

This is the author accepted manuscript. To be cited as:
Guo, Cheng, Harsha Nagarajan, and Merve Bodur. “Tightening Quadratic Convex Relaxations for the AC Optimal Transmission Switching Problem.” *INFORMS Journal on Computing* (2025), DOI: <https://doi.org/10.1287/ijoc.2023.0236>.

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

School of Mathematics and Maxwell Institute for Mathematical Sciences, University of Edinburgh, Edinburgh, EH9 3FD, UK, merve.bodur@ed.ac.uk

The Alternating Current Optimal Transmission Switching (ACOTS) problem incorporates line switching decisions into the AC Optimal Power Flow (ACOPF) framework, offering well-known benefits in reducing operational costs and enhancing 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 in the relaxation using extreme-point representation, demonstrating theoretical tightness, and efficiently incorporate on/off cycle-based polynomial constraints through disjunctive programming-based cutting planes. Combined with an optimization-based bound tightening algorithm, this results in the tightest QC-based ACOTS relaxation to date. We additionally propose a novel maximum spanning tree-based heuristic to improve the computational performance by fixing certain lines to be switched on. Our extensive numerical experiments on medium-scale PGLib instances show significant improvements on relaxation bounds, while tests on large-scale instances with up to 2,312 buses demonstrate substantial performance gains. To our knowledge, this is the first ACOTS relaxation-based approach to demonstrate near-optimal switching solutions on realistic large-scale power grid instances.

Key words: Optimal transmission line switching, On/off quadratic convex relaxation, Extreme-point representation, Cycle constraints, 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), relies on this DC approximation. Kocuk et al. (2016b) developed a cycle-induced relaxation for a DCOTS model, and characterized its convex hull.

The drawback of DCOTS approximation is that the optimal decisions may not accurately represent power flows or may even be infeasible in the AC setting (Coffrin et al. 2014). This drawback motivates 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, both standard SDP and QC relaxations are at least as strong as the SOC relaxation, while the SDP and QC relaxation strengths are not directly 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 additional 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 coordinates. 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. (2023) 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. Additionally, Fattahi et al. (2018) employs bound tightening methods for the simpler DCOTS model, testing them on large-scale networks. 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. It can also solve several large-scale PGLib instances with 500 to 2,312 buses, which, to the best of our knowledge, have not been previously solved by the ACOTS-QC relaxation.

(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. The lifted cycle constraints are also useful for obtaining solutions of large-scale PGLib instances.

(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.

(4) We propose a novel heuristic that solves a *restricted* ACOTS-QC model by fixing a maximum spanning tree of the network to remain switched on. This approach significantly improves computational performance on medium-scale networks, enables the solution of large-scale instances that were previously unsolved, and provides near-optimal solutions.

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. $\Re(\cdot)$ and $\Im(\cdot)$ respectively denote the real and imaginary parts of a complex number. Given any two complex numbers (variables/constants) z_1 and z_2 , $z_1 \geq z_2$ implies $\Re(z_1) \geq \Re(z_2)$ and $\Im(z_1) \geq \Im(z_2)$. $|\cdot|$ 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. 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}^{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.
c_0^g, c_1^g, c_2^g	Generation cost coefficients.
\mathbf{j}	Unit imaginary number.
$Y_{ij} = g_{ij} + \mathbf{j}b_{ij}$	Admittance on line (i, j) .
$Y_{ij}^c = g_{ij}^c + \mathbf{j}b_{ij}^c$	Charging admittance on line (i, j) .
$Y_{ij}^s = g_{ij}^s + \mathbf{j}b_{ij}^s$	Shunt admittance on line (i, j) .
$T_{ij} = t_{ij}^R + \mathbf{j}t_{ij}^I$	Branch complex transformation ratio (tap ratio) on line (i, j) .
$S_i^d = p_i^d + \mathbf{j}q_i^d$	AC power demand at bus i .
$\underline{c}_{ij}, \bar{c}_{ij}$	Cosine function ($\cos(\theta_{ij})$) bounds on line (i, j) .
$\underline{s}_{ij}, \bar{s}_{ij}$	Sine function ($\sin(\theta_{ij})$) bounds on line (i, j) .
$\underline{s}_{ij}^a, \bar{s}_{ij}^a$	Apparent power bound on line (i, j) .
$\underline{\theta}_{ij}, \bar{\theta}_{ij}$	Phase angle difference bounds on line (i, j) .
$\underline{v}_i, \bar{v}_i$	Voltage magnitude bounds at bus i .
$\underline{S}_i^g, \bar{S}_i^g$	Power generation bounds at bus i .
\underline{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 .
$V_i = v_i e^{\mathbf{j}\theta_i}$	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:

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

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

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

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

$$w_i = v_i^2 \quad \forall i \in \mathcal{N} \quad (1e)$$

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

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

$$\underline{\theta}_{ij} z_{ij} - \boldsymbol{\theta}^M (1 - z_{ij}) \leq \theta_{ij} \leq \bar{\theta}_{ij} z_{ij} + \boldsymbol{\theta}^M (1 - z_{ij}) \quad \forall (i,j) \in \mathcal{A} \quad (1h)$$

$$\theta_i = 0 \quad \forall i \in \mathcal{N}^{\text{ref}} \quad (1i)$$

$$\underline{\mathbf{S}}_i^g \leq S_k^g \leq \bar{\mathbf{S}}_k^g \quad \forall k \in \mathcal{G} \quad (1j)$$

$$|S_{ij}|^2 \leq (\bar{\mathbf{s}}_{ij}^a)^2 z_{ij}^2, \quad |S_{ji}|^2 \leq (\bar{\mathbf{s}}_{ij}^a)^2 z_{ij}^2 \quad \forall (i,j) \in \mathcal{A} \quad (1k)$$

$$\underline{v}_i \leq v_i \leq \bar{v}_i \quad \forall i \in \mathcal{N} \quad (1l)$$

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

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 $\boldsymbol{\theta}_{ij}^u = \max(|\underline{\theta}_{ij}|, |\bar{\theta}_{ij}|)$. Let $\boldsymbol{\theta}_{ij,k}^{u,\max}$ be the k th largest value in $\{\boldsymbol{\theta}_{ij}^u \mid (i,j) \in \mathcal{A}\}$, and $\boldsymbol{\theta}^M =$

$\sum_{k=1}^{|\mathcal{N}|-1} \theta_{ij,k}^{\mathbf{u}, \max}$ 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 $(i, j) \in \mathcal{A}$:

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

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

$$w_i - (1 - z_{ij}) \bar{\mathbf{v}}_i^2 \leq w_{ij}^z \leq w_i - (1 - z_{ij}) \underline{\mathbf{v}}_i^2 \quad (2c)$$

$$w_j - (1 - z_{ij}) \bar{\mathbf{v}}_j^2 \leq w_{ji}^z \leq w_j - (1 - z_{ij}) \underline{\mathbf{v}}_j^2 \quad (2d)$$

$$\underline{\mathbf{v}}_i^2 z_{ij} \leq w_{ij}^z \leq \bar{\mathbf{v}}_i^2 z_{ij} \quad (2e)$$

$$\underline{\mathbf{v}}_j^2 z_{ij} \leq w_{ji}^z \leq \bar{\mathbf{v}}_j^2 z_{ij}. \quad (2f)$$

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 ($\forall (i, j) \in \mathcal{A}$):

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

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

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 to a quadratic inequality constraint (4a), as done in (Hijazi et al. 2017), and providing upper bounds via linear McCormick relaxation constraints in (4b):

$$w_i \geq v_i^2 \quad (4a)$$

$$w_i \leq (\underline{v}_i + \bar{v}_i)v_i - \underline{v}_i\bar{v}_i \quad (4b)$$

(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 captures the the convex envelope of $\cos(\theta_{ij})$. We assume $\theta_{ij}^u \leq \pi/2$, 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:

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

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:

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

$$c_{ij} \leq z_{ij} - \frac{1 - \cos(\theta_{ij}^u)}{(\theta_{ij}^u)^2} \theta_{ij}^2 + \frac{1 - \cos(\theta_{ij}^u)}{(\theta_{ij}^u)^2} (\theta^M)^2 (1 - z_{ij}) \quad (5b)$$

$$\underline{c}_{ij} z_{ij} \leq c_{ij} \leq \bar{c}_{ij} z_{ij}. \quad (5c)$$

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 θ^M is used to ensure that when $z_{ij} = 0$, those constraints are valid for $c_{ij} = 0$ and $\theta_{ij} \in [\underline{\theta}_{ij}, \bar{\theta}_{ij}]$. 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 a lifted variable s_{ij} that captures the the convex envelope of $\sin(\theta_{ij})$. When $\theta_{ij}^u \leq \pi/2$, a disjunctive relaxation of the sine function is as follows:

$$s_{ij} - \cos\left(\frac{\theta_{ij}^u}{2}\right) \theta_{ij} \leq \left(\sin\left(\frac{\theta_{ij}^u}{2}\right) - \cos\left(\frac{\theta_{ij}^u}{2}\right) \frac{\theta_{ij}^u}{2}\right) z_{ij} + \cos\left(\frac{\theta_{ij}^u}{2}\right) \theta^M(1 - z_{ij}), \text{ if } \bar{\theta}_{ij} \geq 0 \quad (6a)$$

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

$$s_{ij} - \theta_{ij}^s \theta_{ij} \leq (-\theta_{ij}^s \underline{\theta}_{ij} + \sin(\underline{\theta}_{ij})) z_{ij} + \theta_{ij}^s \theta^M(1 - z_{ij}), \text{ if } \bar{\theta}_{ij} \leq 0 \quad (6c)$$

$$-s_{ij} + \theta_{ij}^s \theta_{ij} \leq (\theta_{ij}^s \underline{\theta}_{ij} - \sin(\underline{\theta}_{ij})) z_{ij} + \theta_{ij}^s \theta^M(1 - z_{ij}), \text{ if } \underline{\theta}_{ij} \geq 0 \quad (6d)$$

$$\underline{s}_{ij} z_{ij} \leq s_{ij} \leq \bar{s}_{ij} z_{ij}, \quad (6e)$$

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) to incorporate on/off variables, as discussed below.

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

$$w_{ij}^R = \sum_{k=1}^8 \lambda_{ij,k}^c (\xi_1^k \xi_2^k \xi_3^k), \quad w_{ij}^I = \sum_{k=1}^8 \lambda_{ij,k}^s (\gamma_1^k \gamma_2^k \gamma_3^k) \quad (7a)$$

$$\sum_{k=1}^8 \lambda_{ij,k}^c \xi_1^k + (1 - z_{ij}) \underline{v}_i \leq v_i \leq \sum_{k=1}^8 \lambda_{ij,k}^c \xi_1^k + (1 - z_{ij}) \bar{v}_i \quad (7b)$$

$$\sum_{k=1}^8 \lambda_{ij,k}^s \gamma_1^k + (1 - z_{ij}) \underline{v}_i \leq v_i \leq \sum_{k=1}^8 \lambda_{ij,k}^s \gamma_1^k + (1 - z_{ij}) \bar{v}_i \quad (7c)$$

$$\sum_{k=1}^8 \lambda_{ij,k}^c \xi_2^k + (1 - z_{ij}) \underline{v}_j \leq v_j \leq \sum_{k=1}^8 \lambda_{ij,k}^c \xi_2^k + (1 - z_{ij}) \bar{v}_j \quad (7d)$$

$$\sum_{k=1}^8 \lambda_{ij,k}^s \gamma_2^k + (1 - z_{ij}) \underline{v}_j \leq v_j \leq \sum_{k=1}^8 \lambda_{ij,k}^s \gamma_2^k + (1 - z_{ij}) \bar{v}_j \quad (7e)$$

$$c_{ij} = \sum_{k=1}^8 \lambda_{ij,k}^c \xi_3^k, \quad s_{ij} = \sum_{k=1}^8 \lambda_{ij,k}^s \gamma_3^k \quad (7f)$$

$$\sum_{k=1}^8 \lambda_{ij,k}^c = z_{ij}, \quad \sum_{k=1}^8 \lambda_{ij,k}^s = z_{ij}, \quad \lambda_{ij,k}^c \geq 0, \quad \lambda_{ij,k}^s \geq 0, \quad \forall k = 1, \dots, 8 \quad (7g)$$

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

where ξ_1^k is the value of v_i in the extreme point ξ^k . The constants ξ_2^k , ξ_3^k , γ_1^k , γ_2^k , and γ_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 $[\underline{v}_i, \bar{v}_i] \times [\underline{v}_j, \bar{v}_j] \times [\underline{c}_{ij}, \bar{c}_{ij}]$. 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:

$$v_i = \sum_{k=1}^8 \lambda_{ij,k}^c \xi_1^k = \sum_{k=1}^8 \lambda_{ij,k}^s \gamma_1^k, \quad (8a)$$

$$v_j = \sum_{k=1}^8 \lambda_{ij,k}^c \xi_2^k = \sum_{k=1}^8 \lambda_{ij,k}^s \gamma_2^k, \quad (8b)$$

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

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

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

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

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

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

Sundar et al. (2023) 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:

$$\eta_0^* = \left(0, 0, \dots, 0, \frac{v_i^* - \sum_{k=1}^8 \lambda_{ij,k}^* \xi_1^k}{1 - z_{ij}^*}, \frac{v_j^* - \sum_{k=1}^8 \lambda_{ij,k}^* \xi_2^k}{1 - z_{ij}^*} \right)$$

$$\eta_1^* = \left(\frac{w_{ij}^{R*}}{z_{ij}^*}, \frac{w_{ij}^{I*}}{z_{ij}^*}, \frac{c_{ij}^*}{z_{ij}^*}, \frac{s_{ij}^*}{z_{ij}^*}, \frac{\lambda_{ij,1}^{c*}}{z_{ij}^*}, \dots, \frac{\lambda_{ij,8}^{c*}}{z_{ij}^*}, \frac{\lambda_{ij,1}^{s*}}{z_{ij}^*}, \dots, \frac{\lambda_{ij,8}^{s*}}{z_{ij}^*}, 1, \sum_{k=1}^8 \frac{\lambda_{ij,k}^{c*}}{z_{ij}^*} \xi_1^k, \sum_{k=1}^8 \frac{\lambda_{ij,k}^{s*}}{z_{ij}^*} \xi_2^k \right)$$

Next, we prove that $\eta_0^* \in H^0$ and $\eta_1^* \in H^1$. Because of (7b), we have $(1 - z_{ij}^*)\underline{\mathbf{v}}_i \leq v_j^* - \sum_{k=1}^8 \lambda_{ij,k}^{c*} \boldsymbol{\xi}_2^k \leq (1 - z_{ij}^*)\bar{\mathbf{v}}_i$. Thus, $\frac{v_i^* - \sum_{k=1}^8 \lambda_{ij,k}^{c*} \boldsymbol{\xi}_1^k}{1 - z_{ij}^*} \in [\underline{\mathbf{v}}_i, \bar{\mathbf{v}}_i]$. Similarly, $\frac{v_j^* - \sum_{k=1}^8 \lambda_{ij,k}^{c*} \boldsymbol{\xi}_2^k}{1 - z_{ij}^*} \in [\underline{\mathbf{v}}_j, \bar{\mathbf{v}}_j]$. Therefore, $\eta_0^* \in H^0$.

For η_1^* , $\frac{w_{ij}^{R*}}{z_{ij}^*} = \sum_{k=1}^8 \frac{\lambda_{ij,1}^{c*}}{z_{ij}^*} (\boldsymbol{\xi}_1^k, \boldsymbol{\xi}_2^k, \boldsymbol{\xi}_3^k)$ because $w_{ij}^{R*} = \sum_{k=1}^8 \lambda_{ij,1}^{c*} (\boldsymbol{\xi}_1^k, \boldsymbol{\xi}_2^k, \boldsymbol{\xi}_3^k)$ 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 $\sum_{k=1}^8 \lambda_{ij,k}^{s*} \gamma_1^k + (1 - z_{ij}^*)\underline{\mathbf{v}}_i \leq \sum_{k=1}^8 \lambda_{ij,k}^{c*} \boldsymbol{\xi}_1^k + (1 - z_{ij}^*)\underline{\mathbf{v}}_i$ and $\sum_{k=1}^8 \lambda_{ij,k}^{s*} \gamma_1^k + (1 - z_{ij}^*)\bar{\mathbf{v}}_i \geq \sum_{k=1}^8 \lambda_{ij,k}^{c*} \boldsymbol{\xi}_1^k + (1 - z_{ij}^*)\bar{\mathbf{v}}_i$, thus $\sum_{k=1}^8 \lambda_{ij,k}^{c*} \boldsymbol{\xi}_1^k = \sum_{k=1}^8 \lambda_{ij,k}^{s*} \gamma_1^k \Rightarrow \sum_{k=1}^8 \frac{\lambda_{ij,k}^{c*}}{z_{ij}^*} \boldsymbol{\xi}_1^k = \sum_{k=1}^8 \frac{\lambda_{ij,k}^{s*}}{z_{ij}^*} \gamma_1^k$, 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 $H \subseteq \text{conv}(H^0 \cup H^1)$. \square

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:

$$\tan(\underline{\boldsymbol{\theta}}_{ij})w_{ij}^R \leq w_{ij}^I \leq \tan(\bar{\boldsymbol{\theta}}_{ij})w_{ij}^R \quad (10a)$$

$$\begin{aligned} \mathbf{v}_i^\sigma \mathbf{v}_j^\sigma (\cos(\phi_{ij})w_{ij}^R + \sin(\phi_{ij})w_{ij}^I) - \bar{\mathbf{v}}_j \cos(\delta_{ij})\mathbf{v}_i^\sigma w_{ij}^z \\ - \bar{\mathbf{v}}_i \cos(\delta_{ij})\mathbf{v}_i^\sigma w_{ji}^z \geq \bar{\mathbf{v}}_i \bar{\mathbf{v}}_j \cos(\delta_{ij})(\underline{\mathbf{v}}_i \underline{\mathbf{v}}_j - \bar{\mathbf{v}}_i \bar{\mathbf{v}}_j)z_{ij} \end{aligned} \quad (10b)$$

$$\begin{aligned} \mathbf{v}_i^\sigma \mathbf{v}_j^\sigma (\cos(\phi_{ij})w_{ij}^R + \sin(\phi_{ij})w_{ij}^I) - \underline{\mathbf{v}}_j \cos(\delta_{ij})\mathbf{v}_i^\sigma w_{ij}^z \\ - \underline{\mathbf{v}}_i \cos(\delta_{ij})\mathbf{v}_i^\sigma w_{ji}^z \geq \underline{\mathbf{v}}_i \underline{\mathbf{v}}_j \cos(\delta_{ij})(\bar{\mathbf{v}}_i \bar{\mathbf{v}}_j - \underline{\mathbf{v}}_i \underline{\mathbf{v}}_j)z_{ij} \end{aligned} \quad (10c)$$

$$|S_{ij}|^2 \leq \frac{w_i}{|\mathbf{T}_{ij}|^2} l_{ij} \quad (10d)$$

$$\begin{aligned} l_{ij} = |\mathbf{Y}_{ij}|^2 \left(\frac{w_{ij}^z}{|\mathbf{T}_{ij}|^2} + w_{ji}^z - 2(\mathbf{t}_{ij}^R w_{ij}^R + \mathbf{t}_{ij}^I w_{ij}^I) / |\mathbf{T}_{ij}|^2 \right) \\ - \frac{|\mathbf{Y}_{ij}^c|^2}{|\mathbf{T}_{ij}|^2} w_{ij}^z + 2(\mathbf{g}_{ij}^c p_{ij} - \mathbf{b}_{ij}^c q_{ij}) \end{aligned} \quad (10e)$$

$$0 \leq l_{ij} \leq \bar{l}_{ij} z_{ij}, \quad (10f)$$

where $\mathbf{v}_i^\sigma = \underline{\mathbf{v}}_i + \bar{\mathbf{v}}_i$, $\phi_{ij} = (\bar{\theta}_{ij} + \underline{\theta}_{ij})/2$, $\delta_{ij} = (\bar{\theta}_{ij} - \underline{\theta}_{ij})/2$ and $\bar{l}_{ij} = \frac{|\mathbf{T}_{ij}|^2 (\bar{\mathbf{s}}_{ij}^a)^2}{\underline{\mathbf{v}}_i^2}$. Constraint (10a) is the phase angle difference constraint, which is a relaxation of the equality $\tan(\theta_{ij}) = \frac{w_{ij}^R}{w_{ij}^I}$. 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 constraints (10e) and (10f) ensure that l_{ij} goes to 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:

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

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

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

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).

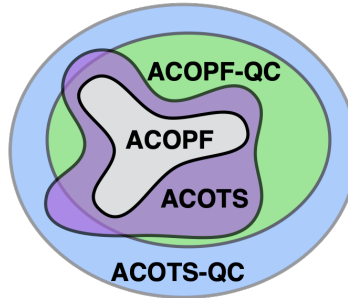


Figure 1 Venn diagram for solution sets of different formulations.

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_{ij}^R , w_{ij}^I , 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 develop 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 $(\hat{v}_1, \hat{v}_2, \dots, \hat{v}_n, \hat{v}_1)$ be a vertex sequence for the cycle \mathcal{C} of length n , and let the cycle be represented by its lines: $\mathcal{C} = \{(\hat{v}_1, \hat{v}_2), (\hat{v}_2, \hat{v}_3), \dots, (\hat{v}_n, \hat{v}_1)\}$, 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_{ij}^R, w_{ij}^I) 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:

$$c_{ik} = c_{ij}c_{jk} - s_{ij}s_{jk} \quad (12a)$$

$$s_{ik} = c_{ij}s_{jk} + s_{ij}c_{jk}, \quad (12b)$$

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):

$$w_j w_{ik}^R = w_{ij}^R w_{jk}^R - w_{ij}^I w_{jk}^I, \quad (13a)$$

$$w_j w_{ik}^I = w_{ij}^R w_{jk}^I + w_{ij}^I w_{jk}^R. \quad (13b)$$

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_{ij}^R , w_{ij}^I , 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 permuted 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:

$$c_{ij}c_{jk}c_{ik} + c_{ij}s_{jk}s_{ik} - s_{ij}s_{jk}c_{ik} + s_{ij}c_{jk}s_{ik} = 1 \quad (14a)$$

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

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 $\{(c, s) : (12) \text{ holds}\} = \{(c, s) : (14) \text{ holds}\}$.*

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). \square

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

$$c_{ij}c_{kl} - s_{ij}s_{kl} = c_{il}c_{jk} + s_{il}s_{jk} \quad (15a)$$

$$c_{ij}s_{kl} + s_{ij}c_{kl} = -c_{il}s_{jk} + s_{il}c_{jk} \quad (15b)$$

$$w_{ij}^R w_{kl}^R - w_{ij}^I w_{kl}^I = w_{il}^R w_{jk}^R + w_{il}^I w_{jk}^I \quad (15c)$$

$$w_{ij}^R w_{kl}^I + w_{ij}^I w_{kl}^R = -w_{il}^R w_{jk}^I + w_{il}^I w_{jk}^R. \quad (15d)$$

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:

$$-3\widehat{z} \leq c_{ik} - c_{ij}c_{jk} + s_{ij}s_{jk} \leq 3\widehat{z} \quad (16a)$$

$$-3\widehat{z} \leq s_{ik} - c_{ij}s_{jk} - s_{ij}c_{jk} \leq 3\widehat{z}. \quad (16b)$$

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 $x_i^c \ \forall i = 1, \dots, 6$ 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:

$$x_3^c = x_1^c x_2^c - x_4^c x_5^c, \quad x_6^c = x_1^c x_5^c + x_2^c x_4^c \quad (17a)$$

$$x_1^c = x_2^c x_3^c + x_5^c x_6^c, \quad x_4^c = x_2^c x_6^c - x_3^c x_5^c \quad (17b)$$

$$x_2^c = x_1^c x_3^c + x_4^c x_6^c, \quad x_5^c = x_1^c x_6^c - x_3^c x_4^c. \quad (17c)$$

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. $x_{j_1 j_2}^c$ is a lifted variable for $x_{j_1}^c x_{j_2}^c$. We can linearize the constraint $x_3^c = x_1^c x_2^c - x_4^c x_5^c$ and connect the constraint with line switching decisions as follows:

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

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):

$$\sum_{i=1}^{64} \lambda_i^{cs} = y_C \quad (19a)$$

$$\lambda_i^{cs} \geq 0 \quad \forall i = 1, \dots, 64 \quad (19b)$$

$$x_j^c \geq \underline{x}_j^c \left(\sum_{i: (\mathcal{X}_j^i = \underline{x}_j^c)} \lambda_i^{cs} \right) + \bar{x}_j^c \left(\sum_{i: (\mathcal{X}_j^i = \bar{x}_j^c)} \lambda_i^{cs} \right) + \min(0, \underline{x}_j^c)(1 - y_C) \quad \forall j = 1, \dots, 6 \quad (19c)$$

$$x_j^c \leq \underline{x}_j^c \left(\sum_{i: (\mathcal{X}_j^i = \underline{x}_j^c)} \lambda_i^{cs} \right) + \bar{x}_j^c \left(\sum_{i: (\mathcal{X}_j^i = \bar{x}_j^c)} \lambda_i^{cs} \right) + \max(0, \bar{x}_j^c)(1 - y_C) \quad \forall j = 1, \dots, 6 \quad (19d)$$

$$x_{j_1 j_2}^c = \sum_{i=1}^{64} \lambda_i^{cs} (\mathcal{X}_{j_1}^i \mathcal{X}_{j_2}^i) \quad \forall (j_1, j_2) \in \mathcal{P} \quad (19e)$$

$$1 - \sum_{(i,j) \in \mathcal{C}} (1 - z_{ij}) \leq y_C \leq \frac{1}{|\mathcal{C}|} \sum_{(i,j) \in \mathcal{C}} z_{ij} \quad (19f)$$

$$y_C \in \{0, 1\} \quad (19g)$$

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 $x_j^c \in [\underline{x}_j^c, \bar{x}_j^c] \forall j = 1, \dots, 6$. Constraints (19a) and (19b) set bounds for auxiliary multiplier variables. When $y_C = 1$, constraints (19c), (19d), and (19e) represent the convex hull consisting of variables $x_j^c (\forall j)$ and $x_{j_1 j_2}^c (\forall (j_1, j_2))$; when $y_C = 0$ constraints (19c) and (19d) become redundant. Constraint (19f) connects y_C and z_{ij} : when all lines are switched on, (19f) fixes y_C to 1. If any line is switched off, $1 - \sum_{(i,j) \in \mathcal{C}} (1 - z_{ij}) \leq 0$ and $\frac{1}{|\mathcal{C}|} \sum_{(i,j) \in \mathcal{C}} z_{ij} \in [0, 1]$, which enforce $y_C = 0$. Also note that when $y_C = 0$, (19a), (19b) and (19e) ensure $x_{12}^c = x_{45}^c = 0$, 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 $\sum_{(j_1, j_2) \in \mathcal{P}} x_{j_1}^c x_{j_2}^c$. Without loss of generality, we set the domain of $x_i^c, i = 1, \dots, 6$ 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.

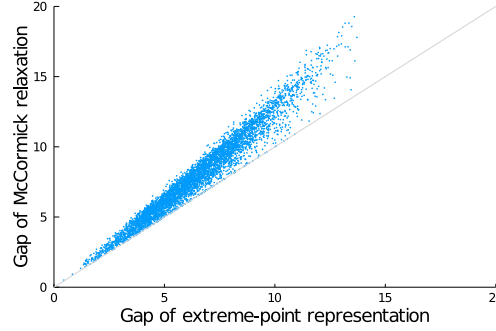


Figure 2 Scatter plot comparing extreme-point formulation with McCormick relaxation for summation of bilinear terms.

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 $x^c = (c_{ij}, c_{jk}, c_{ik}, s_{ij}, s_{jk}, s_{ik})$ and let x^{c*} be an optimal solution of the ACOTS-QC model. We solve the following separation problem:

$$\min \quad 0 \tag{20a}$$

$$\text{s.t.} \quad \text{extreme-point representation of (17)} \tag{20b}$$

$$x^c = x^{c*}. \tag{20c}$$

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

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

where β and \mathbf{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_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 $y_c = 1$. In other words, the feasible region defined by the reformulated Benders cut is a union of the following two sets:

$$\Gamma_0 = \{(x^c, y_c) : y_c = 0, \min(0, \underline{x}^c) \leq x^c \leq \max(0, \overline{x}^c)\}$$

$$\Gamma_1 = \{(x^c, y_c) : y_c = 1, \beta^\top x^c \leq \mathbf{b}\}$$

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

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

where \mathcal{N} is the set of indices for β and x^c . Intuitively, when $y_c = 1$, (22) is exactly the Benders cut (21); when $y_c = 0$, (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^c and switching decisions z_{ij} , and denote them respectively by \mathbf{x}^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 and obtain near global-optimal solutions for ACOPF problems (Chen et al. 2015, Cengil et al. 2022), 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.

6. Maximum Spanning Tree Heuristic

Heuristics have been developed to quickly find good feasible solutions for the OTS problem, prioritizing speed over optimality (Fuller et al. 2012, Barrows et al. 2012, Soroush and Fuller 2013, Hinneck and Pozo 2022). These approaches typically reduce the solution space by fixing a set of candidate lines based on objective sensitivities. However, they often apply heuristics to the simpler, less accurate DCOTS model for large-scale networks or focus on

quantifying the accuracy of such decisions. None have explored the problem using more accurate ACOTS relaxations, particularly for large-scale networks.

In this section, we propose a heuristic to accelerate the solution of the **(ACOTS-QC)** formulation in (11), where we find a maximum spanning tree in the network, and keep all lines in the spanning tree switched on. Note that a maximum spanning tree is a spanning tree of a weighted graph with the maximum total weight.

Based on empirical observations, most transmission lines remain switched on in the optimal solution. Inspired by this observation, we heuristically identify the lines that are likely the most essential for transmission and that maintain network connectivity, keeping these lines switched on to reduce the number of binary variables. Specifically, we assign the weight $\max\left(\frac{|\hat{S}_{ij}|^2}{(\bar{s}_{ij}^a)^2}, \frac{|\hat{S}_{ji}|^2}{(\bar{s}_{ij}^a)^2}\right)$ to each line (i, j) , where \hat{S}_{ij} is the locally optimal solution corresponding to the AC power flow variable, S_{ij} , in the ACOPF problem (e.g., obtained using Ipopt) based on the original topology with all lines active. Intuitively, lines with higher weights exhibit lower slack. We then find a maximum spanning tree for the network and allow only lines outside the tree to be switched off.

This heuristic restricts the solution space of switching variables, potentially leading to sub-optimal solutions for (11). However, even sub-optimal line-switching can reduce grid operating costs. Our numerical experiments in Section 8.2 demonstrate that the heuristic can quickly find near-optimal solutions for many instances, particularly larger ones. Additionally, in Section 8.3, we show that the heuristic obtains results for several large instances with 500 to 2,312 buses, which, to our knowledge, have not been previously explored for ACOTS with tight convex relaxations.

7. Overview of Proposed Algorithm

In this section, we provide a summary of our algorithm with all proposed improvements (with lifted cycle constraints added in branch-and-cut fashion).

We start the algorithm by preprocessing variable bounds via OBBT as described in Section 5. Those bounds are added to **(ACOTS-QC)**, which is then solved via branch-and-bound. Inside the branch-and-bound search, lifted cycle constraints are generated as described in Section 4.3 and added at integral nodes via lazy callback of the solver. The algorithm terminates when the gap between the upper and lower bounds of the branch-and-bound search is below a small tolerance. Finally, if we opt to use the maximum spanning

tree heuristic, it can be incorporated before OBBT. We summarize the implementation of the algorithm in Figure 3.

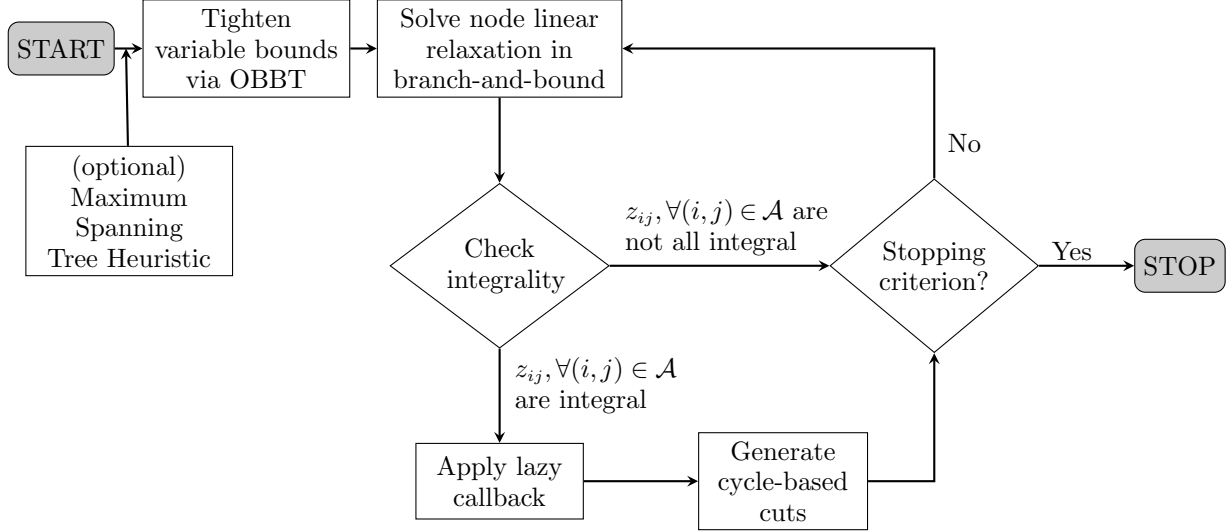


Figure 3 Flow chart of the proposed algorithm.

Combining all strengthening techniques in Figure 3 provides the tightest QC-based ACOTS relaxation in the literature. Note that when strengthening the ACOTS relaxations, two factors are crucial: (i) tightness of the MINLP convexification, and (ii) tightness of the corresponding ACOPF relaxation. The later is important because if the binary variables are known, then the ACOTS problem simplifies to an ACOPF problem. In our work, several approaches, such as disjunctive programming-based lifted cycle constraints, are employed to strengthen the MINLP bound. While some of our proposed methods, such as the lifted cycle constraints, can also improve the ACOPF relaxation, as shown in Section 8.5, our main focus is not on improving the ACOPF bound, which remains challenging on a large-scale and warrants further exploration.

8. 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 (Babaeinejadsarookolaei et al. 2019). For medium-scale instances, we use a Linux workstation with 3.6GHz Intel Core i9-9900K CPUs and 128GB memory; For large-scale instances

in Section 8.3, we use Linux workstations with Intel CPUs and 250GB memory. The programming language is Julia v1.6. We locally solve all non-convex MINLP (ACOTS) formulations with both Juniper.jl (v0.7.0) (Kroger et al. 2018) and Knitro (v13.0.1) (Byrd et al. 2006). Non-convex NLP (ACOPF) formulations are solved locally using Ipopt (v3.13.4) (Wachter and Biegler 2006). 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. The code for our experiments are available in the INFORMS Journal on Computing GitHub software repository (Guo et al. 2025).

8.1. Relaxations for ACOTS

We compare five different types of relaxations for ACOTS:

- (1) “P”: 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 “P” 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 $z_{ij}v_iv_js_{ij}$ and $z_{ij}v_iv_js_{ij}$ in (3).
- (3) “EC”: Tightened “E” with lifted cycle constraints. The lifted cycle constraints are included in (**ACOTS-QC**) at the start of the algorithm.
- (4) “ECB”: Includes all proposed improvements (extreme-point representation, lifted cycle constraints, and OBBT). The lifted cycle constraints are included in (**ACOTS-QC**) at the start of the algorithm (Compare with “ECB*” below).
- (5) “ECB*”: The same as “ECB”, except the lifted cycle constraints are added via branch-and-cut framework as in Section 4.3. The number of added cuts is bounded at 200.

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 “P”), 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), which are all created by (Babaeinejadsarookolaee et al. 2019). More specifically, the TYP cases are base cases with no change in the standard PGLib-OPF networks. The SAD cases modify the TYP cases by reducing the voltage angle difference on all branches of the standard networks, while the API cases modify the TYP cases by increasing active power demand throughout the standard networks.

In Table 2 we present results for cases that are solved in the 2-hour time limit (within 0.1% optimality tolerance) for “E”. As shown in Table 7 of the Online Supplement, which provides the relative gap at termination for all PGLib instances up to 300 buses, instances not solved by “E” are also unsolved by other methods. 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 $(UB - LB)/UB \times 100$ 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 four types of methods:

- Non-convex ACOTS model (1) with the Knitro solver.
- Non-convex ACOTS model (1) with the Juniper solver.
- 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 separation problems in “ECB*”

within the branch-and-cut algorithm (we separately provide the separation problem building time in Table 8 of the Online Supplement), as any overhead in such time is an artifact of the mathematical modeling package within Julia. We set the feasibility tolerance for “EC” and “ECB*” to 10^{-4} , instead of Gurobi’s default of 10^{-6} , because with a smaller tolerance the solver may incorrectly reject feasible warm-start solutions for some instances. 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 “P”, 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 “P” 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%.

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*” may not be as tight as “ECB,” since “ECB” includes all cycle constraints, while “ECB*” adds up to 200 cycle constraints via cutting planes. However, when “ECB*” is not as tight, the difference is only 0.1%, indicating that “ECB*” remains relatively strong. Due to the efficient implementation of the branch-and-cut framework, “ECB*” reduces the solution time significantly in many instances. 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% of the gap when compared with “E”, and proves global optimality for `case3_lmbd_api`. Compared with “E”, “ECB” and the faster “ECB*” reduce the gap of nine instances to below 1.0%. In fact, with “ECB” and “ECB*”, we close the optimality gaps to less than 1.0% for $\approx 75\%$ of all instances; these improvements are also significant when compared with state-of-the-art implementation in “P” 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.

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**).

Test Case	UB	Optimality Gap (%)					Runtime (seconds)				
		P	E	EC	ECB	ECB*	P	E	EC	ECB	ECB*
Typical Operating 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	ns.	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	754083.7	0.1	0.1	0.1	0.1	0.1	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	106633.6	ns.	0.1	ns.	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	2925.1	5.4	5.4	5.4	5.1	5.1	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	ns.	1.2	1.2	128.08	3869.93	tl.	3307.26	328.03
case89_pegase_api	100325.3	ns.	0.2	ns.	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	ns.	ns.	ns.	tl.	tl.	tl.	tl.	tl.
case179_goc_api	1932020.5	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	4638308.7	ns.	0.6	0.6	0.5	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.

Note that there are a few instances that are not solved due to no feasible solution found within the time limit. In particular, for the instances with 89 buses, we do not provide warm-start solutions when cycle constraints are included, as these instances have a large number of cycles, resulting in large warm-start files that cause a stack overflow error in the solver. In addition, the warm-start solution is incorrectly rejected by the solver when solving case200_activ with “ECB*”. Without warm start, these instances struggle to find a feasible solution.

We also check the tightness of the root node relaxation. To find the values of the root node relaxation, we set the Gurobi “NodeLimit” parameter to 0 and obtain the optimal objective bound. In Table 9 of the Online Supplement, we present the optimality gaps at the root node for the ACOTS relaxations, which equals $100 \times (\text{UB} - \text{root node relaxation value}) / \text{UB}$. In contrast to the results in Table 2, the improvement of “E” is relatively small at the root node, indicating that “E” closes more gap during branch-and-bound. “EC” provides tight root relaxations for several SAD instances. The “ECB” relaxation provides tighter root node relaxations for many instances, in particular for SAD and API ones, which possibly explains the reduced final optimality gaps in Table 2. Note that unlike “ECB”, the bound tightening optimization models for “ECB*” are not strengthened by cycle constraints, leading to potentially weaker bounds after OBBT and weaker root node relaxations. In fact, “ECB*” shows weaker root relaxations than “ECB” in several instances, suggesting that the tighter root relaxation of “ECB” is a result of both bound tightening and cycle constraints.

In addition, we include the results for the “PB” and “EB” relaxations (i.e., tightened “P” and “E” respectively with OBBT) in Table 10 of the Online Supplement. We highlight with boldface-italic the optimality gaps that can be further improved by “EB” (for “PB”) and by “ECB” (for “EB”). The difference between “PB” and “EB” shows that our extreme-point representation is useful for tightening many instances, even after OBBT is implemented. Also, adding lifted cycle constraints (“ECB”) tightens “EB” for many instances.

To summarize, our basic relaxation with extreme-point representation (i.e., “E”) provides improvements over many of the instances compared with the state-of-the-art benchmark, with comparable computational time. Both the lifted cycle constraints and OBBT provide enhancements to the basic relaxation. Combining all the improvements provides

the tightest QC-based ACOTS relaxation in the literature, with the runtime comparable to the basic relaxation when implemented in the branch-and-cut format.

Note that for obtaining the UB, Juniper fails to find feasible solutions for many instances within the time limit (2 hours), while Knitro found feasible solutions for all instances. On the other hand, for the instances that Juniper can solve, it sometimes provides better solutions. We provide a comparison of the results from the two solvers in Table 11 of the Online Supplement.

Although not shown in the result table, it is worthy to mention that tightening the ACOTS relaxations does lead to different line switching decisions. For example, in “case24_ieee_rts_api”, 13.9% of the lines have different switching decisions between “E” and “ECB”. If we fix the on/off statuses of the lines based on the solution of “E” and then solve the problem with “ECB”, the cost increases by 4.5%. Similarly, in “case30_as_api”, 7.3% of the lines have different switching decisions, leading to a 79.0% cost increase when using the “E” solution. Therefore, by tightening the relaxation, we make better decisions and obtain better approximations of the true cost after line switching.

8.2. Results with the Maximum Spanning Tree Heuristic

Table 3 compares results of our ACOTS relaxations with and without the maximum spanning tree heuristic proposed in Section 6. We include PGLib instances with up to 300 buses, which are solved with the same setups as in Section 8.1. In addition, when using the heuristic, we solve one “ECB” instance (“case30_as_api”) and the three 200-bus “ECB*” instances with Gurobi’s presolve disabled and with one thread. This is necessary to avoid the solver either incorrectly terminating with a locally optimal solution or incorrectly reporting infeasibility.

The “Objective Difference” values are calculated by $100 \times (LB_2 - LB_1) / LB_1$, where LB_1 and LB_2 are respectively the optimal value with and without the heuristic. We use “s.” for cases solved with the heuristic but not without it, “ws.” for cases solved without the heuristic but not with it, and “ns.” for cases not solved by either method. “Runtime Saved” shows the saved runtime by using the heuristic.

With the heuristic, we can solve many instances significantly faster, and solve 12 more instances within the 0.1% tolerance. The heuristic can lead to higher optimal values for some instances. The objective difference is generally smaller for larger cases, suggesting that their accuracy is less affected by the heuristic. This is probably because larger networks

Table 3 Comparing the objective difference and saved runtime of ACOTS relaxations with and without the spanning tree heuristic (“s.”: solved with heuristic and unsolved without; “ns.”: unsolved with either method; “ws.” solved without the heuristic and unsolved with it).

Test Case	Objective Difference (%)				Runtime Saved (seconds)			
	E	EC	ECB	ECB*	E	EC	ECB	ECB*
Typical Operating Conditions (TYP)								
case3_lmbd	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.2
case5_pjm	0.0	0.0	0.1	0.1	0.0	0.1	0.1	-0.9
case14_ieee	0.0	0.0	0.0	0.0	0.0	0.1	-0.2	2.4
case24_ieee_rts	0.0	0.0	0.0	0.0	-0.4	304.2	339.9	2.7
case30_as	0.0	0.0	0.0	0.0	18.9	433.1	593.7	-15.7
case30_ieee	0.0	0.0	20.2	20.2	0.4	1.2	-2.1	6.7
case39_epri	0.0	0.0	0.0	0.1	0.1	0.2	-1.2	3.3
case57_ieee	0.0	0.0	0.0	0.0	14.4	31.6	56.9	17.2
case73_ieee_rts	0.0	0.0	0.0	0.0	13.6	4796.2	5037.2	-1.9
case89_pegase	0.0	ns.	ns.	ns.	4667.2	-	-	-
case118_ieee	0.0	0.1	0.2	0.1	364.1	1251.9	-334.1	1200.9
case162_ieee_dtc	s.	ns.	ns.	ns.	-	-	-	-
case179_goc	0.0	0.0	0.0	0.0	296.8	129.0	2.2	-220.0
case200_activ	0.0	0.0	0.0	s.	2654.5	4307.8	-991.4	-
case240_pserc	0.0	0.0	ns.	0.2	4653.3	1778.3	-	0.0
case300_ieee	ns.	ns.	ns.	ns.	-	-	-	-
Small Angle Difference Conditions (SAD)								
case3_lmbd_sad	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3
case5_pjm_sad	0.0	0.0	0.1	0.1	0.0	0.0	0.0	-0.1
case14_ieee_sad	0.4	0.4	0.5	0.6	0.3	0.5	1.7	0.0
case24_ieee_rts_sad	1.1	1.3	1.9	2.0	13.4	78.8	94.8	14.6
case30_as_sad	0.0	0.0	1.1	1.1	7.9	27.7	30.1	6.3
case30_ieee_sad	3.5	3.6	0.1	0.1	1.9	1.4	-1.4	-0.3
case39_epri_sad	0.4	0.4	0.5	0.5	14.8	12.9	13.4	7.6
case57_ieee_sad	0.0	0.0	0.1	0.1	39.9	121.0	135.7	67.3
case73_ieee_rts_sad	s.	s.	s.	s.	-	-	-	-
case89_pegase_sad	0.0	ns.	ns.	ns.	0.0	-	-	-
case118_ieee_sad	1.4	1.4	ns.	1.7	6416.4	-312.4	-	1261.6
case162_ieee_dtc_sad	ns.	ns.	ns.	ns.	-	-	-	-
case179_goc_sad	0.1	0.1	ws.	0.2	6885.0	5242.6	-	5368.4
case200_activ_sad	0.0	0.0	0.0	s.	6980.2	6292.4	4481.7	-
case240_pserc_sad	ns.	ns.	ns.	ns.	-	-	-	-
case300_ieee_sad	ns.	ns.	ns.	ns.	-	-	-	-
Congested Operating Conditions (API)								
case3_lmbd_api	1.2	2.0	5.6	5.6	0.0	0.1	-0.1	-1.1
case5_pjm_api	0.0	0.0	1.8	1.8	0.1	0.1	0.1	1.3
case14_ieee_api	0.0	0.0	0.3	0.4	0.1	0.3	-0.8	-0.4
case24_ieee_rts_api	3.5	3.7	12.1	12.0	3.1	77.7	22.0	15.1
case30_as_api	0.0	0.0	29.3	29.0	1.8	27.2	-27.0	2.6
case30_ieee_api	0.0	0.0	0.2	0.1	0.3	0.5	-0.8	0.0
case39_epri_api	0.0	0.0	1.5	1.5	0.3	1.0	2.5	7.1
case57_ieee_api	0.0	0.0	0.0	0.0	9.9	36.1	32.5	18.7
case73_ieee_rts_api	1.2	s.	7.1	7.2	3818.8	-	3068.8	294.7
case89_pegase_api	0.0	ns.	ns.	ns.	4415.6	-	-	-
case118_ieee_api	0.1	0.1	ws.	s.	7113.7	6853.6	-	-
case162_ieee_dtc_api	0.0	s.	ns.	s.	5543.4	-	-	-
case179_goc_api	0.0	0.1	0.0	0.0	729.2	7074.7	-492.9	925.7
case200_activ_api	0.0	0.0	0.0	0.0	2404.3	6913.0	2016.3	647.8
case240_pserc_api	0.0	0.0	s.	ws.	5559.7	0.0	-	-
case300_ieee_api	0.0	0.0	0.1	0.2	6316.0	5198.2	0.0	5655.5

tend to have more flexibility even after the heuristic fixes some lines to be switched on. There are three instances that can be solved without the heuristic, while the heuristic leads to relative gaps larger than 0.1% (and below 0.5%) at termination.

We also tested with assigning the same weight to all lines, but this approach does not lead to as much improvement as shown in Table 3.

8.3. Performance on Large-Scale ACOTS Instances

In this section, we present experimental results for large-scale PGLib instances with 500 to 2,312 buses. Note that to the best of our knowledge, the largest ACOTS-QC instances investigated in the literature are up to 300 buses (Hijazi et al. 2017, Bestuzheva et al. 2020). However, with our proposed relaxations and heuristic, we are able to obtain results for some of these larger instances.

In Table 4 we present results for the upper bound and the relaxations of ACOTS-QC. Here we include instances for which a feasible solution is found by either “P”, “E”, or “EC”, and provide the results for all instances in Table 12 of the Online Supplement. The “UB” and relaxation results are obtained with a similar method as in Section 2, with the following adjustment: solver time limits are set to 3 hours, and solver feasibility tolerances for the relaxations are set to 10^{-3} to reduce the likelihood of incorrectly rejecting warm-start solutions. Note that when using Gurobi’s default feasibility tolerance of 10^{-6} , none of the instances is solved. We include results for “P”, “E”, and “EC” relaxations. None of the “ECB” and “ECB*” relaxation instances is solved within the time limit. The performance measures we use include the objective value and runtime (or the relative gap at termination for cases hit the time limit). We use “nf.” as the objective value if no feasible solution is found within the time limit.

Table 4: Objective value, runtime, and relative gap of ACOTS relaxations for instances with 500 to 2,312 buses (“nf.”: no feasible solution within time limit).

Test Case	UB	Objective Value			Runtime (seconds) (Relative Gap)		
		P	E	EC	P	E	EC
case500_goc_sad	487397.2	nf.	455966.2	nf.	-	(0.5%)	-
case1354_pegase_sad	1258848.0	nf.	1239689.5	1239701.2	-	(0.1%)	(0.2%)
case2000_goc_sad	992879.8	nf.	nf.	979501.4	-	-	(65.1%)
case500_goc_api	675967.9	nf.	668613.9	668617.9	-	2332.84	9289.02
case588_sdet_api	391951.0	389971.1	nf.	nf.	(0.2%)	-	-
case1354_pegase_api	1498271.0	nf.	nf.	1490268.6	-	-	(0.1%)
case2000_goc_api	1468630.3	nf.	nf.	1438342.5	-	-	(86.2%)
case2312_goc_api	571446.1	nf.	496817.5	nf.	-	(26.3%)	-
case500_goc	454946.0	nf.	453852.0	nf.	-	8897.06	-

case588_sdet	310016.0	307339.6	nf.	nf.	(0.1%)	-	-
case793_goc	259097.8	256776.8	nf.	nf.	(0.0%)	-	-
case1354_pegase	1258844.0	nf.	1239314.7	1239327.2	-	1770.48	(0.1%)
case1951_rte	2085530.3	nf.	2083344.3	nf.	-	(0.1%)	-
case2000_goc	973432.5	nf.	970466.8	nf.	-	(57.8%)	-

The state-of-the-art “P” relaxation solves 3 cases with at most 793 buses. In comparison, the “E” relaxation solves more cases to a below 0.5% relative gap, including some cases with over 1000 buses such as “case1354_pegase_sad” and “case1951_rte”. In particular, the objective value of “case1951_rte” is within 0.1% of UB, suggesting it is very close to the global optimal solution. The “EC” relaxation solves “case1354_pegase_api” within a 0.1% relative gap, while this case is not solved by “E”.

In Table 5, we also present results with the maximum spanning tree heuristic. Here we include instances that obtains a feasible solution from any one of the ACOTS-QC relaxations, and present the results for all instances in Table 13 of the Online Supplement. This method enables us to heuristically solve more large instances either to optimality or to within a 0.5% relative gap. Interestingly, “EC” can sometimes solve cases that “E” cannot solve, such as “case1354_pegase_api” and “case2000_goc_api”. This is likely due to the cycle constraints reduce the search space. The “ECB” and “ECB*” relaxations solve fewer instances compared with “E” and “EC”, which is possibly because more warm-start solutions being incorrectly rejected after the bounds are tightened.

Table 5: **With the spanning tree heuristic: the objective value, runtime, and relative gap of ACOTS relaxations for instances with 500 to 2,312 buses (“nf.”: no feasible solution within time limit).**

Test Case	Objective Value				Runtime (seconds) (Relative Gap)			
	E	EC	ECB	ECB*	E	EC	ECB	ECB*
case500_goc_sad	455966.2	nf.	nf.	nf.	376.13	-	-	-
case588_sdet_sad	309538.7	nf.	nf.	nf.	(0.6%)	-	-	-
case793_goc_sad	267942.0	268916.6	274171.5	273799.2	(1.4%)	(1.8%)	(3.3%)	(3.2%)
case1354_pegase_sad	1239689.5	1239701.2	nf.	nf.	(0.1%)	(0.1%)	-	-
case2000_goc_sad	nf.	979501.4	nf.	nf.	-	(65.1%)	-	-
case500_goc_api	668613.9	668617.9	nf.	nf.	8774.69	753.22	-	-
case588_sdet_api	389283.1	389286.7	390014.4	390166.5	3974.67	(0.0%)	(0.1%)	(0.0%)
case793_goc_api	nf.	276754.6	281581.3	283829.4	-	(0.3%)	(1.6%)	(2.4%)
case1354_pegase_api	1502166.8	1490268.6	nf.	nf.	(0.8%)	(0.0%)	-	-
case2000_goc_api	nf.	1438342.5	nf.	nf.	-	286.95	-	-
case2312_goc_api	496817.5	nf.	nf.	nf.	(26.3%)	-	-	-
case500_goc	453852.0	nf.	nf.	nf.	3512.71	-	-	-
case588_sdet	307165.3	307164.4	307746.1	307766.5	(0.1%)	(0.1%)	(0.1%)	(0.1%)

case793_goc	256772.0	256772.4	256852.8	256908.6	(0.0%)	(0.0%)	(0.0%)	(0.0%)
case1354_pegase	1239314.7	1239327.2	nf.	nf.	(0.1%)	(0.1%)	-	-
case2000_goc	970466.8	nf.	nf.	nf.	(57.8%)	-	-	-

In sum, the “E” and “EC” relaxations with the max spanning tree heuristic are very useful in obtaining results for larger PGLib cases. Given the critical role of warm-start solution in performance, it is promising that more large cases can be handled once the solver can effectively adopt feasible warm-start solutions.

8.4. 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.

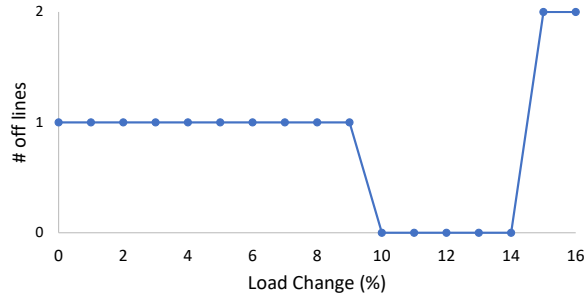


Figure 4 Number of lines switched off with different load levels.

8.5. 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 switched on. These cycle constraints are linearized using the strong extreme-point representation, which is also new. We experiment with “TYP”, “SAD” and “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 6, and include the results for all tested cases in Table 14 of the Online Supplement. 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.

Table 6 Comparing optimality gaps (%) for “E” and “EC” relaxations for ACOPF-QC. Including all cases with improvement greater than 0.03%.

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

9. 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. We also speed up the solution with a maximum spanning tree-based heuristic. 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. We also experiment with large-scale instances that are unexplored in the literature, demonstrating the usefulness of our relaxations and heuristic. Our proposed lifted cycle constraints improve bounds of ACOTS-QC as well as ACOPF-QC relaxations, and are helpful for solving more large-scale instances. We believe that our methods and findings make an important contribution towards solving ACOTS problems of more practical scale.

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. It would also be interesting to improve the maximum spanning tree heuristic, further reducing binary switching decisions while maintaining high

accuracy. To address scaling issues, it would be useful to (i) balance the trade-off between run time and tighter formulations; (ii) develop faster decomposition-based distributed algorithms; and (iii) study more scalable ACOPF relaxations.

Acknowledgments

The authors gratefully acknowledge support from the U.S. Department of Energy through Los Alamos National Laboratory's Directed Research and Development (LDRD) program, under project 20230091ER: Learning to Accelerate Global Solutions for Non-Convex Optimization. The authors also thank Clemson University for the allocation of compute time on the Palmetto Cluster. Additionally, we thank the anonymous reviewers for their critical feedback, which improved the presentation of this paper.

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, Bent R (2012) Computationally efficient optimal transmission switching: Solution space reduction. *2012 IEEE Power and Energy Society General Meeting*, 1–8 (IEEE).
- 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.
- Byrd RH, Nocedal J, Waltz RA (2006) Knitro: An integrated package for nonlinear optimization. *Large-scale nonlinear optim.* 35–59.
- Cengil F, Nagarajan H, Bent R, Eksioglu S, Eksioglu B (2022) Learning to accelerate globally optimal solutions to the AC optimal power flow problem. *Electric Power Systems Research* 212:108275.
- 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.
- Fattahi S, Lavaei J, Atamtürk A (2018) A bound strengthening method for optimal transmission switching in power systems. *IEEE Transactions on Power Systems* 34(1):280–291.
- Fisher EB, O’Neill RP, Ferris MC (2008) Optimal transmission switching. *IEEE Trans. Power Syst.* 23(3):1346–1355.
- Fuller JD, Ramasra R, Cha A (2012) Fast heuristics for transmission-line switching. *IEEE Transactions on Power Systems* 27(3):1377–1386.
- 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.
- Guo C, Nagarajan H, Bodur M (2025) Tightening quadratic convex relaxations for the AC optimal transmission switching problem. URL <http://dx.doi.org/10.1287/ijoc.2023.0236.cd>, available for download at <https://github.com/INFORMSJoC/2023.0236>.
- 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.
- Hinneck A, Pozo D (2022) Optimal transmission switching: Improving solver performance using heuristics. *IEEE Transactions on Power Systems* 38(4):3317–3330.
- 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).
- Soroush M, Fuller JD (2013) Accuracies of optimal transmission switching heuristics based on DCOPF and ACOPF. *IEEE Transactions on Power Systems* 29(2):924–932.
- Sundar K, Nagarajan H, Misra S, Lu M, Coffrin C, Bent R (2023) Optimization-based bound tightening using a strengthened QC-relaxation of the optimal power flow problem. *62nd Conference on Decision and Control (CDC)*, 4598–4605 (IEEE), <https://doi.org/10.1109/CDC49753.2023.10384116>.
- 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.

Appendix A: Additional Experiment Results for ACOTS Instances Up to 300 Buses

Table 7: Relative gap (%) of ACOTS relaxations at termination (“nf.”: no feasible solution within time limit).

Test Case	P	E	EC	ECB	ECB*
case3_lmbd	0.0	0.0	0.0	0.0	0.0
case5_pjm	0.0	0.0	0.0	0.0	0.0
case14_ieee	0.0	0.0	0.0	0.0	0.0
case24_ieee_rts	0.0	0.0	0.0	0.0	0.0
case30_as	0.0	0.0	0.0	0.0	0.0
case30_ieee	0.0	0.0	0.0	0.0	0.0
case39_epri	0.0	0.0	0.0	0.0	0.0
case57_ieee	0.0	0.0	0.0	0.0	0.0
case73_ieee_rts	0.0	0.0	0.0	0.0	0.0
case89_pegase	0.1	0.1	nf.	nf.	nf.
case118_ieee	0.0	0.0	0.0	0.0	0.0
case162_ieee_dtc	0.1	0.1	0.2	0.6	0.2
case179_goc	0.0	0.0	0.0	0.0	0.0
case200_activ	0.0	0.0	0.0	0.0	nf.
case240_pserc	0.1	0.0	0.0	0.2	0.0
case300_ieee	nf.	2.6	3.0	1.8	nf.
case3_lmbd_sad	0.0	0.0	0.0	0.0	0.0
case5_pjm_sad	0.0	0.0	0.0	0.0	0.0
case14_ieee_sad	0.0	0.0	0.0	0.0	0.0
case24_ieee_rts_sad	0.0	0.0	0.0	0.0	0.0
case30_as_sad	0.0	0.0	0.0	0.0	0.0
case30_ieee_sad	0.0	0.0	0.0	0.0	0.0
case39_epri_sad	0.0	0.0	0.0	0.0	0.0
case57_ieee_sad	0.0	0.0	0.0	0.0	0.0
case73_ieee_rts_sad	0.3	0.1	0.2	0.5	0.2
case89_pegase_sad	0.0	0.1	nf.	nf.	nf.
case118_ieee_sad	0.0	0.0	0.0	0.1	0.0
case162_ieee_dtc_sad	0.2	0.3	0.4	0.6	0.4
case179_goc_sad	nf.	0.0	0.0	0.1	0.1
case200_activ_sad	0.0	0.0	0.0	0.0	nf.
case240_pserc_sad	nf.	0.3	0.8	0.4	nf.
case300_ieee_sad	nf.	1.9	2.8	2.8	2.0
case3_lmbd_api	0.0	0.0	0.0	0.0	0.0
case5_pjm_api	0.0	0.0	0.0	0.0	0.0
case14_ieee_api	0.0	0.0	0.0	0.0	0.0
case24_ieee_rts_api	0.0	0.0	0.0	0.0	0.0
case30_as_api	0.0	0.0	0.0	0.0	0.0
case30_ieee_api	0.0	0.0	0.0	0.0	0.0
case39_epri_api	0.0	0.0	0.0	0.0	0.0
case57_ieee_api	0.0	0.0	0.0	0.0	0.0
case73_ieee_rts_api	0.0	0.0	2.7	0.0	0.0
case89_pegase_api	nf.	0.1	nf.	nf.	nf.
case118_ieee_api	0.0	0.0	0.1	0.0	0.1
case162_ieee_dtc_api	0.1	0.1	0.2	0.9	0.2
case179_goc_api	0.0	0.0	0.0	0.0	0.0
case200_activ_api	0.0	0.0	0.0	0.0	0.0
case240_pserc_api	nf.	0.0	0.0	0.1	0.1
case300_ieee_api	0.0	0.0	0.0	0.0	0.0

Table 8: Separation problem building time (seconds) of ECB*.

Test Case	Time
case3_lmbd	1.56
case5_pjm	2.24
case14_ieee	5.56
case24_ieee_rts	6.42
case30_as	5.59
case30_ieee	7.62
case39_epri	8.08
case57_ieee	4.66
case73_ieee_rts	6.53
case89_pegase	4.73
case118_ieee	8.30
case179_goc	9.09
case200_activ	19.22
case3_lmbd_sad	1.54
case5_pjm_sad	1.77
case14_ieee_sad	3.10
case24_ieee_rts_sad	2.54
case30_as_sad	2.58
case30_ieee_sad	3.33
case39_epri_sad	2.65
case57_ieee_sad	3.25
case89_pegase_sad	5.30
case118_ieee_sad	6.47
case179_goc_sad	3.70
case200_activ_sad	7.04
case3_lmbd_api	0.23
case5_pjm_api	3.25
case14_ieee_api	3.18
case24_ieee_rts_api	17.61
case30_as_api	3.59
case30_ieee_api	3.57
case39_epri_api	6.88
case57_ieee_api	7.42
case73_ieee_rts_api	8.83
case89_pegase_api	6.21
case118_ieee_api	5.81
case162_ieee_dtc_api	7.03
case179_goc_api	13.66
case200_activ_api	20.32
case240_pserc_api	5.96
case300_ieee_api	7.03

Table 9: Optimality gap (%) of ACOTS relaxations at the root node (bold numbers: improved gaps after tightening the relaxation with “E”, “EC” and “ECB”; bold-italic numbers: “ECB*” optimality gaps can be improved with “ECB”).

Instance	P	E	EC	ECB	ECB*
case3_lmbd	3.9	3.9	3.8	2.8	<i>3.1</i>
case5_pjm	2.4	2.4	2.4	2.4	2.4
case14_ieee	5.9	5.9	5.9	5.8	5.8

case24_ieee_rts	4.1	4.1	4.1	4.1	4.1
case30_as	4.5	4.5	4.5	4.5	4.5
case30_ieee	25.8	25.8	25.8	25.1	25.1
case39_epri	1.2	1.2	1.2	1.2	1.2
case57_ieee	7.6	7.6	7.6	7.6	7.6
case73_ieee_rts	4.0	4.0	4.0	4.0	4.0
case89_pegase	1.8	1.8	1.8	1.8	1.8
case118_ieee	4.1	4.1	4.2	4.2	4.1
case179_goc	0.5	0.5	0.5	0.5	0.5
case200_activ	0.3	0.3	0.3	0.3	0.3
case3_lmbd_sad	6.1	5.8	5.1	1.6	1.6
case5_pjm_sad	4.5	4.5	2.1	1.1	3.0
case14_ieee_sad	24.8	24.8	24.8	9.5	12.5
case24_ieee_rts_sad	10.8	10.8	10.4	7.7	10.4
case30_as_sad	12.1	12.1	11.6	9.7	11.9
case30_ieee_sad	21.2	21.2	21.2	5.0	5.9
case39_epri_sad	1.5	1.5	1.5	1.4	1.4
case57_ieee_sad	2.5	2.5	2.5	2.4	2.5
case89_pegase_sad	1.7	1.7	1.7	1.7	1.7
case118_ieee_sad	5.0	5.0	5.0	5.0	5.0
case179_goc_sad	0.5	0.5	0.5	0.5	0.5
case200_activ_sad	0.3	0.3	0.3	0.3	0.3
case3_lmbd_api	6.6	6.5	6.1	0.4	7.0
case5_pjm_api	2.9	2.9	2.9	1.2	1.2
case14_ieee_api	22.1	22.1	22.1	11.3	16.2
case24_ieee_rts_api	17.2	17.2	17.2	14.0	16.3
case30_as_api	10.6	10.6	10.6	10.2	10.2
case30_ieee_api	15.1	15.1	15.1	5.7	7.5
case39_epri_api	2.2	2.2	2.2	2.1	2.1
case57_ieee_api	3.9	3.9	3.9	3.9	3.9
case73_ieee_rts_api	11.9	11.9	11.9	11.3	11.3
case89_pegase_api	2.4	2.5	2.5	2.5	2.5
case118_ieee_api	13.7	13.3	13.3	13.2	13.2
case162_ieee_dtc_api	6.7	6.7	6.8	6.7	6.7
case179_goc_api	11.7	11.7	11.7	1.7	1.7
case200_activ_api	2.0	2.0	2.0	2.0	2.0
case240_pserc_api	2.0	2.1	2.1	2.1	2.1
case300_ieee_api	4.6	4.6	4.6	4.6	4.4

Table 10: Optimality gap and runtime of ACOTS relaxations “PB” and “EB” (bold-italic numbers: optimality gaps can be improved after tightening the relaxation; “tl.”: hits the time limit).

Test Case	Optimality Gap (%)		Runtime (seconds)	
	PB	EB	PB	EB
case3_lmbd	0.2	0.1	0.03	0.03
case5_pjm	1.2	1.1	0.04	0.05
case14_ieee	0.1	0.1	0.28	0.42
case24_ieee_rts	0.0	0.0	0.74	1.70
case30_as	0.1	0.1	1.63	11.93
case30_ieee	11.7	11.0	1.20	2.36
case39_epri	0.1	0.0	0.36	0.62
case57_ieee	0.1	0.1	12.14	24.97

case73_ieee_rts	0.0	0.0	14.94	90.88
case89_pegase	0.1	0.1	tl.	tl.
case118_ieee	0.3	0.3	263.30	679.13
case179_goc	0.1	0.1	88.37	180.36
case200_activ	0.0	0.0	790.71	4200.74
case3_lmbd_sad	0.3	0.2	0.01	0.02
case5_pjm_sad	0.4	0.3	0.03	0.04
case14_ieee_sad	2.3	0.8	0.25	0.51
case24_ieee_rts_sad	2.3	0.9	3.36	11.33
case30_as_sad	1.9	1.1	2.94	16.89
case30_ieee_sad	0.5	0.2	0.92	2.02
case39_epri_sad	0.1	0.1	6.29	12.47
case57_ieee_sad	0.2	0.1	13.12	42.12
case89_pegase_sad	0.1	0.1	tl.	tl.
case118_ieee_sad	0.9	0.9	2297.54	tl.
case179_goc_sad	0.1	0.1	tl.	tl.
case200_activ_sad	0.0	0.0	479.80	1910.26
case3_lmbd_api	0.0	0.0	0.01	0.01
case5_pjm_api	0.7	0.4	0.04	0.05
case14_ieee_api	3.4	1.1	0.14	0.18
case24_ieee_rts_api	4.0	1.2	1.13	3.96
case30_as_api	5.3	5.3	1.22	8.01
case30_ieee_api	0.8	0.4	0.61	1.04
case39_epri_api	0.5	0.4	0.65	1.82
case57_ieee_api	0.1	0.0	13.73	27.64
case73_ieee_rts_api	2.9	1.2	64.52	107.39
case89_pegase_api	0.2	0.2	tl.	tl.
case118_ieee_api	6.3	6.1	1056.73	5482.16
case162_ieee_dtc_api	1.0	1.0	tl.	tl.
case179_goc_api	0.6	0.6	129.72	218.81
case200_activ_api	0.0	0.0	769.25	884.19
case240_pserc_api	0.6	0.6	tl.	tl.
case300_ieee_api	0.7	0.7	tl.	tl.

Table 11: Locally optimal objective values of the non-convex ACOTS model (1) obtained by Juniper and Knitro (bold numbers: better objective values; “nf.”: no feasible solution within time limit).

Test Case	Juniper	Knitro
case3_lmbd	5812.6	5812.6
case5_pjm	15174.0	15174.0
case14_ieee	2178.1	2178.1
case24_ieee_rts	63352.2	63352.2
case30_as	803.1	803.1
case30_ieee	7579.0	7579.0
case39_epri	137728.7	137746.4
case57_ieee	37559.3	37559.5
case73_ieee_rts	189764.1	189764.1
case89_pegase	106656.2	106623.0
case118_ieee	96739.8	96730.7
case179_goc	nf.	754083.7
case200_activ	27557.6	27558.3
case3_lmbd_sad	5959.3	5959.3

case5_pjm_sad	26108.8	26108.8
case14_ieee_sad	2727.5	2768.5
case24_ieee_rts_sad	75794.0	76613.0
case30_as_sad	893.9	894.3
case30_ieee_sad	8188.6	8188.6
case39_epri_sad	147472.9	147908.9
case57_ieee_sad	38597.8	38597.8
case89_pegase_sad	nf.	106633.6
case118_ieee_sad	97572.5	100429.9
case179_goc_sad	nf.	762532.5
case200_activ_sad	27557.6	27558.3
case3_lmbd_api	10636.0	10636.0
case5_pjm_api	75190.3	75190.3
case14_ieee_api	5999.4	5999.4
case24_ieee_rts_api	119743.1	119743.1
case30_as_api	3065.8	2925.1
case30_ieee_api	17936.5	17936.5
case39_epri_api	246723.0	246723.0
case57_ieee_api	49271.9	49277.9
case73_ieee_rts_api	385277.3	387328.1
case89_pegase_api	100325.3	123031.4
case118_ieee_api	181535.8	210056.6
case162_ieee_dtc_api	116923.8	118378.1
case179_goc_api	nf.	1932020.5
case200_activ_api	35701.5	35700.8
case240_pserc_api	4640000.0	4638308.7
case300_ieee_api	nf.	684990.4

Appendix B: Results for All ACOTS Instances with 500 to 2312 Buses

Table 12: Objective value, runtime, and relative gap of ACOTS relaxations for all instances with 500 to 2312 buses (“nf.”: no feasible solution within time limit).

Test Case	UB	Objective Value			Runtime (seconds) (Relative Gap)		
		P	E	EC	P	E	EC
case500_goc_sad	487397.2	nf.	455966.2	nf.	-	(0.5%)	-
case588_sdet_sad	313797.2	nf.	nf.	nf.	-	-	-
case793_goc_sad	285798.4	nf.	nf.	nf.	-	-	-
case1354_pegase_sad	1258848.0	nf.	1239689.5	1239701.2	-	(0.1%)	(0.2%)
case1888_rte_sad	1413922.9	nf.	nf.	nf.	-	-	-
case1951_rte_sad	2089370.1	nf.	nf.	nf.	-	-	-
case2000_goc_sad	992879.8	nf.	nf.	979501.4	-	-	(65.1%)
case2312_goc_sad	461748.3	nf.	nf.	nf.	-	-	-
case500_goc_api	675967.9	nf.	668613.9	668617.9	-	2332.84	9289.02
case588_sdet_api	391951.0	389971.1	nf.	nf.	(0.2%)	-	-
case793_goc_api	318853.5	nf.	nf.	nf.	-	-	-
case1354_pegase_api	1498271.0	nf.	nf.	1490268.6	-	-	(0.1%)
case1888_rte_api	1951676.1	nf.	nf.	nf.	-	-	-
case1951_rte_api	2410827.1	nf.	nf.	nf.	-	-	-
case2000_goc_api	1468630.3	nf.	nf.	1438342.5	-	-	(86.2%)
case2312_goc_api	571446.1	nf.	496817.5	nf.	-	(26.3%)	-
case500_goc	454946.0	nf.	453852.0	nf.	-	8897.06	-
case588_sdet	310016.0	307339.6	nf.	nf.	(0.1%)	-	-

case793_goc	259097.8	256776.8	nf.	nf.	(0.0%)	-	-
case1354_pegase	1258844.0	nf.	1239314.7	1239327.2	-	1770.48	(0.1%)
case1888_rte	1375888.3	nf.	nf.	nf.	-	-	-
case1951_rte	2085530.3	nf.	2083344.3	nf.	-	(0.1%)	-
case2000_goc	973432.5	nf.	970466.8	nf.	-	(57.8%)	-
case2312_goc	441330.3	nf.	nf.	nf.	-	-	-

Table 13: With the spanning tree heuristic: the objective value, runtime, and relative gap of ACOTS relaxations for all instances with 500 to 2312 buses (“nf.”: no feasible solution within time limit).

Test Case	Objective Value				Runtime (seconds)		(Relative Gap)	
	E	EC	ECB	ECB*	E	EC	ECB	ECB*
case500_goc_sad	455966.2	nf.	nf.	nf.	376.13	-	-	-
case588_sdet_sad	309538.7	nf.	nf.	nf.	(0.6%)	-	-	-
case793_goc_sad	267942.0	268916.6	274171.5	273799.2	(1.4%)	(1.8%)	(3.3%)	(3.2%)
case1354_pegase_sad	1239689.5	1239701.2	nf.	nf.	(0.1%)	(0.1%)	-	-
case1888_rte_sad	nf.	nf.	nf.	nf.	-	-	-	-
case1951_rte_sad	nf.	nf.	nf.	nf.	-	-	-	-
case2000_goc_sad	nf.	979501.4	nf.	nf.	-	(65.1%)	-	-
case2312_goc_sad	nf.	nf.	nf.	nf.	-	-	-	-
case500_goc_api	668613.9	668617.9	nf.	nf.	8774.69	753.22	-	-
case588_sdet_api	389283.1	389286.7	390014.4	390166.5	3974.67	(0.0%)	(0.1%)	(0.0%)
case793_goc_api	nf.	276754.6	281581.3	283829.4	-	(0.3%)	(1.6%)	(2.4%)
case1354_pegase_api	1502166.8	1490268.6	nf.	nf.	(0.8%)	(0.0%)	-	-
case1888_rte_api	nf.	nf.	nf.	nf.	-	-	-	-
case1951_rte_api	nf.	nf.	nf.	nf.	-	-	-	-
case2000_goc_api	nf.	1438342.5	nf.	nf.	-	286.95	-	-
case2312_goc_api	496817.5	nf.	nf.	nf.	(26.3%)	-	-	-
case500_goc	453852.0	nf.	nf.	nf.	3512.71	-	-	-
case588_sdet	307165.3	307164.4	307746.1	307766.5	(0.1%)	(0.1%)	(0.1%)	(0.1%)
case793_goc	256772.0	256772.4	256852.8	256908.6	(0.0%)	(0.0%)	(0.0%)	(0.0%)
case1354_pegase	1239314.7	1239327.2	nf.	nf.	(0.1%)	(0.1%)	-	-
case1888_rte	nf.	nf.	nf.	nf.	-	-	-	-
case1951_rte	nf.	nf.	nf.	nf.	-	-	-	-
case2000_goc	970466.8	nf.	nf.	nf.	(57.8%)	-	-	-
case2312_goc	nf.	nf.	nf.	nf.	-	-	-	-

Appendix C: Experiment Results for All ACOPF Instances

Table 14: Comparing optimality gaps (%) for “E” and “EC” relaxations for all ACOPF-QC instances (Diff.= gap of “E”- gap of “EC”).

Test Case	E	EC	Diff.
case3_lmbd_sad	1.38	1.31	0.07
case5_pjm_sad	0.63	0.62	0.01
case14_ieee_sad	19.16	13.10	6.06
case24_ieee_rts_sad	2.74	2.20	0.53
case30_as_sad	2.30	2.26	0.04
case30_ieee_sad	5.66	5.66	0.00

case39_epri_sad	0.20	0.20	0.00
case57_ieee_sad	0.32	0.25	0.07
case73_ieee_rts_sad	2.37	1.80	0.57
case89_pegase_sad	0.70	0.70	0.00
case118_ieee_sad	6.63	6.61	0.02
case162_ieee_dtc_sad	6.21	6.20	0.01
case179_goc_sad	0.99	0.97	0.02
case200_activ_sad	0.00	0.00	0.00
case240_pserc_sad	4.34	4.24	0.10
case300_ieee_sad	2.34	2.32	0.02
case500_goc_sad	6.45	6.44	0.01
case588_sdet_sad	5.94	5.94	0.00
case793_goc_sad	6.22	6.22	0.00
case1354_pegase_sad	1.52	1.52	0.00
case1888_rte_sad	2.80	2.80	0.00
case1951_rte_sad	0.42	0.42	0.00
case2000_goc_sad	1.35	1.35	0.00
case2312_goc_sad	3.39	3.37	0.02
case2383wp_k_sad	1.91	1.88	0.03
case2736sp_k_sad	1.31	1.31	0.00
case2737sop_k_sad	1.69	1.69	0.00
case2742_goc_sad	1.33	1.33	0.00
case2746wop_k_sad	1.96	1.96	0.00
case2746wp_k_sad	1.63	1.63	0.00
case2848_rte_sad	0.23	0.23	0.00
case2853_sdet_sad	1.64	1.64	0.00
case2868_rte_sad	0.55	0.55	0.00
case2869_pegase_sad	1.01	1.01	0.00
case3_lmbd_api	4.53	3.85	0.68
case5_pjm_api	4.09	4.09	0.00
case14_ieee_api	5.13	5.13	0.00
case24_ieee_rts_api	11.02	10.88	0.14
case30_as_api	44.60	44.60	0.00
case30_ieee_api	5.45	5.45	0.00
case39_epri_api	1.69	1.69	0.00
case57_ieee_api	0.08	0.08	0.00
case73_ieee_rts_api	9.52	9.31	0.22
case89_pegase_api	23.06	23.06	0.00
case118_ieee_api	29.61	29.60	0.01
case162_ieee_dtc_api	4.32	4.32	0.01
case179_goc_api	5.86	5.75	0.11
case200_activ_api	0.02	0.02	0.00
case240_pserc_api	0.62	0.62	0.00
case300_ieee_api	0.82	0.81	0.00
case500_goc_api	3.44	3.44	0.00
case588_sdet_api	1.39	1.39	0.00
case793_goc_api	13.19	13.19	0.00
case1354_pegase_api	0.54	0.53	0.00
case1888_rte_api	0.22	0.22	0.00
case1951_rte_api	0.51	0.50	0.00
case2000_goc_api	2.06	2.06	0.00
case2312_goc_api	13.07	13.07	0.00
case2383wp_k_api	0.00	0.00	0.00
case2736sp_k_api	10.83	10.82	0.00
case2737sop_k_api	5.88	5.88	0.00
case2742_goc_api	24.40	24.40	0.00

case2746wop_k_api	0.00	0.00	0.00
case2746wp_k_api	0.00	0.00	0.00
case2848_rte_api	0.25	0.25	0.00
case2853_sdet_api	1.91	1.91	0.00
case2868_rte_api	0.17	0.17	0.00
case2869_pegase_api	0.99	0.98	0.00
case3_lmbd	0.97	0.97	0.00
case5_pjm	14.54	14.53	0.01
case14_ieee	0.11	0.11	0.00
case24_ieee_rts	0.01	0.01	0.00
case30_as	0.06	0.06	0.00
case30_ieee	18.67	18.67	0.00
case39_epri	0.54	0.54	0.00
case57_ieee	0.16	0.16	0.00
case73_ieee_rts	0.03	0.03	0.00
case89_pegase	0.75	0.74	0.00
case118_ieee	0.77	0.76	0.01
case162_ieee_dtc	5.84	5.83	0.00
case179_goc	0.15	0.15	0.00
case200_activ	0.00	0.00	0.00
case240_pserc	2.72	2.72	0.00
case300_ieee	2.56	2.55	0.00
case500_goc	0.24	0.24	0.00
case588_sdet	1.91	1.91	0.00
case793_goc	1.32	1.32	0.00
case1354_pegase	1.55	1.55	0.00
case1888_rte	2.04	2.04	0.00
case1951_rte	0.13	0.13	0.00
case2000_goc	0.30	0.30	0.00
case2312_goc	1.89	1.89	0.00
case2383wp_k	0.96	0.96	0.00
case2736sp_k	0.30	0.29	0.00
case2737sop_k	0.26	0.25	0.01
case2742_goc	1.32	1.32	0.00
case2746wop_k	0.35	0.35	0.00
case2746wp_k	0.31	0.31	0.00
case2848_rte	0.12	0.12	0.00
case2853_sdet	0.86	0.86	0.00
case2868_rte	0.09	0.09	0.00
case2869_pegase	1.00	1.00	0.00