informs
https://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH

Vol. 00, No. 0, Xxxxx 0000, pp. 000–000

Submitted to *Mathematics of Operations Research*

ISSN 0364-765X, EISSN 1526-5471

Compact MILP Relaxations of Symmetric Nonconvex Constraints

Gonzalo Gomez-Abejon
Department of Computational and Applied Mathematics and Operations Research, Rice University, gg51@rice.edu

Illya V. Hicks
Department of Computational and Applied Mathematics and Operations Research, Rice University, ivhicks@rice.edu

Abstract. In optimization, under multiple nonconvex constraints, mixed integer linear programming relaxations can be used to obtain arbitrarily tight dual bounds. Previous work has studied how to reduce the number of variables and constraints used to formulate those relaxations. We propose a general construction to obtain compact mixed integer linear programming formulations that generalizes previous results and extends them to other types of constraints. We also prove that some of them are minimal or close to minimal in size among those that achieve the same approximation error, and compare their performance with that of larger formulations in small instances of the AC optimal power flow problem.

Funding: This research was supported by grant N000142412648 of the ONR.

Key words: Mixed integer programming, Nonconvex programming, Disjunctive programming, Lifting, Optimization

1. Introduction

Approximations and relaxations of nonconvex constraints have been studied extensively, and a well-known tool to formulate them is disjunctive programming, as detailed in [15]. The general approach is to perform some sort of spatial branching, partitioning the domain into smaller convex regions, and constructing a linear approximation or relaxation of the feasible set in each region. The problem is then to find a union of convex polyhedra that approximates or contains the feasible region, and optimize over that set instead, using a disjunctive program or a mixed integer linear program (MILP).

There are general approaches for regions in standard form $\bigcap_j \{x \in \mathbb{R}^n : g_j(x) \leq 0\}$. Piecewise McCormick relaxations involve dividing the domain of the functions into hyperrectangles $\prod_{i=1}^n [l_i, u_i]$ and finding convex envelopes for each function g_j using the bounds of the variables

[19]. These envelopes have also been studied for specific constraints, including quadratic [2, 3], bilinear [8, 9] and multilinear [18].

There are procedures for finding disjunctive programs or MILPs that represent general unions of polyhedra, such as those summarized in [20]. If we have n polytopes given by linear constraints, their union can be formulated as a MILP with big-M constraints and $\lceil \log_2 n \rceil$ binary variables. If they are given as a convex hull of points, we can instead use combinatorial disjunctive constraints (CDCs) to obtain MILP formulations [4, 10], and in most practical cases the number of binary variables needed is also $\lceil \log_2 n \rceil$ [12, 16].

However, in both cases, either the number of continuous variables or the number of constraints needed grows linearly in n. Therefore, there has been some interest in formulations that only require $O(\log n)$ total constraints and variables, which is possible for bilinear constraints [8] and has also been described for quadratic constraints [3].

In Section 2, we give a construction based on Ben-Tal and Nemirovsky's approximation of the second-order cone [5] that produces relaxations of general symmetrical nonconvex constraints. We find $O(\log n)$-sized formulations for the constraints that were known to have one and many others. In Section 3, we prove that, for most of those constraints, the total size of our formulation is optimal or close to optimal. Finally, we compare the proposed formulations with optimal CDC formulations in Section 4, and draw conclusions in Section 5.

2. General Construction

We start with some definitions:

DEFINITION 1. We say that a constraint relaxes another when the set defined by the former contains the set defined by the latter. If $A \supseteq B$, then A or $x \in A$ is a relaxation of B or $x \in B$, and we will use the set or the constraint interchangeably from now on.

DEFINITION 2. We say that a continuous function $f : \mathbb{R}^n \to \mathbb{R}_+$ is an approximation error of S if $f(x) = 0$ for every $x \in S$, and $S = \lim_{\varepsilon \to 0^+} \{x \in \mathbb{R}^n : f(x) < \varepsilon\}$. For instance, $|f_1|$ and $\max(f_2, 0)$ are approximation error functions of $S_1 = \{x : f_1(x) = 0\}$ and $S_2 = \{x : f_2(x) \leq 0\}$.

Given a relaxation $A \supseteq B$, and an approximation error f of B, we say $\max\{f(x) : x \in A\}$ is the relaxation error of A. Since the minimum of a function over A is lower than the minimum over B, a solving a relaxation of a problem (relaxing the feasible region) produces a dual bound for the original problem. Lower relaxation errors tend to produce tighter dual bounds, and therefore it is natural to look for the simplest possible relaxation with the smallest possible relaxation error.

Suppose that a set $S = f([0,T], Y) = \{f(t, y) : 0 \leq t \leq T, y \in Y\}$ is parametrized by a function $f : \mathbb{R} \times \mathbb{R}^{m'} \mapsto \mathbb{R}^m$ where $Y \subseteq \mathbb{R}^{m'}$. If there are affine transformations g_t satisfying $g_t(f([0,t], Y)) =$

$f([t,2t],Y)$ for each $t \in [0,T]$ (or even just for $t = 2^{-k}T$, where $k \in \mathbb{N}$), we can obtain an envelope for S from any relaxation of a small subset $f([0,2^{-\nu}T],Y)$. If $P = \{Ax \le b\}$ is a polyhedral relaxation of $f([0,2^{-\nu}T])$ for some $\nu \in \mathbb{N}$, then the system

$$\begin{cases} x^{(i)} = g_{2^{1-i}T}(x^{(i-1)}) \quad \text{or} \quad x^{(i)} = x^{(i-1)} & \forall i \in \{1 \dots \nu\} \\ x^{(\nu)} \in P \end{cases} \tag{1}$$

gives a relaxation of $x^{(0)} \in S$ when projected in the space of the variable $x^{(0)}$.

THEOREM 1. *If Π_ν is the set defined by the system (1), then $S \subseteq \{x^{(0)} : (x^{(0)}, \dots x^{(\nu)}) \in \Pi_\nu\}$.*

Proof Given any point $x^{(0)} \in S$, we have $x^{(0)} = f(t,y)$. If $t \in [0,t]$, then $x^{(1)} = x^{(0)}$ satisfies the right-hand side of the first disjunction in (1). If instead $t \in [\frac{1}{2}T, T]$, by definition of g_t there exist some $t' \in [0,t]$ and $y' \in Y$ with $f(t',y') = g_T(f(t,y)) = g_T(x^{(0)})$, so choosing $x^{(1)} = f(t',y')$ satisfies the left-hand side of the disjunction. We can repeat this process, dividing the interval in half to obtain $x^{(2)}, x^{(3)} \dots x^{(\nu)}$ with $x^{(i)} \in f([0,2^{-i}T],Y)$ at each step, which is feasible in the system. $\square$

We know that the projection $\{x^{(0)} : (x^{(0)}, x^{(1)}, \dots, x_\nu) \in \Pi_\nu\}$ is a union of 2^ν convex polyhedra, for instance $\bigcup_{j=0}^{2^\nu-1} P(j)$ where $P(\sum_i z_i 2^i) := \{x : x^{(i)} - A_i(z)x^{(i-1)} = 0 \ \forall i, Ax^{(\nu)} \le b\}$, and $A_i(z)$ is the matrix representation of $g_{2^{-i}T}$ if $z_i = 0$ and the identity if $z_i = 1$. In other words, the feasible region S is enclosed in the union of 2^ν polyhedra using $O(\nu)$ variables and constraints. This representation suggests the following big-M formulation, which is only correct under certain assumptions:

$$\begin{cases} \left\{ \begin{array}{l} -M_i z_i \le x^{(i)} - g_{2^{1-i}T}(x^{(i-1)}) \le M_i z_i \\ -(1-z_i)M_i \le x^{(i)} - x^{(i-1)} \le (1-z_i)M_i \\ z_i \in \{0,1\} \end{array} \right\} & \forall i \in \{1 \dots \nu\} \\ Ax^{(\nu)} \le b \end{cases} \tag{2}$$

If we assume that the affine maps g_t are bijective and P is a polytope, it is not hard to see that every polyhedron P_j defined above is also bounded, and therefore the linear expressions $x^{(i)} - g_{2^{1-i}T}(x^{(i-1)})$ and $x^{(i)} - x^{(i-1)}$ are bounded for every i. Then, Formulation (2) is equivalent to Formulation (1) for a big enough value of M, and we have a MILP relaxation of $x \in S$.

The maps g_t can be thought of as some affine symmetries of the function f, which describe the way in which any "piece" of the graph of f can be recovered from any other piece by an affine transformation. For example, if $f(t,y) \mapsto f(t+a,y)$ is affine, that is, if for every $a \in \mathbb{R}$ there are

$A(a) \in \mathbb{R}^{m\times m}$ and $b(a) \in \mathbb{R}^m$ such that $f(t+a, y) = A(a)f(t, y) + b(a)$ for every t and y, then $g_t : x \mapsto A(t)x + b(t)$ is an adequate choice.

As an example, the conical surface $x_3 = \sqrt{x_1^2 + x_2^2}$ can be parametrized by the angle of the first two coordinates, as $S = \{(y\cos(t), y\sin(t), y) : y \in \mathbb{R}^+, t \in [0, 2\pi]\}$. There is a rotational symmetry of sorts in $f : (t, y) \mapsto (y\cos(t), y\sin(t), y)$, since adding a to the first argument rotates the first two entries of the image by $R_a = \begin{pmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{pmatrix}$, and the affine map $g_t : (x_1, x_2, x_3) \mapsto \left(R_t \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, x_3 \right)$ can be used in the construction, producing the following system:

$$\begin{cases} x^{(i)} = \begin{pmatrix} R_{2^{2-i}\pi} & 0 \\ 0 & 1 \end{pmatrix} x^{(i-1)} \quad \text{or} \quad x^{(i)} = x^{(i-1)} & \forall i \in \{1 \dots v\} \\ \cos\left(2^{-v}\pi\right) {x^{(v)}}_3 \le \cos\left(2^{-v}\pi\right) {x^{(v)}}_1 + \sin\left(2^{-v}\pi\right) {x^{(v)}}_2 \le {x^{(v)}}_3 \\ 0 \le {x^{(v)}}_2 \le \tan\left(2^{1-v}\pi\right) {x^{(v)}}_1 \end{cases} \tag{3}$$

The last two lines define a relaxation of $f([0, 2^{1-v}\pi], \mathbb{R}^+)$, as we will show later. Hence, by Theorem 1, (3) is a relaxation of ${x^{(0)}}_3 = \sqrt{{x^{(0)}}_1^2 + {x^{(0)}}_2^2}$. Since all variables $x^{(i)}$ have the same third coordinate, we can use only one variable, which yields a construction similar to that of [5].

Symmetries of this type exist for many other families of constraints, including the following:

Table 1. Some constraints that admit affine symmetries

Constraint	Implicit function f	Symmetry
$x_1 = x_2 x_3$	$f(t, y) = (ty, t, y)$	$f(t+a, y) = \begin{pmatrix} 1 & 0 & a \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} f(t, y) + \begin{pmatrix} 0 \\ a \\ 0 \end{pmatrix}$
$x_2 = x_1^2$	$f(t) = (t, t^2)$	$f(t+a) = \begin{pmatrix} 1 & 0 \\ 2a & 1 \end{pmatrix} f(t) + \begin{pmatrix} a \\ a^2 \end{pmatrix}$
$x_3 = \sqrt{x_1^2 + x_2^2}$	$f(t, y) = (y\sin(t), y\cos(t), y)$	$f(t+a, y) = \begin{pmatrix} R_a & 0 \\ 0 & 1 \end{pmatrix} f(t, y)$
$x_1 + ix_2 = x_3 \exp(ix_4)$	$f(t, y) = (y\sin(t), y\cos(t), y, t)$	$f(t+a, y) = \begin{pmatrix} R_a & 0 \\ 0 & I_2 \end{pmatrix} f(t, y) + ae_4$
$x_2 = \exp(x_1)$	$f(t) = (t, \exp(t))$	$f(t+a, y) = \begin{pmatrix} 1 & 0 \\ 0 & \exp(a) \end{pmatrix} f(t) + \begin{pmatrix} a \\ 0 \end{pmatrix}$

Note that in all the examples above, Y is polyhedral (when y is omitted, Y can be taken to be a single point) and f is linear in y and differentiable in t, so as the interval $[0, 2^{-v}T]$ shrinks, we can find better polyhedral relaxations of $f([0, 2^{-v}T], Y)$. If this is the case, we get better overall relaxations of S by increasing v, but the size of the formulations grows logarithmically as a function of the number of polyhedra needed to cover the relaxed feasible set.

For the bilinear constraint $x_1 = x_2x_3$ over $\mathbb{R}\times[0,T]\times\mathbb{R}_+$, the set $\{(ty,t,y) : t\in[a,b], y\in\mathbb{R}\}$ can be enclosed in $\{x\in\mathbb{R}^3 : ax_3 \le x_1 \le bx_3, a\le x_2\le b\}$, and if x is in the second set, there is a bound $|x_1/x_3 - x_2| \le 2|a-b|$ on the relaxation error. This gives a formulation very similar to [9], where using $O(\nu)$ total variables and constraints we have a relaxation error of $O(2^{-\nu})$.

For the parabola $x_2 = x_1^2$, the triangle $\max(2ax_1 + a^2, 2bx_1 + b^2) \le x_2 \le ax_1 + bx_1 - ab$ encloses the arc $\{(t,t^2) : t\in[a,b]\}$, and with a relaxation error bounded by $|x_2 - x_1^2| \le \frac{1}{4}(a-b)^2$. In this case, the construction yields a formulation very similar to [3], with the exact same feasible area. In this case, using $O(\nu)$ total variables, we achieve an error of $O(4^{-\nu})$.

In both cases, we have generalized previously studied formulations and attained relaxations that only require $O(\log(\varepsilon^{-1}))$ variables and constraints to ensure an error of ε. In the next Section, we show that this is the most compact formulation up to a constant factor.

Finally, if Formulation (2) is a valid MILP relaxation for S, and the maps $g_t : x\mapsto A(t)x + b(t)$ are constructed from a symmetry $f(t+a,y) = A(a)f(t,y)+b(a)$, the polyhedral piece represented by some binary vector $z\in\{0,1\}^\nu$ contains $f([\frac{N}{2^\nu}T, \frac{N+1}{2^\nu}T],T)$, where $N = \sum_{i=1}^{\nu} 2^{\nu-i}z_i$. Sorting the binary vectors along the domain of t produces the binary representations of the natural numbers, just as in the "zig-zag" formulation presented in [12]. If, instead, we construct g_t as before for $A(\cdot)$ and $b(\cdot)$ such that $f(2a-t,y) = A(a)f(t,y)+b(a)$, the binary vectors form a binary reflected Gray code, and its branching behavior on the domain of t is also described in [12, 17]. This can also be accomplished by replacing $x_i = x_{i-1}$ as the right-hand side of the disjunction in (1) by $x_i = h_t(x_{i-1})$ where $h_t(f([0,t],Y)) = f([0,t],Y)$ (so that Theorem 1 still holds) and $h_t(f(a,y)) = f(t-a,y)$.

3. Optimal Formulations

Our starting point is a result in [5], where the authors show that the second-order cone $x_3^2 \ge x_1^2 + x_2^2$ can be approximated by the projection of a polyhedron defined by $O(\nu)$ variables and constraints, achieving a relaxation error of $\frac{\sqrt{x_1^2+x_2^2}-x_3^2}{\sqrt{x_1^2+x_2^2}} \le \cos^{-1}(\frac{\pi}{2^{\nu+1}}) - 1 = O(4^{-\nu})$. Furthermore, the authors show the following:

THEOREM 2 **(Ben-Tal and Nemirovsky [5]).** *If a polyhedron* $\left\{x\in\mathbb{R}^3 : A\begin{pmatrix} x \\ x' \end{pmatrix} \le b\right\}$ *satisfies* $x_3^2 \ge x_1^2 + x_2^2 \Rightarrow x\in\Pi'_\nu \Rightarrow (1+\varepsilon)x_3^2 \ge x_1^2 + x_2^2$, *then the dimensions of* A *are at least* $C\log(\varepsilon^{-1})$ *for some fixed positive constant* C.

This means that the size of the proposed formulation is within a constant factor of the most compact one, that is, the one with the least total variables and constraints. The formulations we propose are also optimal or within a constant factor of being optimal for several nonconvex constraints.

3.1. Conical Constraints We start with the complement of the cone, $x_3^2 \le x_1^2 + x_2^2$.

THEOREM 3. *Let $\theta_j := \frac{\pi}{2^{j-1}}$ and let $\Pi_{(v)}$ be the polyhedron defined by the following constraints:*

$$\begin{aligned} &(a)\begin{cases} x_1^{(j)} = \cos\left(\theta_j\right) x_1^{(j-1)} + \sin\left(\theta_j\right) x_2^{(j-1)} \\ x_2^{(j)} = \left|-\sin\left(\theta_j\right) x_1^{(j-1)} + \cos\left(\theta_j\right) x_2^{(j-1)}\right| \end{cases} \quad \forall j \in \{1,\ldots,v\} \\ &(b)\begin{cases} \cos\left(\theta_{v+1}\right) x_1^{(v)} + \sin\left(\theta_{v+1}\right) x_2^{(v)} \ge \cos\left(\theta_{v+1}\right) x_3 \ge 0 \\ 0 \le \cos\left(\theta_v\right) x_2^{(v)} \le \sin\left(\theta_v\right) x_1^{(v)} \end{cases} \end{aligned} \tag{4}$$

Then, for $v \ge 2$ the projection $\Pi'_{(v)} := \{(x^{(0)}, x_3) : x \in \Pi_{(v)}\}$ satisfies

$$0 \le x_3 \le \sqrt{x_1^2 + x_2^2} \quad \Rightarrow \quad x \in \Pi'_{(v)} \quad \Rightarrow \quad 0 \le \cos\left(\frac{\pi}{2^v}\right) x_3 \le \sqrt{x_1^2 + x_2^2} \tag{5}$$

Proof For the first implication, we use Theorem 1. $\left\{x \in \mathbb{R}^3 : 0 \le x_3 \le \sqrt{x_1^2 + x_2^2}\right\} = f(2\pi, \mathbb{R}^2_+)$ for the function $f(t,y) = (y_1\cos(t), y_1\sin(t), y_1 + y_2)$, and for the affine maps we can choose the functions $g_t : x \mapsto \begin{pmatrix} \cos(t) & \sin(t) & 0 \\ \sin(t) & -\cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix} x$, and $h_t : x \mapsto \begin{pmatrix} \cos(t) & \sin(t) & 0 \\ -\sin(t) & \cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix} x$ All the third coordinates are equal, so we can use only one variable x_3. We only need to show that the inequalities (b) hold in $f([0, 2^{-v}2\pi], \mathbb{R}^2_+) = f([0,\theta_v], \mathbb{R}^2_+)$. If $(x^{(v)}{}_1, x^{(v)}{}_2, x_3) \in f([0,\theta_v], \mathbb{R}^2_+)$, the angle of the point $x^{(v)}$ is between 0 and $\theta_v = \frac{\pi}{2^{v-1}}$, so the second constraint holds. Finally,

$$2\sin(\theta_{v+1})\cos(\theta_{v+1})x^{(v)}{}_1 = \sin(\theta_v)x^{(v)}{}_1 \ge \cos(\theta_v)x^{(v)}{}_2 = (\cos(\theta_{v+1})^2 - \sin(\theta_{v+1})^2)x^{(v)}{}_2$$

$$\left(\cos(\theta_{v+1})x^{(v)}{}_1 + \sin(\theta_{v+1})x^{(v)}{}_2\right)^2 = \cos(\theta_{v+1})^2 x^{(v)2}_1 + 2\cos(\theta_{v+1})\sin(\theta_{v+1})x^{(v)}{}_1 x^{(v)}{}_2 + \sin(\theta_{v+1})^2 x^{(v)2}_2 \ge$$

$$\ge \cos(\theta_{v+1})x^{(v)2}_1 + (\cos(\theta_{v+1})^2 - \sin(\theta_{v+1})^2)x^{(v)2}_2 + \sin(\theta_{v+1})x^{(v)2}_2 = \cos(\theta_{v+1})^2(x^{(v)2}_1 + x^{(v)2}_2) = \cos(\theta_{v+1})^2 x_3^2$$

Since $x_3, x^{(v)}{}_2$ and $x^{(v)}{}_1$ are positive if $v \ge 2$, we can remove the squares and the first inequality in (b) holds as well.

On the other hand, if $(x_1^{(0)}, x_2^{(0)}, x_3) \in \Pi'_{(v)}$, then by the restrictions in (a),

$$(x_0^{(j)})^2 + (x_1^{(j)})^2 = \left(\cos\left(\theta_j\right) x_0^{(j-1)} + \sin\left(\theta_j\right) x_1^{(j-1)}\right)^2 + \left(-\sin\left(\theta_j\right) x_0^{(j-1)} + \cos\left(\theta_j\right) x_1^{(j-1)}\right)^2 = (x_0^{(j-1)})^2 + (x_1^{(j-1)})^2$$

So by induction in j, we get $(x_1^{(0)})^2 + (x_2^{(0)})^2 \ge \cdots \ge (x_1^{(v)})^2 + (x_2^{(v)})^2$, and by the scalar product inequality and constraints (b),

$$(x_1^{(v)})^2 + (x_2^{(v)})^2 = \|(x_1^{(v)}, x_2^{(v)})\|_2^2 \cdot \|(\cos(\theta_{v+1}), \sin(\theta_{v+1}))\|_2^2 \ge (\cos(\theta_{v+1})x_1^{(v)} + \sin(\theta_{v+1})x_2^{(v)})^2 \ge (\cos(\theta_{v+1})x_3)^2$$

and thus $\sqrt{(x_1^{(0)})^2 + (x_2^{(0)})^2} \ge \cos(\theta_{v+1})x_3 = \cos\left(\frac{\pi}{2^v}\right)x_3$. □

The bottom constraint in (a) is really a disjunctive constraint, since it is equivalent to either $x_2^{(j)} = -\sin(\theta_j)\, x_1^{(j-1)} + \cos(\theta_j)\, x_2^{(j-1)}$ or $x_2^{(j)} = \sin(\theta_j)\, x_1^{(j-1)} - \cos(\theta_j)\, x_2^{(j-1)}$ being true, which is the disjunction we get from our choice of f and g_t. We will see that Formulation (4) achieves the lowest error possible for a formulation with v disjunctions.

THEOREM 4. *Let $P = \bigcap_{i=1}^{v} \left(Q_i^0 \cup Q_i^1\right)$ be a set delimited by v binary disjunctions where $Q_i^j \subseteq \mathbb{R}^n$ are polyhedra, and $P' := \{(x_1,x_2,x_3) : x \in P\}$ its projection. If P' contains the cone complement $\{\sqrt{x_1^2+x_2^2} \geq x_3 \geq 0\}$ but not the smaller cone $\{0 \leq (1-\varepsilon)x_3 \geq \sqrt{x_1^2+x_2^2}\}$ for some $\varepsilon > 0$, then $\varepsilon > 1 - \cos\left(\frac{\pi}{2^v}\right)$.*

Proof We argue by contradiction and suppose that P exists. Then, intersecting P with the plane $x_3 = 1$ we obtain a set $\bar{P}$ such that $\{x \in \mathbb{R}^2 : x_1^2 + x_2^2 \geq 1\} \subset \{(x_1,x_2) : x \in \bar{P}\}$ and $x \in \bar{P} \Rightarrow x_1^2 + x_2^2 \geq (1-\varepsilon)^2$ for some $\varepsilon < 1 - \cos\left(\frac{\pi}{2^v}\right)$. We know $\bar{P} = \bigcap_{i=1}^{v} \left(\bar{Q}_i^0 \cup \bar{Q}_i^1\right)$ for $\bar{Q}_i^j = Q_i^j \cap \{x_3 = 1\}$, so it can be decomposed as the union of 2^v polyhedrons:

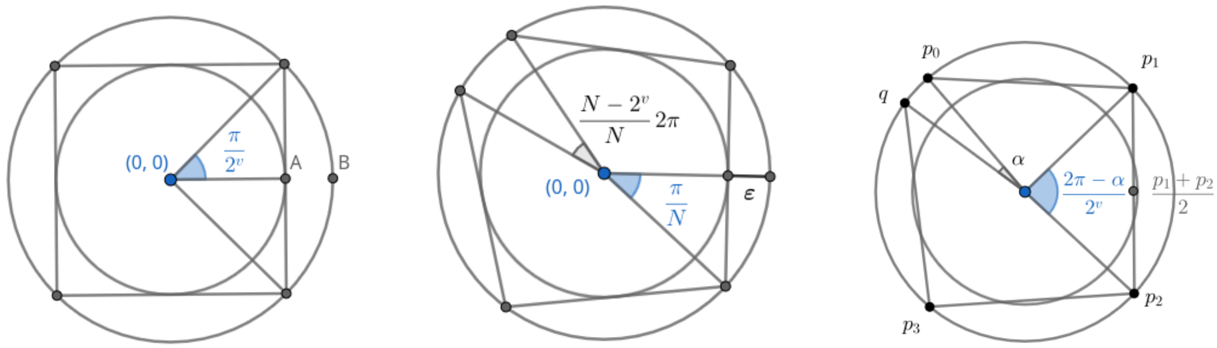
$$\bar{P} = \bigcup_{z \in \{0,1\}^v} \bigcap_{i=1}^{v} \bar{Q}_i^{z_i}$$

Therefore, the projection P' of $\bar{P}$ in the (x_1,x_2) plane can also be written as the union of the 2^v projections $P^j := \bigcap_{i=1}^{v} \bar{Q}_i^{z(j-1)_i}$ where $z(j-1) \in \{0,1\}^v$ is the binary representation of $j - 1 \in \{0, 1, \ldots 2^v\}$. Our goal is to show that each point in the circumference of radius 1 is only in one of these polyhedra, but any two points within an angle $\alpha > 0$ are in the same one, which is impossible.

Since $\varepsilon < 1 - \cos\left(\frac{\pi}{2^v}\right)$, we must have $\varepsilon = 1 - \cos\left(\frac{\pi}{N}\right)$ for some $N > 2^v$. Suppose two points p_0 and q in the circumference are at an angle of less than $\frac{N-2^v}{N} 2\pi$, that is, $p_0 = (\cos(\phi), \sin(\phi))$ and $q = (\cos(\phi+\alpha), \sin(\phi+\alpha))$, with $0 < \alpha < \frac{N-2^v}{N} 2\pi$. We consider the $2^v - 1$ points $p_i = (\cos(\phi - i\frac{2\pi-\alpha}{2^v}), \sin(\phi - i\frac{2\pi-\alpha}{2^v}))$ for $i = 1, 2 \ldots 2^v - 1$. The arc between p_i and p_j is at least $\frac{2\pi-\alpha}{2^v} > \frac{2\pi}{N}$ for any $0 \leq i < j \leq 2^v - 1$, so their midpoint is at most $\cos(\frac{\pi}{N}) = 1 - \varepsilon$ units away from the origin, and therefore not in P'.

This means no two points in $\{p_i : 0 \leq i \leq 2^v - 1\}$ lie in the same polyhedron P^j, and each of them must be in exactly one of the polyhedra. The same is true for $\{p_i : 1 \leq i \leq 2^v - 1\} \cup \{q\}$, which is the same as the previous set rotated an angle of $\frac{2\pi-\alpha}{2^v}$. Therefore, p_0 and q are in the same polyhedron and only in that one. This is true for any two points at an angle of less than $\frac{N-2^v}{N} 2\pi > 0$, so every point in the circumference is in the same P_i, and that is not possible since we saw that p_0 and p_1 were in different ones. □

FIGURE 1. Proof sketch for $v = 2$



(a) Optimal relaxation for $x^2 + y^2 \geq 1$, with error $d(A, B) = 1 - \cos\left(\frac{\pi}{2^v}\right)$

(b) Consecutive tangent chords for $\varepsilon < 1 - \cos\left(\frac{\pi}{2^v}\right)$

(c) Construction that shows p_0 and q are in the same polyhedron

COROLLARY 1. *Any disjunctive linear programming relaxation (DLPR) of $x_3 = \sqrt{x_1^2 + x_2^2}$ with v disjunctions can achieve an error $\left|\frac{x_3 - \sqrt{x_1^2+x_2^2}}{x_3}\right|$ of $1 - \cos\left(\frac{\pi}{2^v}\right)$ but not less.*

Proof If a relaxation with v disjunctions attained a lower error, adding the inequality $0 \leq x_4 \leq x_3$ we would have a relaxation for $0 \leq x_4 \leq \sqrt{x_1^2 + x_2^2}$ with the same error, which is not possible. On the other hand, an error of $1 - \cos\left(\frac{\pi}{2^v}\right)$ can be achieved by adding a sufficiently tight LP relaxation of $x_3 \geq \sqrt{x_1^2 + x_2^2}$ (like the one in [5]) to Formulation (4). □

COROLLARY 2. *Any disjunctive linear programming relaxation (DLPR) of $x_1^2 + x_2^2 = 1$ with v disjunctions can achieve an error $|\sqrt{x_1^2 + x_2^2} - 1|$ of $1 - \cos\left(\frac{\pi}{2^v}\right)$ but not less.*

Proof This is easy to see from the proof of Theorem 4, but it is also implied by Corollary 1, because if $\bigcap_{i=1}^{v}\{A_i x \leq b_i\} \cup \{A_i' x \leq b_i'\}$ achieves a certain relaxation error for $x_1^2 + x_2^2 = 1$, it is easy to see that $\bigcap_{i=1}^{v}\{A_i x \leq b_i y\} \cup \{A_i' x \leq b_i' y\}$ achieves the same error for $x_1^2 + x_2^2 = y^2$. □

3.2. Trigonometric Constraints The results for relaxations of the circumference also imply optimality for some trigonometric constraints. We start from the following relaxation of the helix:

$$
\begin{aligned}
&(a)\begin{cases} x_1^{(j)} = \cos(\theta_j)\, x_1^{(j-1)} + \sin(\theta_j)\, x_2^{(j-1)} \\ x_2^{(j)} = -\sin(\theta_j)\, x_1^{(j-1)} + \cos(\theta_j)\, x_2^{(j-1)} \\ x_4^{(j)} = x_4^{(j-1)} - \theta_j \end{cases} \text{ or } \begin{cases} x_1^{(j)} = \cos(\theta_j)\, x_1^{(j-1)} + \sin(\theta_j)\, x_2^{(j-1)} \\ x_2^{(j)} = \sin(\theta_j)\, x_1^{(j-1)} - \cos(\theta_j)\, x_2^{(j-1)} \\ x_4^{(j)} = -x_4^{(j-1)} + \theta_j \end{cases} \\
&(b)\begin{cases} \cos(\theta_v)\, x_3 \leq \cos(\theta_{v+1})\, x_1^{(v)} + \sin(\theta_{v+1})\, x_2^{(v)} \leq x_3 \\ 0 \leq \cos(\theta_v)\, x_2^{(v)} \leq \sin(\theta_v)\, x_1^{(v)} \\ 0 \leq x_4^{(v)} \leq \theta_v \end{cases}
\end{aligned}
\tag{6}
$$

THEOREM 5. *The projection in $x^{(0)}$ space of the set delimited by Formulation (6) contains the helix $H := \{x \in \mathbb{R}^4 : x_1 + ix_2 = x_3 \exp(ix_4)\}$ and satisfies both $|\angle(x_1 + x_2 i) - x_4| \leq \frac{\pi}{2^{v-1}}$ and $x_3^2 \cos\left(\frac{\pi}{2^v}\right)^2 \leq x_1^2 + x_2^2 \leq x_3^2 \cos\left(\frac{\pi}{2^v}\right)^{-2}$ for any $v \geq 2$.*

Proof We use Theorem 1 with $g_t : x \mapsto (\cos(t)x_1 + \sin(t)x_2, \sin(t)x_1 - \cos(t)x_2, x_3, x_4 - t)$ and $h_t : x \mapsto (\cos(t)x_1 + \sin(t)x_2, -\sin(t)x_1 + \cos(t)x_2, x_3, t - x_4)$ as the affine maps to certify that (6) is a relaxation of H, with the parametrization in table 1. The inequalities in (b) hold for $x^{(v)} \in f([0, 2^{-v}2\pi], \mathbb{R}^+)$: the third one is obvious, and we already showed that $\cos(\theta_v)\, x_2^{(v)} \leq \sin(\theta_v)\, x_1^{(v)}$ and $\cos(\theta_v)\, x_3 \leq \cos(\theta_{v+1})\, x_1^{(v)} + \sin(\theta_{v+1})\, x_2^{(v)}$ in the proof of Theorem 3. If we use $(x_3^{(v)})^2 = (x_2^{(v)})^2 + (x_1^{(v)})^2$, the only new inequality follows by the scalar product inequality $\cos(\theta_{v+1})x_1^{(v)} + \sin(\theta_{v+1})x_1^{(v)} \leq \sqrt{(x_1^{(v)})^2 + (x_2^{(v)})^2}\sqrt{\cos(\theta_{v+1})^2 + \sin(\theta_{v+1})^2} = x_3^{(v)}$.

On the other hand, if $(x^{(0)}, \ldots, x^{(v)})$ satisfies (6), the lower bound on $x_1^2 + x_2^2$ also follows from 3. For the upper bound, we use that for $v \geq 2$, $\cos(\theta_{v+1}) > 0 \Rightarrow x_2^{(v)} \geq 0$ and the angle of $\angle(x_1^{(v)}, x_2^{(v)})$ is between 0 and θ_v and then $x_3 \geq \cos(\theta_{v+1})x_1^{(v)} + \sin(\theta_{v+1})x_2^{(v)} \geq 0$. Squaring both quantities,

$$(x_3^{(v)})^2 \geq \cos(\theta_{v+1})^2(x_1^{(v)})^2 + \sin(\theta_{v+1})^2(x_2^{(v)})^2 + 2\cos(\theta_{v+1})x_1^{(v)} \sin(\theta_{v+1})x_2^{(v)}$$

$$\geq \cos(\theta_{v+1})^2(x_1^{(v)})^2 + \sin(\theta_{v+1})^2(x_2^{(v)})^2 + (\cos(\theta_{v+1})^2 - \sin(\theta_{v+1})^2)(x_2^{(v)})^2 = \cos(\theta_{v+1})^2({x_1^{(v)}}^2 + {x_2^{(v)}}^2)$$

since $2\sin(\theta_{v+1})\cos(\theta_{v+1})x^{(v)}{}_1 = \sin(\theta_v)x^{(v)}{}_1 \geq \cos(\theta_v)x^{(v)}{}_2 = (\cos(\theta_{v+1})^2 - \sin(\theta_{v+1})^2)x^{(v)}{}_2$, and by the inequalities in (a), $(x_1^{(0)})^2 + (x_2^{(0)})^2 = \cdots = (x_1^{(v)})^2 + (x_2^{(v)})^2 \leq \cos(\theta_{v+1})^{-2}x_3^2$.

Finally, by the inequalities in (a), $\angle(x_1^{(j)}, x_2^{(j)})$ and $x_4^{(j)}$ can be obtained from $\angle(x_1^{(v)}, x_2^{(v)})$ and $x_4^{(v)}$ respectively through a composition of the same sequence of functions $\alpha \mapsto \alpha + \theta_j$ or $\alpha \mapsto \theta_j - \alpha$ (depending on which clause of each disjunction is true). Since these maps preserve distances in $\mathbb{R}$, $|\angle(x_1^{(0)}, x_2^{(0)}) - x_4^{(0)}| = |\angle(x_1^{(v)}, x_2^{(v)}) - x_4^{(v)}| \leq \theta_v$, as (b) implies $\angle(x_1^{(v)}, x_2^{(v)}), x_4^{(v)} \in [0, \theta_v]$. □

If we set $x_3 = 1$ in Formulation (6), and we look only at the projection in the space $(x_1^{(0)}, x_4^{(0)})$, we get an envelope of $\{(\cos(\alpha), \alpha) : \alpha \in [0, 2\pi]\}$, and Theorem 5 gives us the following bound:

$$|x_1^{(0)} - \cos(x_4^{(0)})| \leq |x_1^{(0)} - \cos(\angle(x_1^{(0)}, x_2^{(0)}))| + |\cos(\angle(x_1^{(0)}, x_2^{(0)})) - \cos(x_4^{(0)})|$$

$$= \left|\sqrt{(x_1^{(0)})^2 + (x_2^{(0)})^2} - 1\right| |\cos(\angle(x_1^{(0)}, x_2^{(0)})| + |\sin(\beta)||\angle(x_1^{(0)}, x_2^{(0)}) - x_4^{(0)}| \leq \cos(\theta_v)^{-1} - 1 + \theta_v$$

using the Mean Value Theorem at some intermediate angle β. Since $\cos(\theta_v)^{-1} - 1 \approx \frac{1}{2}(\theta_v)^2 = O(4^{-v})$, the error is dominated by the second term $\theta_v = \frac{\pi}{2^{v-1}} = O(2^{-v})$. A similar argument shows that $|x_2^{(0)} - \sin(x_4^{(0)})| \leq \cos(\theta_v)^{-1} - 1 + \theta_v$. We will now see that, in order to obtain an error bound of $|x_1^{(0)} - \cos(x_4^{(0)})| \leq \varepsilon$, we need at least $-C\log(\varepsilon)$ disjunctions, where $C > 0$ is a fixed constant.

THEOREM 6. *If the two-dimensional projection of a set P defined by v binary disjunctions contains the set $\{(\cos(\alpha),\alpha) : \alpha \in [0,2\pi]\}$ and satisfies $|y-\cos(x)| \le \varepsilon$ for all $(x,y,u) \in P$, then $v \ge -C\log(\varepsilon)$ for some positive constant C.*

Proof We argue by contradiction, and assume that $P_\varepsilon(x,y,u)$ approximates $y=\cos(x)$ in $[0,2\pi]$ within error ε, using $v(\varepsilon)$ disjunctions, with $\lim_{\varepsilon\to 0}\frac{v(\varepsilon)}{-\log(\varepsilon)}=0$. Since $\sin(x)=\cos(x-\frac{\pi}{2})=\cos(x+\frac{3\pi}{2})$, a relaxation for $y=\sin(x)$ with the same error is given by the set $Q_\varepsilon(x,y,u) := \left(P(x-\frac{\pi}{2},y,u)\cup P(x+\frac{3\pi}{2},y,u)\right)\cap([0.2\pi]\times R^2)$, which can be defined with $2v(\varepsilon)+1$ disjunctions. We claim that $X_\varepsilon := \{x_1,x_2 : \exists(x_1,x_2,\alpha,u_1,u_2)\in P_\varepsilon(\alpha,x_1,u_1)\cap Q_\varepsilon(\alpha,x_2,u_2)\}$ contains the circumference $\{x_1^2+x_2^2=1\}$ and approximates it within an error of $2\varepsilon^2+4\varepsilon$.

Given a point $(x_1,x_2)=(\cos(\alpha),\sin(\alpha))$ in the circumference, there exist vectors u_1,u_2 for which $(\alpha,x_1,u_1)\in P_\varepsilon$ and $(\alpha,x_2,u_2)\in Q_\varepsilon$. On the other hand, for any $(x_1,x_2)\in X_\varepsilon$, there is an angle α and vectors u_1,u_2 such that $(\alpha,x_1,u_1)\in P_\varepsilon$ and $(\alpha,x_2,u_2)\in Q_\varepsilon$ and hence $|x_1^2+x_2^2-1| \le |x_1^2-\cos(\alpha)^2|+|x_2^2-\sin(\alpha)^2| \le \varepsilon|\cos(\alpha)+x_1|+\varepsilon|\sin(\alpha)+x_2| \le \varepsilon(2\cos(\alpha)+\varepsilon+2\sin(\alpha)+\varepsilon) \le 2\varepsilon^2+4\varepsilon$ as we wanted.

Finally, we showed in Corollary 2 that a relaxation of the circumference $x_1^2+x_2^2=1$ with w disjunctions can only achieve an error bound of $1-\sqrt{x_1^2-x_2^2} \le 1-\cos\left(\frac{\pi}{2^{w-1}}\right)$, which translates to $x_1^2+x_2^2 \ge \cos\left(\frac{\pi}{2^{w-1}}\right)^2$. For the set X, defined by $3v(\varepsilon)+1$ disjunctions, the error $|x_1^2+x_2^2-1|$ must be bigger than $1-\cos\left(\frac{\pi}{2^{3v(\varepsilon)}}\right)^2=\sin\left(\frac{\pi}{2^{3v(\varepsilon)}}\right)^2$. However, in that case

$$\lim_{\varepsilon\to 0}\frac{v(\varepsilon)}{-\log(\varepsilon)} = \lim_{\varepsilon\to 0}\frac{v(\varepsilon)}{\log(4)-\log(2\varepsilon^2+4\varepsilon)} \ge \lim_{\varepsilon\to 0}\frac{v(\varepsilon)}{\log(4)-\log(\max_{x\in X}|x_1^2+x_2^2-1|)}$$

$$\ge \lim_{\varepsilon\to 0}\frac{v(\varepsilon)}{\log(4)-\log(\sin(2^{-3v(\varepsilon)}\pi)^2)} = \lim_{\varepsilon\to 0}\frac{v(\varepsilon)}{\log(4)+6v(\varepsilon)\log(2)-2\log(\pi)} = \frac{1}{6\log(2)} > 0$$

which contradicts our hypothesis. □

3.3. Quadratic Constraints We conclude with similar results for the quadratic and bilinear constraints. The following is a relaxation of $x_2^{(0)}=(x_1^{(0)})^2$ over the interval $x_1^{(0)}\in[0,1]$:

$$\begin{array}{l}
\text{(a)}\begin{cases} x_1^{(j)} = \left|x_1^{(j-1)}-2^{-j}\right| \\ x_2^{(j)} = x_2^{(j-1)}-2^{1-j}x_1^{(j-1)}+4^{-j} \end{cases} \quad \forall j\in\{1,\dots,v\} \\
\text{(b)}\begin{cases} 0 \le x_1^{(v)} \le 2^{-j} \\ 0 \le 2^{-v}x_1^{(v)}-x_2^{(v)} \le 4^{-v-1} \end{cases}
\end{array} \tag{7}$$

THEOREM 7. *(7) is a relaxation of $x_2^{(0)} = (x_1^{(0)})^2$ and satisfies $|x_2^{(0)} - (x_1^{(0)})^2| \leq 4^{-v-1}$. This is the lowest error that we can achieve with v disjunctions.*

Proof We use Theorem 1 for $f : t \mapsto (t, t^2)$ and the maps $g_t : x \mapsto (x_1 - t, x_2 - 2tx_1 + t^2)$ and $h_t : x \mapsto (t - x_1, x_2 - 2tx_1 + t^2)$. Given a point $x^{(v)} \in \{(t, t^2) : t \in [0, 2^{-v}]\}$, obviously the first inequality in (b) is true, and the second follows from $x_2^{(v)} - 2^{-v}x_1^{(v)} = x_1^{(v)}(2^{-v} - x_1^{(v)}) \geq 0$ and $x_2^{(v)} - 2^{-v}x_1^{(v)} + 4^{-v-1} = (x_1^{(v)} - 2^{-v-1})^2 \geq 0$, so (7) is a relaxation of $x^{(0)} \in \{(t, t^2) : t \in [0, 1]\}$.

On the other hand, given a point $(x^{(0)}, \dots, x^{(v)})$ that satisfies (7), we have $\left|(x_1^{(v)})^2 - x_2^{(v)}\right| = \left|2^{-v}x_1^{(v)} - x_2^{(v)} - (2^{-v} - x_1^{(v)})x_1^{(v)}\right| \leq 4^{-v-1}$ since by (b), $x_1^{(v)} - x_2^{(v)} \in [0, 4^{-v-1}]$, and we already saw that $(2^{-v} - x_1^{(v)})x_1^{(v)} \in [0, 4^{-v-1}]$ for $x_1^{(v)} \in [0, 2^{-v}]$. Then, using (a), we see that $\left|x_2^{(j)} - (x_1^{(j)})^2\right| = \left|x_2^{(j-1)} - 2^{1-j}x_1^{(j-1)} + 4^{-j} - |x_1^{(j-1)} - 2^{-j}|^2\right| = \left|x_2^{(j-1)} - (x_1^{(j-1)})^2\right|$, and therefore $\left|x_2^{(0)} - (x_1^{(0)})^2\right| = \dots = \left|x_2^{(v)} - (x_1^{(v)})^2\right| \leq 4^{-v-1}$ as we needed.

Finally, if the union of 2^v polyhedra contains the parabola arc $\{(t, t^2) : t \in [0, 2^{-v}]\}$, in particular it contains the $2^v + 1$ points $\{(t, t^2) : 2^v t \in \{0, 1 \dots 2^v\}\}$, and by the pigeonhole principle, some polyhedron must contain two of them, say $(2^{-v}a, 4^{-v}a^2)$ and $(2^{-v}b, 4^{-v}b^2)$ with $|a - b| \geq 1$. In that case, it also contains the midpoint $(2^{-v-1}(a + b), 2^{-2v-1}(a^2 + b^2))$, but the error $|x_2 - x_1^2|$ at that point is $|2^{-2v-1}(a^2 + b^2) - 2^{-2v-2}(a + b)^2| = 4^{-v-1}|a^2 + b^2 - 2ab| = 4^{-v-1}(a - b)^2 \geq 4^{-v-1}$. □

COROLLARY 3. *Any DLPR of $x_3 = x_1x_2$ over the rectangle $(x_1, x_2) \in [l_1, u_1] \times [l_2, u_2]$ that satisfies $|x_3 - x_1x_2| \leq \varepsilon$ must have at least $C \log(\varepsilon^{-1})$ disjunctions, where C is a positive constant that depends on the domain.*

Proof From any DLPR of $x_3 = x_1x_2$ with error ε, we can obtain a relaxation of the parabola $z = y^2$ over $[0, 1]$ by adding the variables y and z together with the linear constraints $\frac{x_1 - l_1}{u_1 - l_1} = \frac{x_2 - l_2}{u_2 - l_2} = y$ and $z = \frac{x_3 - x_1x_2}{(u_1 - l_1)(u_2 - l_2)} + \frac{x_1 - l_1}{u_1 - l_1}\frac{x_2 - l_2}{u_2 - l_2}$. The new disjunctive relaxation satisfies $|z - y^2| = \frac{|x_3 - x_1x_2|}{(u_1 - l_1)(u_2 - l_2)} \leq \frac{\varepsilon}{(u_1 - l_1)(u_2 - l_2)}$, so by Theorem 7, we need at least $\log_4\left(\frac{(u_1 - l_1)(u_2 - l_2)}{\varepsilon}\right) - 1$ binary disjunctions, and that bound is linear on $\log(\varepsilon^{-1})$. □

The approximation error $|x_3/x_1 - x_2|$ has the same property if the domain of x_1 is bounded, because if a formulation satisfies $|x_3/x_1 - x_2| < \varepsilon$ we also have $|x_3 - x_1x_2| < M\varepsilon$ for $M = \max(|u_1|, |l_1|)$. This means the formulation in [8] is also within a constant factor of the most compact one.

Finally, any binary variable can be modeled with a binary disjunction. Therefore, all the previous results imply that a MILP relaxation of any of the sets discussed in this section also needs at least $O(\log(\varepsilon^{-1}))$ variables and constraints to get an approximation error of at most ε.

4. Computational Experiments In order to evaluate the practical performance of these formulations, we use them to implement a MILP relaxation of the AC optimal power flow problem (AC-OPF). An overview of the problem can be found in [13], and the test cases we will use are from the PGLib library [1] and NESTA [7]. MILP relaxations for this problem have already been proposed in [6] and implemented in [14], where the authors perform spatial branching in the domain of the voltage variables e_i and f_i, or the polar formulation variables c_{ij} and s_{ij}.

We first see that the standard AC-OPF problem can be formulated just in terms of convex constraints, conical surfaces and helices. The polar formulation in [13] only has the following two nonconvex constraints:

$$\mathbf{c}_{ij}^2 + \mathbf{s}_{ij}^2 = \mathbf{c}_{ii}\mathbf{c}_{jj} \qquad \forall (i,j) \in \mathcal{L} \tag{8}$$

$$\boldsymbol{\theta}_j - \boldsymbol{\theta}_i = atan2(\mathbf{s}_{ij}, \mathbf{c}_{ij}) := \angle(\mathbf{c}_{ij} + i\mathbf{s}_{ij}) \qquad \forall (i,j) \in \mathcal{L} \tag{9}$$

If we rewrite $c_{ij}^2 + s_{ij}^2 = c_i c_j$ as two conical surfaces $c_{ij}^2 + s_{ij}^2 = z^2$ and $(2z)^2 + (c_i - c_j)^2 = (c_i + c_j)^2$, then the arctangent constraint $\theta_j - \theta_i = \angle(c_{ij} + is_{ij})$ and the first conical surface are together equivalent to the 4-dimensional helix constraint $(c_{ij}, s_{ij}, z, \theta_j - \theta_i) \in \{(x_3\cos(\alpha), x_3\sin(\alpha), x_3, \alpha) : x_3 \in \mathbb{R}_+, \alpha \in [-\pi, \pi]\}$.

We implement the relaxation (6) for the helix constraints and (4) for the conical surface, with the additional bound $\cos(\theta_{v+1})\, x_1^{(v)} + \sin(\theta_{v+1})\, x_2^{(v)} \leq x_3$. Since in AC-OPF the voltages at every node have explicit bounds, $\mathbf{c}_{ij}$ and $\mathbf{s}_{ij}$ are also bounded and the disjunctive constraints can be transformed into big-M constraints as we saw before. We compare that approach with CDC formulations of the same regions, which can be constructed from reflected gray codes as detailed in [11].

For each of the test cases, and each of the three variants under different network conditions, we constructed MILP relaxations with $v = 6$ to 9 binary variables per constraint, and solved them with Gurobi 11 on a computer with an Intel Core i7-12700 CPU and 32 GB RAM.

The first experiment compares our formulation with CDC-based formulations in terms of solving time. We focus on small instances of PGLib and NESTA (with less than 30 nodes) for which the relaxations can be solved exactly, allowing up to 10 hours of solving time. If both relaxations are solved by Gurobi within that time, we compute the ratio between the solving times.

For most of the instances, the formulation we propose is faster, and the speedup tends to increase as the relaxations get more precise. However, in some of the problems (notably the networks with 24 nodes), the CDC-based formulation is solved over 10 times faster and seems to scale better as well.

FIGURE 2. Performance of the compact formulation in PGLib instances, with solving times for the CDC formulation (dashed lines) and compact formulation (solid lines) on the left, and speedup of the compact formulation on the right, for $v \in \{6, 7, 8, 9\}$

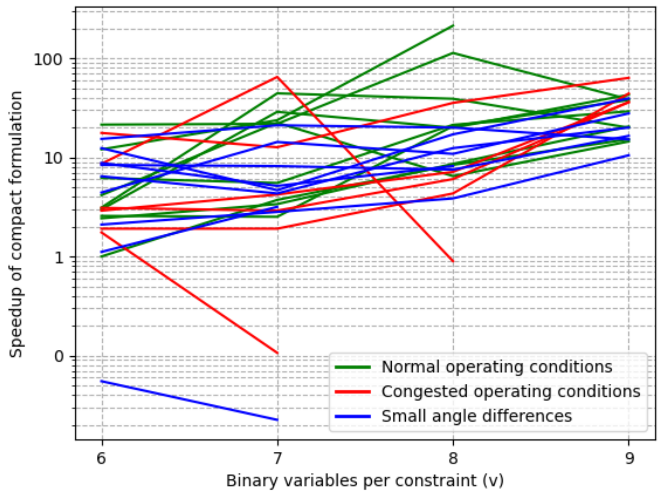


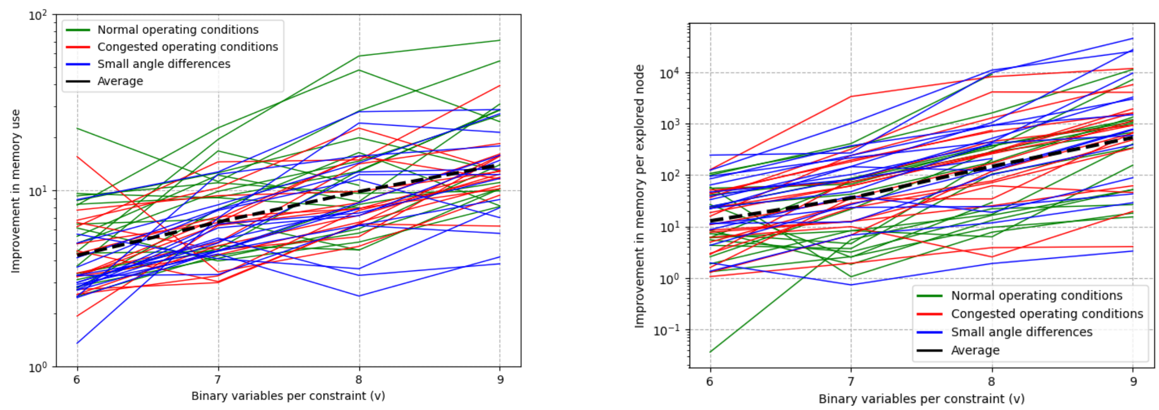
Table 2 lists all the solving times, and if the time limit is reached, we give the gap between the dual bounds obtained by the solver and the primal solution obtained by IPOPT [21] for comparison.

TABLE 2. Complete solving times in seconds for compact and CDC formulations, with binary variables per constraint in parenthesis

Case	CDC (6)	Compact (6)	CDC (7)	Compact (7)	CDC (8)	Compact (8)	CDC (9)	Compact (9)
3_lmbd	0.5	**0.2**	0.7	**0.2**	1.8	**0.2**	4.3	**0.2**
5_pjm	0.1	0.1	0.4	**0.1**	2.6	**0.3**	21.5	**1.4**
6_c	0.5	**0.1**	3.1	**0.1**	2.0	**0.3**	13.9	**1.0**
6_ww	0.5	**0.2**	12.0	**0.3**	78.4	**2.0**	52.9	**2.6**
9_wscc	0.9	**0.1**	1.1	**0.2**	5.1	**0.2**	7.9	**0.3**
14_ieee	4.5	**0.2**	17.7	**0.8**	114.8	**1.0**	325.3	**8.4**
24_ieee_rts	1.3	**0.5**	6.8	**2.7**	352.6	**17.0**	1254.6	**32.9**
30_as	2.1	**0.7**	75.5	**2.6**	179.9	**9.0**	1083.0	**25.2**
30_ieee	60.5	**5.0**	212.2	**8.9**	9477.4	**44.2**	(1.24%)	**263.3**
3_lmbd__api	0.3	**0.1**	0.6	**0.2**	1.2	**0.2**	7.6	**0.2**
5_pjm__api	0.2	**0.1**	0.2	**0.1**	0.9	**0.2**	13.7	**0.3**
6_c__api	0.2	**0.1**	2.0	**0.1**	5.2	**0.2**	16.0	**0.5**
9_wscc__api	0.2	**0.1**	0.3	**0.1**	1.0	**0.1**	2.2	**0.2**
14_ieee__api	0.6	**0.2**	1.3	**0.3**	120.3	**16.8**	1106.7	**30.6**
24_ieee_rts__api	448.5	**256.2**	**1592.4**	14959.4	**13366.8**	(1.44%)	**(1.56%)**	(1.95%)
30_as__api	36.3	**4.2**	5791.2	**89.0**	**27865.8**	31032.3	(39.77%)	**(39.25%)**
30_ieee__api	26.7	**1.5**	106.7	**8.4**	574.1	**16.1**	21515.2	**337.8**
3_lmbd__sad	0.2	**0.1**	0.3	**0.1**	0.8	**0.2**	2.2	**0.2**
5_pjm__sad	0.7	**0.1**	0.9	**0.2**	2.6	**0.2**	6.2	**0.3**
6_c__sad	0.6	**0.1**	2.6	**0.2**	3.6	**0.3**	15.3	**0.6**
6_ww__sad	7.5	**0.6**	11.0	**2.4**	62.5	**3.6**	132.6	**3.4**
9_wscc__sad	2.6	**0.2**	7.2	**0.3**	10.4	**0.5**	19.9	**1.3**
14_ieee__sad	23.6	**2.8**	50.9	**6.2**	233.8	**31.0**	817.4	**49.5**
24_ieee_rts__sad	**203.0**	3692.3	**643.1**	28733.5	**5614.0**	(3.53%)	**14940.4**	(3.55%)
30_as__sad	1677.6	**1509.2**	18788.2	**5921.4**	**(0.57%)**	(0.59%)	(0.59%)	**(0.48%)**
30_ieee__sad	98.8	**11.2**	406.2	**78.7**	7374.5	**899.8**	(0.50%)	**6563.9**

The reduction in the number of continuous variables does not always reduce the solving time, but it should reduce the memory needed to store the solutions of the LP relaxations at each node in a branch and bound algorithm. To test this, we design a second experiment in which we measure the memory used by Gurobi and the number of nodes it explores, this time using PGLib cases with up to 300 nodes, and with a time limit of 1 hour. To account for the root node, we use the quotient between the memory and the number of nodes explored plus one as the memory used per node.

FIGURE 3. Improvement in memory used as a function of v



(a) Ratio between the memory used by each formulation

(b) Ratio between the memory used by each formulation per explored node

Our compact formulation clearly dominates the CDC-based formulation in memory use, and it scales better as well, using around 4 times less memory and 13 times less memory per explored node at $v = 6$ variables per constraint but around 14 times less memory and 530 less per explored node at $v = 9$. For the CDC formulations of some of the larger cases, Gurobi runs out of memory before reaching the time limit, while our formulation never uses more than 4GB. A detailed table of these values can be found in the Appendix.

5. Conclusion We have shown that the general formulation (1) can be used to produce disjunctive or MILP envelopes of many common nonconvex constraints, with the size of the formulation scaling logarithmically on the number of polyhedra that enclose the feasible set. We also proved that we only need $O(\log(\varepsilon^{-1}))$ total variables and constraints to achieve an approximation error of ε, and we cannot attain the same error with fewer variables and constraints than $O(\log(\varepsilon^{-1}))$.

These formulations tend to perform and scale better overall than those based on combinatorial disjunctive constraints with the same branching pattern, but in some instances they perform significantly worse. In practice, the number of continuous variables is not as crucial as the number of integer variables, and a larger number of constraints or continuous variables can be overcome by using column generation or cut generation in the simplex method at each iteration of the branching algorithm.

However, the reduction in the number of total variables or constraints makes the proposed formulations much more efficient in terms of memory, and more robust against numerical problems than previously studied formulations. In problems where extensive branching is needed and the nonconvex constraints have to be approximated very precisely, these compact formulations perform better.

Finally, other implementations of piecewise linear relaxations compute a different formulation at each branch at run-time [14] instead of solving a single MILP formulation. Future research may determine whether the latter approach is more effective in practice when the formulation does not need to be solved exactly.

Appendix. Additional computational results

Table 3. Memory used in GB for each formulation under typical operating conditions

Case	Compact (6)	CDC (6)	Compact (7)	CDC (7)	Compact (8)	CDC (8)	Compact (9)	CDC (9)
3_lmbd	0.0155	0.0778	0.0176	0.174	0.0199	0.291	0.021	0.561
5_pjm	0.00468	0.0391	0.0258	0.236	0.0386	0.496	0.0429	1.24
14_ieee	0.0165	0.372	0.0787	0.885	0.0994	2.81	0.108	5.89
24_ieee_rts	0.0311	0.171	0.153	1.87	0.43	3.72	0.272	8.42
30_ieee	0.138	1.14	0.199	4.51	0.369	17.8	0.819	20.1
30_as	0.0475	0.214	0.146	2.84	0.201	11.7	0.243	17.3
39_epri	0.176	1.54	0.46	5.81	0.864	17.2	1.05	13.5
57_ieee	0.579	3.54	2.14	8.54	1.75	8.93	2.03	16.5
60_c	0.39	3.64	1.2	13.3	1.46	14	1.42	14.1
73_ieee_rts	0.4	2.58	1.56	10.7	2.68	12.3	2.09	21.3
89_pegase	2.1	6.56	2.19	10.2	1.79	23.3	1.65	47.3
118_ieee	2.46	10.9	2.54	11.5	2.53	21.1	2.54	28.2
162_ieee_dtc	2.61	7.29	2.08	16.3	1.71	28.1	1.83	15
179_goc	2.8	8	2.67	11	2.82	17.1	2.86	40.1
197_snem	1.12	10.7	1.44	13.1	2.94	23.5	2.64	(**)
200_activ	2.91	10.8	1.31	21.9	1.96	21	2.61	(*)
240_pserc	3.48	8.74	2.89	12.3	3.7	21.2	2.72	27.8
300_ieee	3.09	10.4	3.74	18.2	3.09	14.2	3.47	(*)

*: Out of memory **: Encountered numerical problems

Table 4. Memory used in GB for each formulation under congested operating conditions

Case	Compact (6)	CDC (6)	Compact (7)	CDC (7)	Compact (8)	CDC (8)	Compact (9)	CDC (9)
3_lmbd__api	0.0141	0.0882	0.0159	0.231	0.0182	0.271	0.0188	0.742
5_pjm__api	0.00559	0.0344	0.00601	0.06	0.0197	0.384	0.0352	1.15
14_ieee__api	0.011	0.0937	0.0178	0.224	0.218	2.23	0.337	9.41
24_ieee_rts__api	1.44	2.8	1.14	7.24	1.23	9.73	0.93	9.52
30_as__api	0.139	1.08	0.697	6.52	1.18	11.7	0.928	11.9
30_ieee__api	0.129	0.876	0.176	1.82	0.342	7.74	1.66	21.7
39_epri__api	0.27	1.35	0.559	3.67	0.94	14.8	1.28	14.9
57_ieee__api	1.99	6.45	1.5	6.47	1.3	8.61	1.39	14.8
60_c__api	0.416	6.48	2.12	7.32	1.69	8.14	1.42	13.3
73_ieee_rts__api	1.72	7.81	2.16	6.59	1.79	11.6	1.75	27.6
89_pegase__api	1.73	5.87	1.96	9.94	1.64	23.3	1.42	26.3
118_ieee__api	2.4	6.16	2.13	10.4	2.59	18.7	2.37	29.3
162_ieee_dtc__api	2.46	7.97	2.14	13.7	1.92	16.3	2.1	33.8
179_goc__api	3.63	9.34	3.26	10.7	2.67	20.6	2.57	30.6
197_snem__api	2.22	14.6	3.01	13.9	4.3	23.5	2.34	36.8
200_activ__api	0.847	3.71	0.955	6.78	3.08	24.3	2.81	36.5
240_pserc__api	2.95	8.09	2.25	6.77	2.64	17	2.4	15.1
300_ieee__api	3.15	10.4	2.59	17.9	2.86	44.7	3.76	(*)

*: Out of memory **: Encountered numerical problems

Table 5. Memory use in GB for each formulation under small angle differences

Case	Compact (6)	CDC (6)	Compact (7)	CDC (7)	Compact (8)	CDC (8)	Compact (9)	CDC (9)
3_lmbd__sad	0.00939	0.0235	0.0101	0.125	0.017	0.267	0.0187	0.335
5_pjm__sad	0.024	0.121	0.0288	0.239	0.0326	0.468	0.0398	1.08
14_ieee__sad	0.079	0.701	0.123	1.57	0.142	3.98	0.373	10.8
24_ieee_rts__sad	2.16	2.94	0.874	6.1	0.625	15.1	0.865	18.5
30_as__sad	0.611	3.46	1.21	4.95	1.32	4.75	1.26	10.1
39_epri__sad	0.9	3.74	1.24	9.73	1.33	16.9	1.19	15.8
57_ieee__sad	1.79	4.42	1.42	6.43	1.37	(**)	1.28	11.4
60_c__sad	3.13	8.84	1.22	7.88	1.45	10.8	1.15	13.4
73_ieee_rts__sad	2.35	7.72	1.95	6.5	1.66	10.5	1.59	9.06
89_pegase__sad	2.07	5.67	1.43	8.79	2.05	14.8	1.77	26.3
118_ieee__sad	2.07	5.62	1.85	8.41	1.92	16.3	1.71	27.1
162_ieee_dtc__sad	2.07	7.59	1.79	13.8	1.71	19.7	1.78	12.5
179_goc__sad	1.94	5.49	2.07	10.4	1.8	22.2	2.08	25.7
197_snem__sad	2.28	7.54	2.42	11	2.27	17.9	2.71	(*)
200_activ__sad	2.04	6.14	2.07	10.8	2.48	19.5	2.5	(*)
240_pserc__sad	2.7	7.93	2.34	10.2	2.37	5.97	2.27	9.54
300_ieee__sad	3.06	9.2	2.79	15	2.27	7.48	2.65	10.2

*: Out of memory **: Encountered numerical problems

Table 6. Nodes explored for each formulation under typical operating conditions

Case	Compact (6)	CDC (6)	Compact (7)	CDC (7)	Compact (8)	CDC (8)	Compact (9)	CDC (9)
3_lmbd	271	1034	435	1665	932	1416	965	1670
5_pjm	0	0	117	586	2419	2348	8399	4637
14_ieee	0	623	929	1884	3307	4218	3521	7183
24_ieee_rts	0	0	1197	4592	11559	6021	10431	6176
30_as	12	0	401	7428	1632	12220	4077	15884
30_ieee	2500	11028	9835	26748	24478	187279	352235	55324
39_epri	6303	9144	63001	96248	245820	284848	637188	20138
57_ieee	28761	28820	1065719	57524	636934	11514	549971	6584
60_c	7456	16145	127552	379701	629548	57660	387024	11299
73_ieee_rts	1282	3199	79770	25166	583801	9098	342731	3452
89_pegase	141393	18348	45182	4955	37594	2730	29454	626
118_ieee	599062	47540	283362	19216	185654	4150	93888	957
162_ieee_dtc	50374	3387	28809	2780	13031	633	7995	8
179_goc	315997	37056	138524	6590	93693	1516	55354	645
197_snem	60647	80161	17224	33779	277478	11090	129817	(**)
200_activ	568085	79365	5790	38479	31228	11370	57806	(*)
240_pserc	84881	1974	51013	611	46982	165	16702	14
300_ieee	96587	3263	59645	703	31889	15	19211	(*)

*: Out of memory **: Encountered numerical problems

Table 7. Nodes explored for each formulation under congested operating conditions

Case	Compact (6)	CDC (6)	Compact (7)	CDC (7)	Compact (8)	CDC (8)	Compact (9)	CDC (9)
3_lmbd__api	177	1044	314	2372	537	2065	471	4564
5_pjm__api	0	0	0	0	80	612	724	1194
14_ieee__api	0	0	0	0	37889	6220	40269	24545
24_ieee_rts__api	283818	110461	2107381	475074	1315160	134966	952095	19972
30_as__api	1135	6839	271164	372169	2875802	238922	1603455	31455
30_ieee__api	1163	1020	3937	3240	23736	21168	221432	46264
39_epri__api	19515	8996	92483	17537	430291	195756	1425520	24510
57_ieee__api	1446532	206477	530260	37073	308533	7531	242759	1597
60_c__api	17497	170667	1770136	169870	808281	23029	550197	4504
73_ieee_rts__api	316664	90706	771013	9699	480996	2390	281786	772
89_pegase__api	87995	16013	50308	4127	24978	1358	14659	289
118_ieee__api	353149	20947	189203	4664	117657	1207	77450	591
162_ieee_dtc__api	70912	6273	33899	676	20486	41	10889	42
179_goc__api	557680	27875	291055	13137	169298	4646	91642	557
197_snem__api	165110	379159	573708	32501	486641	5841	128843	2156
200_activ__api	487	712	1344	415	83686	9200	56470	2194
240_pserc__api	76123	1639	24863	21	28012	21	13272	6
300_ieee__api	64172	4593	24809	936	29412	615	30788	(*)

*: Out of memory **: Encountered numerical problems

Table 8. Nodes explored for each formulation under small angle differences

Case	Compact (6)	CDC (6)	Compact (7)	CDC (7)	Compact (8)	CDC (8)	Compact (9)	CDC (9)
3_lmbd__sad	65	83	67	1148	143	1183	175	946
5_pjm__sad	208	779	938	1140	1494	1817	2419	2271
14_ieee__sad	7960	6332	17183	9318	22843	26096	56768	37874
24_ieee_rts__sad	1367701	213592	1370514	245846	754263	963935	606159	145422
30_as__sad	1244854	312181	1661265	142992	1109544	9483	657136	8806
39_epri__sad	168634	61066	698300	445628	1200422	149080	956786	32411
57_ieee__sad	1001883	52512	574300	31810	393875	(**)	277513	749
60_c__sad	1902062	1247096	756624	182663	657246	36932	385986	5717
73_ieee_rts__sad	816045	80661	502592	11373	292877	1756	208391	42
89_pegase__sad	59322	7361	36059	1513	13793	815	14027	271
118_ieee__sad	274261	8800	131754	1611	67607	620	36766	388
162_ieee_dtc__sad	42184	1622	19526	692	12408	324	6907	4
179_goc__sad	135194	5955	86213	1793	46035	609	33850	137
197_snem__sad	342232	28849	217608	9620	128609	2579	95884	(*)
200_activ__sad	94612	11582	75779	4702	64247	2395	37203	(*)
240_pserc__sad	43246	1015	20419	86	13175	2	6105	0
300_ieee__sad	63641	779	34608	686	12002	3	11934	0

*: Out of memory **: Encountered numerical problems

Acknowledgments

We would like to express our sincere gratitude to Carleton Coffrin for providing the latest version of NESTA.

References

[1] Babaeinejadsarookolaee S, Birchfield AB, Christie RD, Coffrin C, DeMarco CL, Diao R, Ferris MC, Fliscounakis S, Greene S, Huang R, Josz C, Korab R, Lesieutre BC, Maeght J, Molzahn DK, Overbye TJ, Panciatici P, Park B, Snodgrass J, Zimmerman RD (2019) The power grid library for benchmarking ac optimal power flow algorithms. *ArXiv* abs/1908.02788, URL `https://api.semanticscholar.org/CorpusID:199501662`.

[2] Bao X, Sahinidis NV, Tawarmalani M (2009) Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs. *Optimization Methods and Software* 24(4-5):485–504, URL `http://dx.doi.org/10.1080/10556780902883184`.

[3] Beach B, Hildebrand R, Huchette J (2022) Compact mixed-integer programming formulations in quadratic optimization. *Journal of Global Optimization* 4(84):869–912, URL `http://dx.doi.org/10.1007/s10898-022-01184-6`.

[4] Belotti P, Liberti L, Lodi A, Nannicini G, Tramontani A (2011) *Disjunctive Inequalities: Applications And Extensions* (John Wiley Sons, Ltd), ISBN 9780470400531, URL `http://dx.doi.org/https://doi.org/10.1002/9780470400531.eorms0537`.

[5] Ben-Tal A, Nemirovski A (2001) On polyhedral approximations of the second-order cone. *Mathematics of Operations Research* 26(2):195–205.

[6] Bienstock D, Muñoz G (2014) On linear relaxations of opf problems. *arXiv: Optimization and Control* URL `https://api.semanticscholar.org/CorpusID:117479734`.

[7] Coffrin C, Gordon D, Scott P (2014) Nesta, the nicta energy system test case archive. *ArXiv* abs/1411.0359, URL `https://api.semanticscholar.org/CorpusID:8719091`.

[8] Glover F (1975) Improved linear integer programming formulations of nonlinear integer problems. *Management Science* 22(4):455–460.

[9] Gupte A, Ahmed S, Cheon MS, Dey S (2013) Solving mixed integer bilinear problems using milp formulations. *SIAM Journal on Optimization* 23(2):721–744, ISSN 1052-6234, URL `http://dx.doi.org/10.1137/110836183`.

[10] Huchette J, Vielma JP (2019) A combinatorial approach for small and strong formulations of disjunctive constraints. *Mathematics of Operations Research* 44(3):793–820, URL `https://doi.org/10.1287/moor.2018.0946`.

[11] Huchette J, Vielma JP (2019) A geometric way to build strong mixed-integer programming formulations. *Operations Research Letters* 47(6):601–606.

[12] Huchette J, Vielma JP (2022) Nonconvex piecewise linear functions: Advanced formulations and simple modeling tools. *Operations Research* 71(5):1835–1856, URL `http://dx.doi.org/10.1287/opre.2019.1973`.

[13] Kocuk B, Dey SS, Sun XA (2016) Strong socp relaxations for the optimal power flow problem. *Operations Research* 64(6):1177–1196, URL `http://dx.doi.org/https://doi.org/10.1287/opre.2016.1489`.

[14] 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. *Mathematical Programming Computation* 10(4):557–596.

[15] Lee J, Leyffer S (2011) *Mixed integer nonlinear programming*, volume 154 (Springer Science & Business Media).

[16] Lyu B, Hicks I, Huchette J (2024) Modeling combinatorial disjunctive constraints via junction trees. *Mathematical Programming* 204:385–413, URL `https://doi.org/10.1007/s10107-023-01955-3`.

[17] Lyu B, Hicks IV, Huchette J (2025) Building formulations for piecewise linear relaxations of nonlinear functions. *Operations Research, to appear* URL `http://dx.doi.org/10.1287/opre.2023.0187`.

[18] Rikun AD (1997) A convex envelope formula for multilinear functions. *Journal of Global Optimization* 10(4):425–437, ISSN 0925-5001, URL `http://dx.doi.org/10.1023/A:1008217604285`.

[19] Tawarmalani M, Richard JP, Xiong C (2013) Explicit convex and concave envelopes through polyhedral subdivisions. *Mathematical Programming* 138:531–577, URL `http://dx.doi.org/10.1007/s10107-012-0581-4`.

[20] Vielma JP, Ahmed S, Nemhauser G (2009) Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research* 2(58):303–315.

[21] Wächter A, Biegler L (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* 106:25–57, URL `http://dx.doi.org/10.1007/s10107-004-0559-y`.