et al. 2017).

(3) *Extreme-point representation for summation of on/off trilinear terms* ($\forall (i,j) \in \mathcal{A}$)*:* Next, we develop a novel extreme-point representation to linearize the sum of two on/off trilinear terms and theoretically prove its tightness. We first substitute the non-convex functions $\cos(\theta_{ij})$ and $\sin(\theta_{ij})$ in constraints (3) with lifted variables c_{ij} and s_{ij} and their convex envelopes, respectively. The remaining non-linearities reduce to trilinear terms $v_i v_j c_{ij}$ and $v_i v_j s_{ij}$, which are further controlled by the status of the on/off variable z_{ij}. To linearize these terms, we generalize the convex hull representation of the extreme-point formulation (Lu et al. 2018, Sundar et al. 2018) to incorporate on/off variables, as discussed below.

Let the extreme points of the domain $[\underline{\boldsymbol{v}}_i, \overline{\boldsymbol{v}}_i] \times [\underline{\boldsymbol{v}}_j, \overline{\boldsymbol{v}}_j] \times [\underline{\boldsymbol{c}}_{ij}, \overline{\boldsymbol{c}}_{ij}]$ be denoted by $\boldsymbol{\xi}^k$ with $k = 1, \ldots, 8$, and the extreme points of the domain $[\underline{\boldsymbol{v}}_i, \overline{\boldsymbol{v}}_i] \times [\underline{\boldsymbol{v}}_j, \overline{\boldsymbol{v}}_j] \times [\underline{\boldsymbol{s}}_{ij}, \overline{\boldsymbol{s}}_{ij}]$ be denoted by $\boldsymbol{\gamma}^k, k = 1, \ldots, 8$. We relax constraints (3) as follows:

$$w_{ij}^R = \sum_{k=1}^{8} \lambda_{ij,k}^c \, (\boldsymbol{\xi}_1^k \boldsymbol{\xi}_2^k \boldsymbol{\xi}_3^k), \quad w_{ij}^I = \sum_{k=1}^{8} \lambda_{ij,k}^s \, (\boldsymbol{\gamma}_1^k \boldsymbol{\gamma}_2^k \boldsymbol{\gamma}_3^k) \tag{7a}$$

$$\sum_{k=1}^{8} \lambda_{ij,k}^c \boldsymbol{\xi}_1^k + (1 - z_{ij}) \underline{\boldsymbol{v}}_i \leqslant v_i \leqslant \sum_{k=1}^{8} \lambda_{ij,k}^c \boldsymbol{\xi}_1^k + (1 - z_{ij}) \overline{\boldsymbol{v}}_i \tag{7b}$$

$$\sum_{k=1}^{8} \lambda_{ij,k}^s \boldsymbol{\gamma}_1^k + (1 - z_{ij}) \underline{\boldsymbol{v}}_i \leqslant v_i \leqslant \sum_{k=1}^{8} \lambda_{ij,k}^s \boldsymbol{\gamma}_1^k + (1 - z_{ij}) \overline{\boldsymbol{v}}_i \tag{7c}$$

$$\sum_{k=1}^{8} \lambda_{ij,k}^c \boldsymbol{\xi}_2^k + (1 - z_{ij}) \underline{\boldsymbol{v}}_j \leqslant v_j \leqslant \sum_{k=1}^{8} \lambda_{ij,k}^c \boldsymbol{\xi}_2^k + (1 - z_{ij}) \overline{\boldsymbol{v}}_j \tag{7d}$$

$$\sum_{k=1}^{8} \lambda_{ij,k}^s \boldsymbol{\gamma}_2^k + (1 - z_{ij}) \underline{\boldsymbol{v}}_j \leqslant v_j \leqslant \sum_{k=1}^{8} \lambda_{ij,k}^s \boldsymbol{\gamma}_2^k + (1 - z_{ij}) \overline{\boldsymbol{v}}_j \tag{7e}$$

$$c_{ij} = \sum_{k=1}^{8} \lambda_{ij,k}^c \, \boldsymbol{\xi}_3^k, \;\; s_{ij} = \sum_{k=1}^{8} \lambda_{ij,k}^s \, \boldsymbol{\gamma}_3^k \tag{7f}$$

$$\sum_{k=1}^{8} \lambda_{ij,k}^c = z_{ij}, \;\; \sum_{k=1}^{8} \lambda_{ij,k}^s = z_{ij}, \;\; \lambda_{ij,k}^c \geqslant 0, \;\; \lambda_{ij,k}^s \geqslant 0, \;\; \forall k = 1, \ldots, 8 \tag{7g}$$

$$\begin{bmatrix} \lambda_{ij,1}^c + \lambda_{ij,2}^c - \lambda_{ij,1}^s - \lambda_{ij,2}^s \\ \lambda_{ij,3}^c + \lambda_{ij,4}^c - \lambda_{ij,3}^s - \lambda_{ij,4}^s \\ \lambda_{ij,5}^c + \lambda_{ij,6}^c - \lambda_{ij,5}^s - \lambda_{ij,6}^s \\ \lambda_{ij,7}^c + \lambda_{ij,8}^c - \lambda_{ij,7}^s - \lambda_{ij,8}^s \end{bmatrix}^\top \begin{bmatrix} \underline{\boldsymbol{v}}_i \cdot \underline{\boldsymbol{v}}_j \\ \underline{\boldsymbol{v}}_i \cdot \overline{\boldsymbol{v}}_j \\ \overline{\boldsymbol{v}}_i \cdot \underline{\boldsymbol{v}}_j \\ \overline{\boldsymbol{v}}_i \cdot \overline{\boldsymbol{v}}_j \end{bmatrix} = 0, \tag{7h}$$

where $\boldsymbol{\xi}_1^k$ is the value of v_i in the extreme point $\boldsymbol{\xi}^k$. The constants $\boldsymbol{\xi}_2^k$, $\boldsymbol{\xi}_3^k$, $\boldsymbol{\gamma}_1^k$, $\boldsymbol{\gamma}_2^k$, and $\boldsymbol{\gamma}_3^k$ are similarly defined. $\lambda_{ij,k}^c$ and $\lambda_{ij,k}^s$ are auxiliary multiplier variables for representing a linear

combination of the extreme points. When $z_{ij} = 1$, constraints (7a) connect values of w_{ij}^R and w_{ij}^I with convex combinations of extreme points for trilinear terms; constraints (7b) - (7f) equate the values of v_i, v_j, c_{ij}, and s_{ij} to convex combinations of their respective extreme points in $[\underline{\boldsymbol{v}}_i, \overline{\boldsymbol{v}}_i] \times [\underline{\boldsymbol{v}}_j, \overline{\boldsymbol{v}}_j] \times [\underline{\boldsymbol{c}}_{ij}, \overline{\boldsymbol{c}}_{ij}]$. When $z_{ij} = 0$, constraints (7a) - (7f) enforce $w_{ij}^R = w_{ij}^I = c_{ij} = s_{ij} = 0$, and impose no constraints on v_i and v_j. Constraints (7g) ensure that the summations of convex combination coefficients equal to 1 when line (i, j) is switched on, and all the coefficients become 0 when the line is switched off. Linking constraints (7h) connect the shared bilinear term $v_i v_j$ that appears in both trilinear terms $v_i v_j \cos(\theta_{ij})$ and $v_i v_j \sin(\theta_{ij})$.

Note that when $z_{ij} = 1$, constraints (7b) - (7e) and (7g) reduce to the following constraints, akin to the ones derived by Lu et al. (2018) and Sundar et al. (2018):

$$v_i = \sum_{k=1}^{8} \lambda_{ij,k}^c \boldsymbol{\xi}_1^k = \sum_{k=1}^{8} \lambda_{ij,k}^s \boldsymbol{\gamma}_1^k, \tag{8a}$$

$$v_j = \sum_{k=1}^{8} \lambda_{ij,k}^c \boldsymbol{\xi}_2^k = \sum_{k=1}^{8} \lambda_{ij,k}^s \boldsymbol{\gamma}_2^k, \tag{8b}$$

$$\sum_{k=1}^{8} \lambda_{ij,k}^c = 1, \ \sum_{k=1}^{8} \lambda_{ij,k}^s = 1. \tag{8c}$$

We next formally show that the constraints in (7) form the *tightest, or the convex hull*, relaxation for the summation of nonlinear terms of the form

$$z_{ij} \left(\boldsymbol{a}_1 v_i v_j c_{ij} + \boldsymbol{a}_2 v_i v_j s_{ij} \right) \quad \forall (i, j) \in \mathcal{A} \cup \mathcal{A}^R. \tag{9}$$

Note that this form appears in constraints (1c) and (1d), where $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ represent coefficients that are functions of $\boldsymbol{g}_{ij}$, $\boldsymbol{b}_{ij}$, $\boldsymbol{t}_{ij}^R$, and $\boldsymbol{t}_{ij}^I$. For this purpose, we define the following: Let $\eta = (w_{ij}^R,\ w_{ij}^I,\ c_{ij},\ s_{ij},\ \lambda_{ij,1}^c,\ \ldots,\ \lambda_{ij,8}^c,\ \lambda_{ij,1}^s,\ \ldots,\ \lambda_{ij,8}^s,\ z_{ij},\ v_i,\ v_j)$ and define the set $H = \{\eta | \eta \text{ satisfies } (7), z_{ij} \in [0, 1]\}$. When $z_{ij} = 0$ and $z_{ij} = 1$, H becomes H^0 and H^1, respectively:

$$H^0 = \left\{ \eta \;\middle|\; \begin{array}{l} w_{ij}^R = w_{ij}^I = c_{ij} = s_{ij} = z_{ij} = 0 \\ \lambda_{ij,k}^c = \lambda_{ij,k}^s = 0, \forall k = 1, \cdots, 8 \\ v_i \in [\underline{\boldsymbol{v}}_i, \overline{\boldsymbol{v}}_i], v_j \in [\underline{\boldsymbol{v}}_j, \overline{\boldsymbol{v}}_j] \end{array} \right\}$$

$$H^1 = \{\eta \mid \eta \text{ satisfies (7a), (7f), (7h) and (8)}\}$$

Sundar et al. (2018) show, for a simpler case when $z_{ij}=1$, that the linearization defined by H^1 is the convex hull of the summation terms in (9) due to the addition of equality constraints (7h). However, to understand the tightness (convex hull property) of the linearization (7) for the generalized ACOTS model, we present the following theorem, based on the literature of perspective formulations for disjunctive programming (Ceria and Soares 1999, Nagarajan et al. 2019b):

THEOREM 1. $H = \text{conv}(H^0 \cup H^1)$.

Proof. First, we prove that $\text{conv}(H^0 \cup H^1) \subseteq H$. For any $\eta^0 \in H^0$, η^0 satisfies constraints in (7) when $z_{ij}=0$. Similarly, for any $\eta^1 \in H^1$, η^1 satisfies constraints in (7) when $z_{ij}=1$. Thus, $H^0 \cup H^1 \subseteq H$. Since H contains only linear constraints and is thus convex, we have $\text{conv}(H^0 \cup H^1) \subseteq H$.

Next, we prove that $H \subseteq \text{conv}(H^0 \cup H^1)$. Let $\eta^* \in H$. If $z^*_{ij}=0$, then $\eta^* \in H^0$, and if $z^*_{ij}=1$, then $\eta^* \in H^1$. When $z^*_{ij} \in (0,1)$, we define the following variables:

$$\eta^*_0 = \left(0,0,\cdots,0, \frac{v^*_i - \sum_{k=1}^8 \lambda^{c*}_{ij,k}\boldsymbol{\xi}^k_1}{1-z^*_{ij}}, \frac{v^*_j - \sum_{k=1}^8 \lambda^{c*}_{ij,k}\boldsymbol{\xi}^k_2}{1-z^*_{ij}}\right)$$

$$\eta^*_1 = \left(\frac{w^{R*}_{ij}}{z^*_{ij}}, \frac{w^{I*}_{ij}}{z^*_{ij}}, \frac{c^*_{ij}}{z^*_{ij}}, \frac{s^*_{ij}}{z^*_{ij}}, \frac{\lambda^{c*}_{ij,1}}{z^*_{ij}}, \cdots, \frac{\lambda^{c*}_{ij,8}}{z^*_{ij}}, \frac{\lambda^{s*}_{ij,1}}{z^*_{ij}}, \cdots, \frac{\lambda^{s*}_{ij,8}}{z^*_{ij}}, 1, \sum_{k=1}^8 \frac{\lambda^{c*}_{ij,k}}{z^*_{ij}}\boldsymbol{\xi}^k_1, \sum_{k=1}^8 \frac{\lambda^{c*}_{ij,k}}{z^*_{ij}}\boldsymbol{\xi}^k_2\right)$$

Next, we prove that $\eta^*_0 \in H^0$ and $\eta^*_1 \in H^1$. Because of (7b), we have $(1-z^*_{ij})\underline{\boldsymbol{v}}_i \leqslant v^*_j - \sum_{k=1}^8 \lambda^{c*}_{ij,k}\boldsymbol{\xi}^k_2 \leqslant (1-z^*_{ij})\overline{\boldsymbol{v}}_i$. Thus, $\frac{v^*_i - \sum_{k=1}^8 \lambda^{c*}_{ij,k}\boldsymbol{\xi}^k_1}{1-z^*_{ij}} \in [\underline{\boldsymbol{v}}_i, \overline{\boldsymbol{v}}_i]$. Similarly, $\frac{v^*_j - \sum_{k=1}^8 \lambda^{c*}_{ij,k}\boldsymbol{\xi}^k_2}{1-z^*_{ij}} \in [\underline{\boldsymbol{v}}_j, \overline{\boldsymbol{v}}_j]$. Therefore, $\eta^*_0 \in H^0$.

For η^*_1, $\frac{w^{R*}_{ij}}{z^*_{ij}} = \sum_{k=1}^8 \frac{\lambda^{c*}_{ij,1}}{z^*_{ij}}(\boldsymbol{\xi}^k_1, \boldsymbol{\xi}^k_2, \boldsymbol{\xi}^k_3)$ because $w^{R*}_{ij} = \sum_{k=1}^8 \lambda^{c*}_{ij,1}(\boldsymbol{\xi}^k_1, \boldsymbol{\xi}^k_2, \boldsymbol{\xi}^k_3)$ and $z^*_{ij} \in (0,1)$. Similarly, it can be proved that η^*_1 satisfies constraints (7a), (7f), (7h), and (8c). For constraint (8a), the first equation follows directly from the definition of η^*_1, and the second equality is correct because the validity of (7b) and (7c) for η^* indicates that $\sum_{k=1}^8 \lambda^{s*}_{ij,k}\boldsymbol{\gamma}^k_1 + (1-z^*_{ij})\underline{\boldsymbol{v}}_i \leqslant \sum_{k=1}^8 \lambda^{c*}_{ij,k}\boldsymbol{\xi}^k_1 + (1-z^*_{ij})\underline{\boldsymbol{v}}_i$ and $\sum_{k=1}^8 \lambda^{s*}_{ij,k}\boldsymbol{\gamma}^k_1 + (1-z^*_{ij})\overline{\boldsymbol{v}}_i \geq \sum_{k=1}^8 \lambda^{c*}_{ij,k}\boldsymbol{\xi}^k_1 + (1-z^*_{ij})\overline{\boldsymbol{v}}_i$, thus $\sum_{k=1}^8 \lambda^{c*}_{ij,k}\boldsymbol{\xi}^k_1 = \sum_{k=1}^8 \lambda^{s*}_{ij,k}\boldsymbol{\gamma}^k_1 \Rightarrow \sum_{k=1}^8 \frac{\lambda^{c*}_{ij,k}}{z^*_{ij}}\boldsymbol{\xi}^k_1 = \sum_{k=1}^8 \frac{\lambda^{s*}_{ij,k}}{z^*_{ij}}\boldsymbol{\gamma}^k_1$, which means η^*_1 satisfies the second equality in (8a). With similar arguments, η^*_1 is also feasible for (8b). Therefore, $\eta^*_1 \in H^1$.

Now note that $\eta^* = (1-z^*_{ij})\eta^*_0 + z^*_{ij}\eta^*_1$, which means $H \subseteq \text{conv}(H^0 \cup H^1)$. □

To the best of our knowledge, this is the first attempt at applying the convex hull-based extreme-point formulation for the ACOTS QC relaxation. Previous works (Hijazi et al.

2017, Lu et al. 2017, Bestuzheva et al. 2020) have utilized recursive McCormick-based relaxations which are not as tight as the above formulation, as the former relaxation when applied to (9) is not as tight as H^1.

(4) *Other valid constraints for strengthening the on/off QC relaxation* ($\forall (i,j) \in \mathcal{A}$)*:* We also add the following constraints to strengthen the on/off QC relaxation:

$$\tan(\underline{\boldsymbol{\theta}}_{ij}) w^R_{ij} \leqslant w^I_{ij} \leqslant \tan(\overline{\boldsymbol{\theta}}_{ij}) w^R_{ij} \tag{10a}$$

$$\begin{aligned} &\boldsymbol{v}^\sigma_i \boldsymbol{v}^\sigma_j (\cos(\boldsymbol{\phi}_{ij}) w^R_{ij} + \sin(\boldsymbol{\phi}_{ij}) w^I_{ij}) - \overline{\boldsymbol{v}}_j \cos(\boldsymbol{\delta}_{ij}) \boldsymbol{v}^\sigma_i w^z_{ij} \\ &- \overline{\boldsymbol{v}}_i \cos(\boldsymbol{\delta}_{ij}) \boldsymbol{v}^\sigma_i w^z_{ji} \geq \overline{\boldsymbol{v}}_i \overline{\boldsymbol{v}}_j \cos(\boldsymbol{\delta}_{ij}) (\underline{\boldsymbol{v}}_i \underline{\boldsymbol{v}}_j - \overline{\boldsymbol{v}}_i \overline{\boldsymbol{v}}_j) z_{ij} \end{aligned} \tag{10b}$$

$$\begin{aligned} &\boldsymbol{v}^\sigma_i \boldsymbol{v}^\sigma_j (\cos(\boldsymbol{\phi}_{ij}) w^R_{ij} + \sin(\boldsymbol{\phi}_{ij}) w^I_{ij}) - \underline{\boldsymbol{v}}_j \cos(\boldsymbol{\delta}_{ij}) \boldsymbol{v}^\sigma_i w^z_{ij} \\ &- \underline{\boldsymbol{v}}_i \cos(\boldsymbol{\delta}_{ij}) \boldsymbol{v}^\sigma_i w^z_{ji} \geq \underline{\boldsymbol{v}}_i \underline{\boldsymbol{v}}_j \cos(\boldsymbol{\delta}_{ij}) (\overline{\boldsymbol{v}}_i \overline{\boldsymbol{v}}_j - \underline{\boldsymbol{v}}_i \underline{\boldsymbol{v}}_j) z_{ij} \end{aligned} \tag{10c}$$

$$|S_{ij}|^2 \leqslant \frac{w_i}{|\boldsymbol{T}_{ij}|^2} l_{ij} \tag{10d}$$

$$\begin{aligned} l_{ij} = |\boldsymbol{Y}_{ij}|^2 &\left(\frac{w^z_{ij}}{|\boldsymbol{T}_{ij}|^2} + w^z_{ji} - 2(\boldsymbol{t}^R_{ij} w^R_{ij} + \boldsymbol{t}^I_{ij} w^I_{ij}) / |\boldsymbol{T}_{ij}|^2 \right) \\ &- \frac{|\boldsymbol{Y}^c_{ij}|^2}{|\boldsymbol{T}_{ij}|^2} w^z_{ij} + 2(\boldsymbol{g}^c_{ij} p_{ij} - \boldsymbol{b}^c_{ij} q_{ij}) \end{aligned} \tag{10e}$$

$$0 \leqslant l_{ij} \leqslant \bar{\boldsymbol{l}}_{ij}, \tag{10f}$$

where $\boldsymbol{v}^\sigma_i = \underline{\boldsymbol{v}}_i + \overline{\boldsymbol{v}}_i$, $\boldsymbol{\phi}_{ij} = (\overline{\boldsymbol{\theta}}_{ij} + \underline{\boldsymbol{\theta}}_{ij})/2$, and $\boldsymbol{\delta}_{ij} = (\overline{\boldsymbol{\theta}}_{ij} - \underline{\boldsymbol{\theta}}_{ij})/2$. Constraint (10a) is the phase angle difference constraint, which is a relaxation of the equality $\tan(\boldsymbol{\theta}_{ij}) = \frac{w^R_{ij}}{w^I_{ij}}$. Constraints (10b) and (10c) are the "lifted nonlinear cuts" from (Bestuzheva et al. 2020), derived using trigonometric identities. Constraints (10d) and (10e) use the relationship between current magnitude and power flow to tighten the QC relaxation. (10f) bounds the squared current magnitude. Note that while (10a), (10d), and (10f) are the same as their counterparts in the ACOPF model, they are still valid for the ACOTS setting. On the other hand, (10b), (10c), and (10e) are modified under the ACOTS case, so that when line (i,j) is switched off, constraints (10b) and (10c) become redundant, while constraint (10e) ensures $l_{ij} = 0$. The use of (10e) is new for the ACOTS QC relaxation.

Putting all the constraints together, we obtain the following QC relaxation for the ACOTS problem:

$$(\textbf{ACOTS-QC}): \min \quad (1a) \tag{11a}$$

$$\text{s.t.} \quad (1b), (1g) - (1m), (2), \tag{11b}$$

$$(4), (5) - (7), (10). \tag{11c}$$

The relationship between the solution sets of different formulations is simplified and shown in Figure 1. Here, ACOPF is the non-convex polar formulation, which is equivalent to the ACOTS model with all the lines switched on; ACOPF-QC is the QC relaxation for ACOPF from (Coffrin et al. 2016).

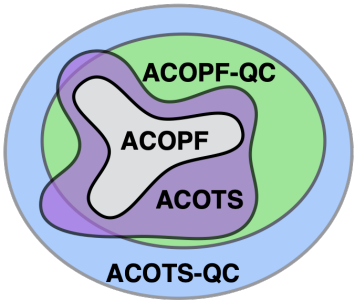

Figure 1 **Venn diagram for solution sets of different formulations.**

Tight convex relaxations for ACOTS such as (ACOTS-QC) can serve several purposes. The solution to a tight relaxation problem is a good approximation for optimal ACOTS decisions. Also, they provide tight lower bounds for evaluating the quality of a feasible ACOTS solution, and can be used to prove global optimality of a feasible solution when the optimality gap is 0. In addition, the convex relaxation can be used in a two-stage process for minimizing the cost of ACOPF, where we first change the network topology using line switching solutions from the relaxation, and then solve the ACOPF problem on the updated network. In what follows, we show how to further tighten (ACOTS-QC) with lifted cycle constraints and OBBT.

4. Cycle-Based On/Off Polynomial Constraints

In this section, we present a novel type of lifted cycle constraints based on lifted trigonometric auxiliary variables c_{ij} and s_{ij}. We use both these new lifted cycle constraints and lifted cycle constraints with voltage product variables w^R_{ij}, w^I_{ij}, and w_i from (Kocuk et al. 2016a) to strengthen the on/off QC relaxation for ACOTS. Since those constraints are polynomial functions in multilinear terms, we linearize them with the extreme-point representation. We also develp novel cutting planes to incorporate those constraints more efficiently via branch-and-cut framework.

4.1. Formulating Lifted Cycle Constraints

The lifted cycle constraints are formulated based on the fact that for any given cycle $\mathcal{C}$ in the transmission network, the voltage angle differences of all lines in $\mathcal{C}$ sum up to 0. More formally, let $(\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_n, \hat{v}_1)$ be a vertex sequence for the cycle $\mathcal{C}$ of length n, and let the cycle be represented by its lines: $\mathcal{C} = \{(\hat{v}_1, \hat{v}_2), (\hat{v}_2, \hat{v}_3), \ldots, (\hat{v}_n, \hat{v}_1)\}$, then we have $\sum_{(i,j)\in\mathcal{C}} \theta_{ij} = 0$.

The method we use to derive lifted cycle constraints for the QC relaxation is similar to that of (Kocuk et al. 2016a) where lifted cycle constraints for the SOC relaxation are derived. However, unlike in the SOC relaxation, the main advantage of the QC relaxation is that we have direct access to both (c_{ij}, s_{ij}) and (w^R_{ij}, w^I_{ij}) variables, enabling us to formulate additional lifted cycle constraints that could further enhance the relaxation quality.

In what follows, we derive lifted cycle constraints for cycles with 3 and 4 nodes. We call the resulting constraints as 3-cycle constraints and 4-cycle constraints, respectively. We first develop those constraints without considering switching (on/off) decisions, and then demonstrate how to reformulate them to include those decisions using a tight big-M formulation.

4.1.1. 3-Cycle Constraints For a cycle of three nodes i, j and k, we have $\theta_{ij} + \theta_{jk} + \theta_{ki} = 0$ or equivalently $\theta_{ik} = \theta_{ij} + \theta_{jk}$, which indicates that $\cos(\theta_{ik}) = \cos(\theta_{ij} + \theta_{jk})$ and $\sin(\theta_{ik}) = \sin(\theta_{ij} + \theta_{jk})$. Expanding the right-hand sides and replacing the trigonometric functions with their corresponding lifted variables, we get the following nonlinear 3-cycle constraints:

$$c_{ik} = c_{ij}c_{jk} - s_{ij}s_{jk} \tag{12a}$$

$$s_{ik} = c_{ij}s_{jk} + s_{ij}c_{jk}, \tag{12b}$$

Though simple, these are novel and are applicable to both ACOPF and ACOTS problems. We can then obtain the lifted cycle constraints in (Kocuk et al. 2016a) by multiplying both sides of (12a) and (12b) with $v_i v_j^2 v_k$, and using the relationships in (1e) and (3) (ignoring switching decisions in (3) for now):

$$w_j w^R_{ik} = w^R_{ij} w^R_{jk} - w^I_{ij} w^I_{jk}, \tag{13a}$$

$$w_j w^I_{ik} = w^R_{ij} w^I_{jk} + w^I_{ij} w^R_{jk}. \tag{13b}$$

Alternatively, constraints (13) can also be derived from minor-based reformulation as in (Kocuk et al. 2018).

For brevity, in the following, we call lifted cycle constraints with c_{ij} and s_{ij} variables as lifted cycle constraints *in the c-s space*, and lifted cycle constraints with w^R_{ij}, w^I_{ij}, and w_i variables as lifted cycle constraints *in the w space*.

We also derive two more sets of lifted cycle constraints by considering other permutations of the equality $\theta_{ij}+\theta_{jk}+\theta_{ki}=0$, including $\theta_{jk}+\theta_{ki}=\theta_{ji}$ and $\theta_{ki}+\theta_{ij}=\theta_{kj}$. Though they look equivalent in the nonlinear forms, the linearized versions (see Section 4.2) of these permutated constraints do not necessarily dominate one another, and we add all of them to tighten the relaxation.

Another type of lifted cycle constraints can be derived from the equality $\theta_{ij}+\theta_{jk}-\theta_{ik}=0$, which leads to the following constraints in the *c-s* space:

$$c_{ij}c_{jk}c_{ik}+c_{ij}s_{jk}s_{ik}-s_{ij}s_{jk}c_{ik}+s_{ij}c_{jk}s_{ik}=1 \tag{14a}$$

$$s_{ij}c_{jk}c_{ik}+s_{ij}s_{jk}s_{ik}+c_{ij}s_{jk}c_{ik}-c_{ij}c_{jk}s_{ik}=0. \tag{14b}$$

The following proposition shows the equivalence between constraints (12) and (14). This proposition is similar to the Proposition 4.1 in (Kocuk et al. 2016a), but our result is in the *c-s* space rather than the *w* space, and we simplify the presentation of the result. Also, we provide a new way to prove this result.

PROPOSITION 1. *For all* $(i,j)\in\mathcal{A}$, *if* c_{ij} *and* s_{ij} *satisfy* $c_{ij}^2+s_{ij}^2=1$, *then* $\{(c,s):(12)\ holds\}=\{(c,s):(14)\ holds\}$.

Proof. We first prove that $\{(c,s):(12)\text{ holds}\}\subseteq\{(c,s):(14)\text{ holds}\}$. If the variables c and s satisfy (12a)-(12b), we multiply both sides of (12a) with c_{ik} and both sides of (12b) with s_{ik}, then sum up those two equations:

$$c_{ik}^2+s_{ik}^2=c_{ik}c_{ij}c_{jk}-c_{ik}s_{ij}s_{jk}+s_{ik}c_{ij}s_{jk}+s_{ik}s_{ij}c_{jk}.$$

Since the left-hand side is equal to 1, this equation is equivalent to constraint (14a). Similarly, we obtain constraint (14b) by multiplying both sides of (12a) with s_{ik} and both sides of (12b) with c_{ik}, and deduct the second equation from the first.

For the reverse direction, if c and s satisfy (14), let $a=c_{ij}c_{jk}-s_{ij}s_{jk}$ and $b=c_{ij}s_{jk}+s_{ij}c_{jk}$, we can rewrite (14a) and (14b) as $c_{ik}a+s_{ik}b=1$ and $c_{ik}b-s_{ik}a=0$, respectively. Solving for a and b, we get $a=c_{ik}$ and $b=s_{ik}$, which are equivalent to constraints (12a)-(12b). □

4.1.2. 4-Cycle Constraints For a 4-cycle with nodes $\{i, j, k, l\}$, we similarly derive lifted cycle constraints based on the equality $\theta_{ij} + \theta_{jk} + \theta_{kl} + \theta_{li} = 0$. More specifically, for the permutation $\theta_{ij} + \theta_{kl} = \theta_{il} - \theta_{jk}$, we derive the following lifted cycle constraints

$$c_{ij}c_{kl} - s_{ij}s_{kl} = c_{il}c_{jk} + s_{il}s_{jk} \tag{15a}$$

$$c_{ij}s_{kl} + s_{ij}c_{kl} = -c_{il}s_{jk} + s_{il}c_{jk} \tag{15b}$$

$$w^R_{ij}w^R_{kl} - w^I_{ij}w^I_{kl} = w^R_{il}w^R_{jk} + w^I_{il}w^I_{jk} \tag{15c}$$

$$w^R_{ij}w^I_{kl} + w^I_{ij}w^R_{kl} = -w^R_{il}w^I_{jk} + w^I_{il}w^R_{jk}. \tag{15d}$$

For the other two permutations, i.e., $\theta_{ij} + \theta_{jk} = \theta_{il} - \theta_{kl}$ and $\theta_{jk} + \theta_{kl} = \theta_{il} - \theta_{ij}$, we can derive similar lifted cycle constraints. Note that for those permutations, the lifted cycle constraints in the w space contain trilinear terms, thus we do not include those constraints in our implementation for efficiency purposes.

4.1.3. On/off cycle constraints for ACOTS To reformulate lifted cycle constraints for ACOTS and include switching decisions, we use the big-M formulation. As an example, we demonstrate the formulation on constraints (12), i.e., the 3-cycle constraints in the c-s space. Constraints (12) are only valid when all lines in the 3-cycle $\mathcal{C}$ are switched on, which is ensured by the following big-M constraints:

$$-3\widehat{z} \leqslant c_{ik} - c_{ij}c_{jk} + s_{ij}s_{jk} \leqslant 3\widehat{z} \tag{16a}$$

$$-3\widehat{z} \leqslant s_{ik} - c_{ij}s_{jk} - s_{ij}c_{jk} \leqslant 3\widehat{z}. \tag{16b}$$

Here, $\widehat{z} = \sum_{(l,m)\in\mathcal{C}}(1 - z_{lm})$ and we use "3" as the big-M constant. It is valid because $c_{ij} \in [0, 1]$ and $s_{ij} \in [-1, 1]$ for the worst-case bounds of $\theta_{ij} \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Although, this could be further improved if the angle-difference bounds are tighter. We also include similar on/off constraints for all 4-cycles with appropriate big-M constants.

4.2. Extreme-Point Representation

The lifted cycle constraints contain bilinear terms, which are usually linearized with McCormick relaxation in the literature (Kocuk et al. 2016b, 2018). We instead use the extreme-point representation to linearize those constraints, which is *guaranteed to capture the convex hull* of the lifted cycle constraints for a given cycle (including all permutations in the c-s or w space).

For example, let $x_i^{\mathbf{c}}\ \forall i=1,\ldots,6$ represent variables $c_{ij}, c_{jk}, c_{ik}, s_{ij}, s_{jk}, s_{ik}$, respectively. We first rewrite 3-cycle constraints (12) and its counterparts by permutation as follows:

$$x_3^{\mathbf{c}} = x_1^{\mathbf{c}}x_2^{\mathbf{c}} - x_4^{\mathbf{c}}x_5^{\mathbf{c}}, \quad x_6^{\mathbf{c}} = x_1^{\mathbf{c}}x_5^{\mathbf{c}} + x_2^{\mathbf{c}}x_4^{\mathbf{c}} \tag{17a}$$

$$x_1^{\mathbf{c}} = x_2^{\mathbf{c}}x_3^{\mathbf{c}} + x_5^{\mathbf{c}}x_6^{\mathbf{c}}, \quad x_4^{\mathbf{c}} = x_2^{\mathbf{c}}x_6^{\mathbf{c}} - x_3^{\mathbf{c}}x_5^{\mathbf{c}} \tag{17b}$$

$$x_2^{\mathbf{c}} = x_1^{\mathbf{c}}x_3^{\mathbf{c}} + x_4^{\mathbf{c}}x_6^{\mathbf{c}}, \quad x_5^{\mathbf{c}} = x_1^{\mathbf{c}}x_6^{\mathbf{c}} - x_3^{\mathbf{c}}x_4^{\mathbf{c}}. \tag{17c}$$

Let binary variable $y_{\mathcal{C}}$ equal 1 if and only if all lines in the cycle $\mathcal{C}=\{(i,j),(j,k),(k,i)\}$ are switched on. $x_{j_1j_2}^{\mathbf{c}}$ is a lifted variable for $x_{j_1}^{\mathbf{c}}x_{j_2}^{\mathbf{c}}$. We can linearize the constraint $x_3^{\mathbf{c}} = x_1^{\mathbf{c}}x_2^{\mathbf{c}} - x_4^{\mathbf{c}}x_5^{\mathbf{c}}$ and connect the constraint with line switching decisions as follows:

$$\min(0,\underline{\boldsymbol{x}}_3^{\mathbf{c}})(1-y_{\mathcal{C}}) \leqslant x_3^{\mathbf{c}} - x_{12}^{\mathbf{c}} + x_{45}^{\mathbf{c}} \leqslant \max(0,\overline{\boldsymbol{x}}_3^{\mathbf{c}})(1-y_{\mathcal{C}}). \tag{18}$$

We will explain this constraint in more detail at the end of this section after introducing constraints (19). Other constraints in (17) can be linearized in a similar way.

In addition to the linearization above, we have the following constraints in the extreme-point representation for constraints (17) (with switching decisions added):

$$\sum_{i=1}^{64} \lambda_i^{cs} = y_{\mathcal{C}} \tag{19a}$$

$$\lambda_i^{cs} \geqslant 0 \qquad \forall i=1,\ldots,64 \tag{19b}$$

$$x_j^{\mathbf{c}} \geq \underline{\boldsymbol{x}}_j^{\mathbf{c}} \left(\sum_{i:(\mathcal{X}_j^i = \underline{\boldsymbol{x}}_j^{\mathbf{c}})} \lambda_i^{cs} \right) + \overline{\boldsymbol{x}}_j^{\mathbf{c}} \left(\sum_{i:(\mathcal{X}_j^i = \overline{\boldsymbol{x}}_j^{\mathbf{c}})} \lambda_i^{cs} \right) + \min(0,\underline{\boldsymbol{x}}_j^{\mathbf{c}})(1-y_{\mathcal{C}}) \qquad \forall j=1,\ldots,6 \tag{19c}$$

$$x_j^{\mathbf{c}} \leqslant \underline{\boldsymbol{x}}_j^{\mathbf{c}} \left(\sum_{i:(\mathcal{X}_j^i = \underline{\boldsymbol{x}}_j^{\mathbf{c}})} \lambda_i^{cs} \right) + \overline{\boldsymbol{x}}_j^{\mathbf{c}} \left(\sum_{i:(\mathcal{X}_j^i = \overline{\boldsymbol{x}}_j^{\mathbf{c}})} \lambda_i^{cs} \right) + \max(0,\overline{\boldsymbol{x}}_j^{\mathbf{c}})(1-y_{\mathcal{C}}) \qquad \forall j=1,\ldots,6 \tag{19d}$$

$$x_{j_1j_2}^{\mathbf{c}} = \sum_{i=1}^{64} \lambda_i^{cs} \left(\mathcal{X}_{j_1}^i \mathcal{X}_{j_2}^i \right) \qquad \forall (j_1,j_2) \in \mathcal{P} \tag{19e}$$

$$1 - \sum_{(i,j)\in\mathcal{C}} (1-z_{ij}) \leqslant y_{\mathcal{C}} \leqslant \frac{1}{|\mathcal{C}|} \sum_{(i,j)\in\mathcal{C}} z_{ij} \tag{19f}$$

$$y_{\mathcal{C}} \in \{0,1\} \tag{19g}$$

where λ_i^{cs} is an auxiliary variable. $\mathcal{P}=\{(1,2),(1,3),(1,5),(1,6),(2,3),(2,4),(2,6),(3,4),(3,5),(4,5),(4,6),(5,6)\}$. $\mathcal{X}$ can be viewed as a matrix of size $2^6\times 6$, such that every row represents all possible combinations of the lower and upper bounds of variables $x_j^{\mathbf{c}} \in$

$[\underline{\boldsymbol{x}}_j^{\mathbf{c}}, \overline{\boldsymbol{x}}_j^{\mathbf{c}}]$ $\forall j = 1, \ldots, 6$. Constraints (19a) and (19b) set bounds for auxiliary multiplier variables. When $y_{\mathcal{C}} = 1$, constraints (19c), (19d), and (19e) represent the convex hull consisting of variables $x_j^{\mathbf{c}}$ $(\forall j)$ and $x_{j_1 j_2}^{\mathbf{c}}$ $(\forall (j_1, j_2))$; when $y_{\mathcal{C}} = 0$ constraints (19c) and (19d) become redundant. Constraint (19f) connects $y_{\mathcal{C}}$ and z_{ij}: when all lines are switched on, (19f) fixes $y_{\mathcal{C}}$ to 1. If any line is switched off, $1 - \sum_{(i,j)\in\mathcal{C}}(1 - z_{ij}) \leqslant 0$ and $\frac{1}{|\mathcal{C}|}\sum_{(i,j)\in\mathcal{C}} z_{ij} \in [0, 1)$, which enforce $y_{\mathcal{C}} = 0$. Also note that when $y_{\mathcal{C}} = 0$, (19a), (19b) and (19e) ensure $x_{12}^{\mathbf{c}} = x_{45}^{\mathbf{c}} = 0$, and constraint (18) becomes redundant. We can similarly derive the convex hull formulation for 3-cycle constraints in the w space and for 4-cycle constraints.

To show the tightness of extreme-point formulation compared with McCormick relaxation, we use the scatter plot method, similar to that of Luedtke et al. (2012). We apply the two different relaxation methods for the summation of bilinear terms $\sum_{(j_1,j_2)\in\mathcal{P}} x_{j_1}^{\mathbf{c}} x_{j_2}^{\mathbf{c}}$. Without loss of generality, we set the domain of $x_i^{\mathbf{c}}, i = 1, ..., 6$ as $[-1, 1]$, which include both positive and negative numbers. We randomly generate 5,000 samples of those points, following a uniform distribution in their domain. For each sample, we then obtain the difference between upper and lower bounds of the summation with the two relaxation methods, and obtain the scatter plot in Figure 2. In the scatter plot, each point (i.e., blue spot) corresponds to the result of one sample. All the points are above the (grey) diagonal line, which implies that the extreme-point representation is either tighter or as tight as the McCormick relaxation.

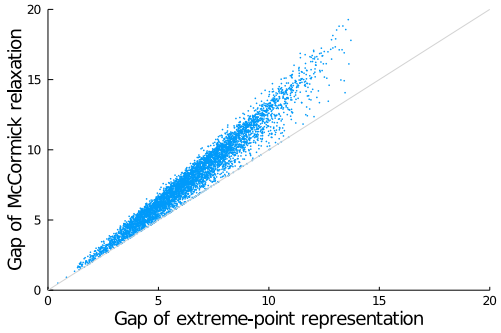

Figure 2 **Scatter plot comparing extreme-point formulation with McCormick relaxation for summation of bilinear terms.**

4.3. Branch-and-Cut Algorithm for Lifted Cycle Constraints

The size of extreme-point formulation for lifted cycle constraints grows quickly with the number of cycles in the network. Therefore, instead of adding all of those constraints at

once, we use a separation scheme which generates cutting planes only when the linearized lifted cycle constraints are violated.

Due to binary line switching decisions in the linearized lifted cycle constraints, the separation problem is not a linear program (LP). To generate cutting planes that separates infeasible solutions, we first ignore the line switching decisions in the linearized lifted cycle constraints and generate Benders feasibility cuts if those constraints are violated. We then incorporate binary variables into the Benders cuts via disjunctive programming.

First, we describe how to generate Benders cuts without considering line switching decisions. At the start of the cutting-plane algorithm we solve the ACOTS-QC model without any lifted cycle constraints, and obtain optimal solutions for c-s and w variables. Then for each cycle we solve a feasibility problem consisting of all the lifted cycle constraints (within the c-s or w space) in the extreme-point formulation without line switching decisions, while fixing c-s or w variables to their optimal values. If this feasibility problem is feasible, then none of the linearized lifted cycle constraints is violated, so we do not need to add any cut; otherwise, we generate a Benders feasibility cut, and add this cut back to the ACOTS-QC model and solve it again. The algorithm terminates if at one iteration the ACOTS-QC model solution is feasible to the lifted cycle constraints for all cycles.

For example, for a 3-cycle with nodes i, j, and k, let $x^{\mathbf{c}} = (c_{ij}, c_{jk}, c_{ik}, s_{ij}, s_{jk}, s_{ik})$ and let $\boldsymbol{x}^{\mathbf{c}*}$ be an optimal solution of the ACOTS-QC model. We solve the following separation problem:

$$\begin{aligned} \min \quad & 0 && \text{(20a)} \\ \text{s.t.} \quad & \text{extreme-point representation of (17)} && \text{(20b)} \\ & x^{\mathbf{c}} = \boldsymbol{x}^{\mathbf{c}*}. && \text{(20c)} \end{aligned}$$

If this problem is infeasible, we can generate a Benders feasibility cut in the following form:

$$\boldsymbol{\beta}^\top x^{\mathbf{c}} \leq \boldsymbol{b} \tag{21}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{b}$ are the coefficient vector and the constant in the Benders cut, respectively. Now we consider the impact of line switching decisions. The Benders cut should be redundant when any line in a cycle is turned off. To ensure this, we reformulate Benders cuts via disjunctive programming. Again, we use the 3-cycle constraints in the c-s space to

demonstrate how this works. Remember the binary variable $y_{\mathcal{C}}$ that equals 1 if and only if all lines in the cycle $\mathcal{C}$ are turned on. The Benders cut (21) should only be active when $y_{\mathcal{C}} = 1$. In other words, the feasible region defined by the reformulated Benders cut is a union of the following two sets:

$$\Gamma_0 = \{(x^{\mathbf{c}}, y_{\mathcal{C}}) : y_{\mathcal{C}} = 0,\ \ \min(0, \underline{\boldsymbol{x}}^{\mathbf{c}}) \le x^{\mathbf{c}} \le \max(0, \overline{\boldsymbol{x}}^{\mathbf{c}})\}$$
$$\Gamma_1 = \{(x^{\mathbf{c}}, y_{\mathcal{C}}) : y_{\mathcal{C}} = 1,\ \ \boldsymbol{\beta}^\top x^{\mathbf{c}} \le \boldsymbol{b}\}$$

where $\underline{\boldsymbol{x}}^{\mathbf{c}}$ and $\overline{\boldsymbol{x}}^{\mathbf{c}}$ are lower and upper bounds of $x^{\mathbf{c}}$. Using disjunctive programming, the following constraint is valid for the convex hull of $\Gamma_0 \cup \Gamma_1$:

$$\boldsymbol{\beta}^\top x^{\mathbf{c}} \le y_{\mathcal{C}} \boldsymbol{b} + (1 - y_{\mathcal{C}}) \left(\sum_{i \in \mathcal{N}: \boldsymbol{\beta}_i < 0} \boldsymbol{\beta}_i \min(0, \underline{\boldsymbol{x}}_i^{\mathbf{c}}) + \sum_{i \in \mathcal{N}: \boldsymbol{\beta}_i > 0} \boldsymbol{\beta}_i \max(0, \overline{\boldsymbol{x}}_i^{\mathbf{c}}) \right) \tag{22}$$

where $\mathcal{N}$ is the set of indices for $\boldsymbol{\beta}$ and $x^{\mathbf{c}}$. Intuitively, when $y_{\mathcal{C}} = 1$, (22) is exactly the Benders cut (21); when $y_{\mathcal{C}} = 0$, (22) is always satisfied and thus becomes redundant.

The separation scheme needs to solve the mixed-integer quadratic ACOTS-QC model at each iteration, which is very inefficient. This is why we use a branch-and-cut method, where the mixed-integer quadratic ACOTS-QC model is only solved once with the branch-and-bound algorithm, and the Benders cuts are added at integral nodes of the branch-and-bound tree. More specifically, at each integral node, we obtain the optimal values of trigonometric terms $x^{\mathbf{c}}$ and switching decisions z_{ij}, and denote them respectively by $\boldsymbol{x}^{\mathbf{c}*}$ and $\mathbf{z}_{ij}^*$. For any cycle $\mathcal{C}$ with all lines turned on (i.e., when $\sum_{(i,j) \in \mathcal{C}} \mathbf{z}_{ij}^* = |\mathcal{C}|$), we solve the separation problem (20). If the problem is infeasible, we add constraints (19f), (19g), and (22).

5. Optimization-Based Bound Tightening

The OBBT method is a technique in non-convex optimization, which aims to improve the convex relaxation bound by tightening the bounds of certain variables. OBBT is often used to improve bounds in AC power flow problems (Chen et al. 2015), and it has the benefit of being massively parallelizable (Gopinath et al. 2020). In our work, we implement OBBT to tighten the bounds of v_i, θ_{ij}, z_{ij}, and y_C variables before solving the relaxations of ACOTS.

To formulate bound tightening optimization models for any variable x, we replace the objective of an ACOTS relaxation (e.g., ACOTS-QC) with $\max x$ or $\min x$. To avoid solving time-consuming MINLPs in OBBT, we linearly relax all integer variables. We denote the

optimal objectives of the bound tightening maximization and minimization problems $\bar{x}$ and $\underline{x}$. If x is a binary variable (such as z_{ij} and y_C), we can further tighten their bounds by fixing x to 1 if $\underline{x} > 0$, and to 0 if $\bar{x} < 1$ within the OBBT iteration. The OBBT algorithm terminates when the bounds of all variables stop improving, or when the algorithm reaches its time/iteration limit.

We summarize the implementation of the algorithm in Figure 3.

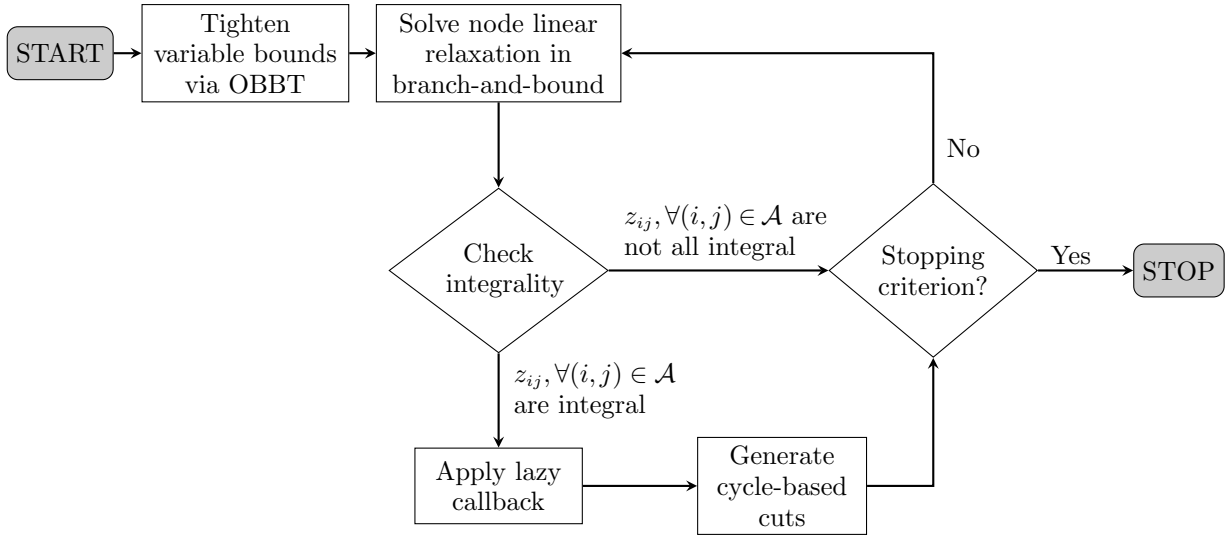

Figure 3 **Flow chart of the proposed algorithm. After OBBT which preprocesses variable bounds, the algorithm enters a branch-and-bound tree where cuts are generated (as described in Section 4.3) and added at integral nodes via lazy callback. The algorithm terminates when the gap between the upper and lower bounds of branch-and-bound search is below a small tolerance.**

6. Numerical Experiments

This section presents the numerical efficacy of the proposed ACOTS-QC and ACOPF-QC relaxations with lifted cycle constraints, and an analysis for ACOTS with different load profiles. Our experiments are conducted on PGLib-OPF v20.07 benchmark library (Babaeinejadsarookolaee et al. 2019). We use a Linux workstation with 3.6GHz Intel Core i9-9900K CPUs and 128GB memory. The programming language is Julia v1.6. We locally solve all non-convex MINLP (ACOTS) and NLP (ACOPF) formulations using Juniper.jl (v0.7.0) (Kroger et al. 2018) and Ipopt (v3.13.4) (Wachter and Biegler 2006), respectively. All relaxation formulations (ACOTS-QC, ACOPF-QC and OBBT iterations) are solved using the Gurobi (v9.0.0) solver. The branch-and-cut framework for cycle constraints is implemented using Gurobi's lazy-constraint callback.

6.1. Relaxations for ACOTS

We compare five different types of relaxations for ACOTS:

(1) "PM": The on/off QC relaxation implemented in PowerModels.jl (Coffrin et al. 2018), which is used as state-of-the-art to benchmark ACOTS relaxations. Formulation within "PM" is based on (Hijazi et al. 2017) which uses on/off trigonometric function relaxations and recursive McCormick linearization of trilinear terms, without additional cycle constraints or the OBBT algorithm.

(2) "E": Proposed ACOTS-QC relaxation with extreme-point representation for linearizing $z_{ij}v_iv_jc_{ij}$ and $z_{ij}v_iv_js_{ij}$ in (3).

(3) "EC": Tightened "E" with lifted cycle constraints.

(4) "ECB": Includes all proposed improvements (extreme-point representation, lifted cycle constraints, and OBBT).

(5) "ECB*": The same as "ECB", except the lifted cycle constraints are added via branch-and-cut framework as in Section 4.3.

We also provide initial feasible solutions as *warm-start* solutions, which are helpful to speedup the convergence for many of the large instances. Those initial feasible solutions are obtained by solving ACOPF-QC relaxation with recursive McCormick linearization for trilinear terms (for PM), ACOPF-QC (with extreme-point linearization, for "E") or ACOPF-QC with lifted cycle constraints (for "EC"), and those solutions are valid when all lines in the network are switched on.

We run PGLib instances with up to 300 buses under typical operating conditions (TYP), as well as cases with small angle-difference conditions (SAD) and congested operating conditions (API). In Table 2 we present results for cases that are solved in the 2-hour time limit (within 0.1% optimality tolerance) for "E". The performance measures we use for comparison include optimality gap and runtime. We put "ns." for the optimality gaps of cases that are not solved to 0.1% optimality tolerance within the time limit, and "tl." for the runtime of test cases that hit the time limit.

The optimality gap is calculated by (UB - LB)/LB*100 where LB is the optimal value from relaxations of ACOTS, and UB is an upper bound for ACOTS. For UB, we take the minimum of *local optimal values* of the following three types of upper bounding models:

- Non-convex ACOTS model (1). Here, note that within the time limit (2 hours), Juniper fails to find feasible solutions for many instances.

- Non-convex ACOPF model with all lines switched on.
- Non-convex ACOPF model with the set of lines switched off, as indicated by the ACOTS-QC solutions.

We highlight with boldface the optimality gaps improved after the relaxations are tightened. All comparisons are between two adjacent columns in the table. We also highlight the reduced runtimes of our branch-and-cut algorithm (in ECB*).

The runtimes of "ECB", and "ECB*" are the runtimes of the ACOTS-QC relaxation problems and do not include the runtimes of OBBT. This is because OBBT time is a constant factor inclusion irrespective of whether the cycle constraints are added to LP-relaxed models ($z_{ij}, y_C \in [0,1]$), directly or in a branch-and-cut fashion within the OBBT algorithm. Moreover, these times are not as significant when compared with the ACOTS-QC relaxation problems, as the OBBT's LP-relaxed models, at every iteration, can be solved in parallel. We also exclude the model building time of sub-problems in ECB* within the branch-and-cut algorithm, as any overhead in such time is an artifact of the mathematical modeling package within Julia. In addition, we set the upper bound on the number of added cuts at 200, as adding too many cuts could slow down the performance. We observe that those added cuts are able to significantly improve the bounds as shown in Table 2.

In Table 2, compared with "PM", our tightened "E" reduces the optimality gap for many benchmark instances, especially for the SAD and API ones. For example, it yields 3.4% gap improvement for case3_lmbd_api and 2.9% improvement for case24_ieee_rts_sad. It also solves several instances to optimality that "PM" is not able to solve within the time limit. The benefit of the lifted cycle constraints ("EC") is most apparent for "SAD" , with case14_ieee_sad closing 6.2% and case89_pegase_sad finding the global optimal solution.

Combining the extreme point formulation, OBBT algorithm, and the lifted cycle constraints, *we obtain the tightest QC-based ACOTS relaxation in the literature*, as highlighted in the optimality gap columns of "ECB" and "ECB*" (see Table 2). Note that "ECB*" is as tight as "ECB" in almost all cases, and reduces the solution time significantly in many instances due to the efficient implementation of the branch-and-cut framework. It is clear from the table that the OBBT algorithm in conjunction with all the proposed enhancements in this paper can provide significant improvements in closing the gap for several benchmark cases. For example, in case14_ieee_sad, "ECB" closes as much as 17.6%

Table 2 Optimality Gap and Runtime of ACOTS Relaxations (bold numbers: improved gaps and run times after tightening the relaxation; 'ns.": not solved to optimality tolerance within time limit; "tl.": hits the time limit)

		Optimality Gap (%)					Runtime (seconds)				
Test Case	UB	PM	E	EC	ECB	ECB*	PM	E	EC	ECB	ECB*
Typical Operaing Conditions (TYP)											
case3_lmbd	5812.6	1.3	**1.0**	1.0	**0.0**	0.1	0.02	0.04	0.19	0.15	0.39
case5_pjm	15174.0	1.1	1.1	1.1	1.1	1.1	0.12	0.04	0.23	0.31	0.43
case14_ieee	2178.1	0.1	0.1	0.1	0.1	0.1	0.44	0.28	1.49	2.50	**1.32**
case24_ieee_rts	63352.2	0.0	0.0	0.0	0.0	0.0	4.28	1.11	310.56	349.21	**3.37**
case30_as	803.1	0.1	0.1	0.1	0.1	0.1	6.21	21.97	446.96	630.60	**9.97**
case30_ieee	7579.0	12.1	**11.9**	11.9	**11.0**	11.0	1.21	1.01	3.30	6.56	**4.79**
case39_epri	137728.7	0.0	0.0	0.0	0.0	0.0	0.65	0.52	0.93	1.17	2.32
case57_ieee	37559.3	0.1	0.1	0.1	0.1	0.1	34.34	17.51	40.27	77.43	**27.96**
case73_ieee_rts	189764.1	0.0	0.0	0.0	0.0	0.0	39.58	33.14	4876.16	5136.27	**34.48**
case89_pegase	106622.2	0.1	0.1	**0.0**	ns.	ns.	tl.	tl.	tl.	tl.	tl.
case118_ieee	96645.9	0.3	0.3	0.3	0.3	0.3	497.58	415.09	1622.20	1778.12	**1389.84**
case179_goc	754266.4	0.2	0.2	0.2	0.2	0.2	1095.93	307.07	175.25	334.33	340.78
case200_activ	27557.6	0.0	0.0	0.0	0.0	ns.	1636.15	2837.87	4398.64	3014.30	tl.
Small Angle Difference Conditions (SAD)											
case3_lmbd_sad	5959.3	3.0	**1.4**	**1.3**	**0.1**	0.1	0.02	0.02	0.07	0.15	0.38
case5_pjm_sad	26108.8	1.4	**0.6**	0.6	**0.2**	0.2	0.04	0.05	0.15	0.20	0.41
case14_ieee_sad	2727.5	20.1	**18.3**	**12.1**	**0.7**	0.8	0.41	0.57	2.55	3.23	**1.15**
case24_ieee_rts_sad	75794.0	5.3	**2.4**	**2.1**	**0.7**	0.8	32.92	14.37	84.41	104.09	**16.97**
case30_as_sad	893.9	4.5	**1.9**	1.9	**1.1**	1.2	18.56	11.68	45.37	65.27	**14.28**
case30_ieee_sad	8188.6	8.8	**8.7**	8.7	**0.1**	0.2	1.50	2.82	5.62	5.89	**2.77**
case39_epri_sad	147472.8	0.1	0.1	0.1	0.1	0.1	11.84	15.59	14.39	16.40	**9.68**
case57_ieee_sad	38597.8	0.2	0.2	**0.1**	0.1	0.1	24.21	44.55	148.33	189.35	**82.79**
case89_pegase_sad	107285.7	ns.	**0.7**	**0.0**	ns.	ns.	tl.	tl.	tl.	tl.	tl.
case118_ieee_sad	97572.5	0.9	0.9	0.9	0.9	0.9	4581.28	3453.41	6818.73	tl.	**4520.33**
case179_goc_sad	755293.1	ns.	**0.1**	0.1	0.1	0.1	tl.	tl.	tl.	tl.	tl.
case200_activ_sad	27557.6	0.0	0.0	0.0	0.0	ns.	3564.09	tl.	1671.87	tl.	tl.
Congested Operating Conditions (API)											
case3_lmbd_api	10636.0	3.8	**0.4**	0.4	**0.0**	0.0	0.02	0.03	0.13	0.09	0.33
case5_pjm_api	75190.3	2.6	2.6	2.6	**0.3**	0.3	0.11	0.07	0.19	0.30	0.55
case14_ieee_api	5999.4	5.1	5.1	5.1	**0.8**	0.9	0.34	0.27	1.24	0.82	0.84
case24_ieee_rts_api	119743.1	5.2	**3.4**	3.4	**1.2**	1.2	7.30	4.16	82.80	31.66	**6.50**
case30_as_api	3065.8	9.7	9.7	9.7	**9.4**	9.5	3.65	3.44	37.99	20.32	**6.00**
case30_ieee_api	17936.5	4.9	4.9	4.9	**0.3**	0.4	1.25	0.86	2.22	4.57	**2.22**
case39_epri_api	246723.0	0.5	0.5	0.5	**0.4**	0.4	1.42	0.75	1.86	4.84	**3.77**
case57_ieee_api	49271.9	0.0	0.1	0.1	**0.0**	0.0	36.50	12.63	44.12	51.50	**27.34**
case73_ieee_rts_api	385277.3	4.3	**2.4**	**1.7**	**1.2**	1.2	128.08	3869.93	tl.	3307.26	**328.03**
case89_pegase_api	100325.3	ns.	**0.2**	**0.0**	ns.	ns.	tl.	tl.	tl.	tl.	tl.
case118_ieee_api	181535.8	6.5	**6.2**	6.2	**6.1**	6.1	tl.	tl.	tl.	tl.	tl.
case162_ieee_dtc_api	116923.8	1.0	1.0	1.0	**0.3**	ns.	tl.	tl.	tl.	tl.	tl.
case179_goc_api	1932043.6	6.3	**5.9**	**5.8**	**0.6**	0.6	tl.	881.61	tl.	343.73	1000.46
case200_activ_api	35701.3	0.0	0.0	0.0	0.0	0.0	1993.90	2764.12	tl.	3029.17	**855.60**
case240_pserc_api	4639006.1	ns.	**0.6**	0.6	0.6	0.6	tl.	tl.	tl.	tl.	tl.
case300_ieee_api	684985.5	0.8	0.8	0.8	**0.7**	0.8	tl.	tl.	tl.	tl.	tl.

of the gap when compared with "E", and proves global optimality for case3_lmbd_api. With "ECB" and the faster "ECB*", we close the optimality gaps to lesser than 1.0% for ≈75% of all instances; these improvements are also significant when compared with state-of-the-art implementation in "PM" and the results in (Bestuzheva et al. 2020). Note that if the optimality gap equals zero, then the corresponding ACOTS relaxation provides a globally optimal solution to the non-convex ACOTS problem.

Although not shown in the result table, it is worthy to mention that tightening the ACOTS relaxations does lead to different line switching decisions. Therefore, by tightening the relaxation, we make better decisions and obtain better approximations of the true cost after line switching.

6.2. Analysis for varying load profiles

We uniformly increase the loading condition, starting from the nominal value, of case30_ieee instance, and observe the number of lines that are switched off. As shown in Figure 4, the number of off lines first decreases, and then increases. This is because when the load is at the lower levels, some lines are redundant and are switched off to save costs. However, when the load is very high and the network is congested, lines are switched off to avoid congestion. As suggested by Fisher et al. (2008), it would be beneficial to solve the ACOTS problem frequently to obtain optimal line switching decisions for different load profiles.

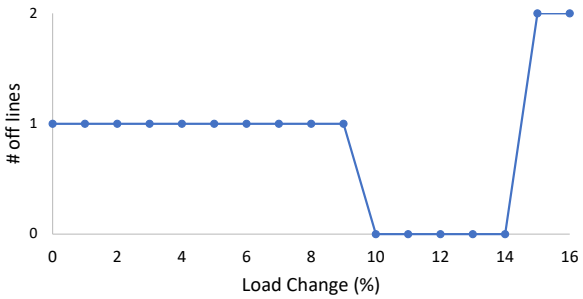

Figure 4 **Number of lines switched off with different load levels.**

6.3. Lifted cycle constraints for ACOPF-QC

We also add the linearized lifted cycle constraints (including the novel ones we derived) to the ACOPF-QC relaxation, This happens to be the special case of the ACOTS-QC with all the lines of the network swithced on. These cycle constraints are linearized using the strong extreme-point representation, which is also new. We experiment with "TYP", "SAD" and

Table 3 Comparing optimality Gaps (%) for "E" and "EC" relaxations for ACOPF-QC.

Test Case	E	EC
case3_lmbd_sad	1.38	1.31
case14_ieee_sad	19.16	13.10
case24_ieee_rts_sad	2.74	2.20
case57_ieee_sad	0.32	0.25
case73_ieee_rts_sad	2.37	1.80
case240_pserc_sad	4.34	4.24
case2383wp_k_sad	1.91	1.88
case3_lmbd_api	4.53	3.85
case24_ieee_rts_api	11.02	10.88
case73_ieee_rts_api	9.52	9.31
case179_goc_api	5.86	5.75

"API" cases with up to 2869 buses (102 cases in total), and observe improvements of optimality gaps in several instances. We report cases with greater than 0.03% improvement in objectives in Table 3. The results show that, for ACOPF-QC the lifted cycle constraints are more useful in tightening "SAD" and "API" instances, and for smaller-size test cases.

7. Conclusion

In this paper, we strengthen the on/off QC relaxation of the ACOTS model by the extreme-point representation technique, several valid inequalities added via branch-and-cut, and the OBBT algorithm. Experiments on PGLib instances show that the strengthened ACOTS-QC formulation significantly improves lower bounds in several instances, especially for small angle-difference instances and congested instances. Our proposed lifted cycle constraints improve bounds of ACOTS-QC as well as ACOPF-QC relaxations.

Considering large-scale grids, operated in a close-to-real-time fashion, ACOTS is still a very hard problem to solve with optimality guarantees. As the line switching decisions are sensitive to load changes, it would be helpful to develop stochastic programming models that provide more robust solutions. To address scaling issues, it would be useful to (i) balance the trade-off between run time and tighter formulations and (ii) develop faster decomposition-based distributed algorithms.

References

Babaeinejadsarookolaee S, Birchfield A, Christie RD, Coffrin C, DeMarco C, Diao R, Ferris M, Fliscounakis S, Greene S, Huang R, et al. (2019) The power grid library for benchmarking AC optimal power flow algorithms. *arXiv preprint arXiv:1908.02788* .

Bai X, Wei H, Fujisawa K, Wang Y (2008) Semidefinite programming for optimal power flow problems. *Int. J. Electr. Power Energy Syst.* 30(6-7):383–392.

Barrows C, Blumsack S, Hines P (2014) Correcting optimal transmission switching for AC power flows. *2014 47th Hawaii Int. Conf. on Syst. Sci.*, 2374–2379 (IEEE).

Bestuzheva K, Hijazi H, Coffrin C (2020) Convex relaxations for quadratic on/off constraints and applications to optimal transmission switching. *INFORMS J. Comput.* 32(3):682–696.

Ceria S, Soares J (1999) Convex programming for disjunctive convex optimization. *Math. Prog.* 86(3):595–614.

Chen C, Atamtürk A, Oren SS (2015) Bound tightening for the alternating current optimal power flow problem. *IEEE Trans. on Power Systems* 31(5):3729–3736.

Coffrin C, Bent R, Sundar K, Ng Y, Lubin M (2018) PowerModels.jl: An open-source framework for exploring power flow formulations. *2018 Power Syst. Comput. Conf. (PSCC)*, 1–8.

Coffrin C, Hijazi HL, Lehmann K, Van Hentenryck P (2014) Primal and dual bounds for optimal transmission switching. *2014 Power Syst. Comput. Conf.*, 1–8 (IEEE).

Coffrin C, Hijazi HL, Van Hentenryck P (2016) The QC relaxation: A theoretical and computational study on optimal power flow. *IEEE Trans. Power Syst.* 31(4):3008–3018.

Fisher EB, O'Neill RP, Ferris MC (2008) Optimal transmission switching. *IEEE Trans. Power Syst.* 23(3):1346–1355.

Glavitsch H (1985) State of the art review: Switching as means of control in the power system. *Int. J. Electr. Power & Energy Syst.* 7(2):92–100.

Goldis EA, Li X, Caramanis MC, Keshavamurthy B, Patel M, Rudkevich AM, Ruiz PA (2013) Applicability of topology control algorithms (TCA) to a real-size power system. *2013 51st Annu. Allerton Conf. on Commun., Control, and Comput. (Allerton)*, 1349–1352 (IEEE).

Gopinath S, Hijazi H, Weisser T, Nagarajan H, Yetkin M, Sundar K, Bent R (2020) Proving global optimality of ACOPF solutions. *Electr. Power Syst. Res.* 189:106688.

Hedman KW, O'Neill RP, Fisher EB, Oren SS (2008) Optimal transmission switching-sensitivity analysis and extensions. *IEEE Trans. on Power Systems* 23(3):1469–1479.

Hedman KW, Oren SS, O'Neill RP (2011) A review of transmission switching and network topology optimization. *2011 IEEE Power and Energy Soc. Gen. Meet.*, 1–7 (IEEE).

Hijazi H, Coffrin C, Van Hentenryck P (2017) Convex quadratic relaxations for mixed-integer nonlinear programs in power systems. *Math. Prog. Comput.* 9(3):321–367.

Jabr RA (2006) Radial distribution load flow using conic programming. *IEEE Trans. Power Syst.* 21(3):1458–1459.

Kocuk B, Dey SS, Sun XA (2016a) Strong SOCP relaxations for the optimal power flow problem. *Oper. Res.* 64(6):1177–1196.

Kocuk B, Dey SS, Sun XA (2017) New formulation and strong MISOCP relaxations for AC optimal transmission switching problem. *IEEE Trans. Power Syst.* 32(6):4161–4170.

Kocuk B, Dey SS, Sun XA (2018) Matrix minor reformulation and SOCP-based spatial branch-and-cut method for the AC optimal power flow problem. *Math. Prog. Comput.* 10(4):557–596.

Kocuk B, Jeon H, Dey SS, Linderoth J, Luedtke J, Sun XA (2016b) A cycle-based formulation and valid inequalities for DC power transmission problems with switching. *Oper. Res.* 64(4):922–938.

Kroger O, Coffrin C, Hijazi H, Nagarajan H (2018) Juniper: an open-source nonlinear branch-and-bound solver in julia. *Internation. Conf. Integr. Constraint Program., Artif. Intell., Oper. Res.*, 377–386 (Springer).

Lehmann K, Grastien A, Van Hentenryck P (2014) The complexity of DC-switching problems. *arXiv preprint arXiv:1411.4369* .

Lu M, Nagarajan H, Bent R, Eksioglu SD, Mason SJ (2018) Tight piecewise convex relaxations for global optimization of optimal power flow. *Power Syst. Comput. Conf.*, 1–7 (IEEE).

Lu M, Nagarajan H, Yamangil E, Bent R, Backhaus S, Barnes A (2017) Optimal transmission line switching under geomagnetic disturbances. *IEEE Trans. Power Syst.* 33(3):2539–2550.

Luedtke J, Namazifar M, Linderoth J (2012) Some results on the strength of relaxations of multilinear functions. *Math. Prog.* 136(2):325–351.

Nagarajan H, Lu M, Wang S, Bent R, Sundar K (2019a) An adaptive, multivariate partitioning algorithm for global optimization of nonconvex programs. *J. Global. Opt.* 74(4):639–675.

Nagarajan H, Sundar K, Hijazi H, Bent R (2019b) Convex hull formulations for mixed-integer multilinear functions. *AIP Conf. Proc.*, volume 2070, 020037 (AIP Publishing LLC).

Purchala K, Meeus L, Van Dommelen D, Belmans R (2005) Usefulness of DC power flow for active power flow analysis. *IEEE Power Eng. Soc. General Meeting, 2005*, 454–459 (IEEE).

Sundar K, Nagarajan H, Misra S, Lu M, Coffrin C, Bent R (2018) Optimization-based bound tightening using a strengthened QC-relaxation of the optimal power flow problem. *arXiv preprint: 1809.04565* .

Taylor JA (2015) *Convex optimization of power systems* (Cambridge University Press).

Wachter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Prog.* 106(1):25–57.