RICE UNIVERSITY

Compact MIP Formulations of Nonlinear Constraints for the Optimal Power Flow Problem

by

Gonzalo Gomez Abejon

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master in Arts

Approved, Thesis Committee:

Illya V. Hicks, Chair
Trustee Professor of Computational and Applied Mathematics and Operations Research

Sebastian Perez-Salazar
Professor of Computational and Applied Mathematics and Operations Research

Cesar Uribe
Professor of Electrical and Computer Engineering

Houston, Texas

April, 2025

ABSTRACT

Compact MIP Formulations of Nonlinear Constraints for the Optimal Power Flow Problem

by

Gonzalo Gomez Abejon

We propose compact formulations for approximating certain families of nonlinear, nonconvex constraints, achieving an exponential reduction in the number of total variables used compared to previous formulations. Using a geometric construction, we show that relaxations of certain polynomial and trigonometric constraints given by the union of n polytopes can be modeled with $O(\log(n))$ total variables and constraints. We prove that this bound is asymptotically optimal, that is, that we can only reduce the formulation size further by a constant factor. We then show how these formulations can be applied to the AC optimal power flow problem, and construct an MIP relaxation for said problem based on its polar formulation. Finally, we present some computational experiments where the performance of MIP relaxations is compared with standard convex relaxations for AC-OPF.

Contents

Illustrations

Tables

Chapter 1

Introduction

1.1 Literature Review

1.1.1 Polyhedral Relaxations of Nonlinear Constraints

A well-known technique for solving optimization problems over nonlinear domains is to replace the feasible region with a piecewise approximation [33]. If the problem has constraints of the form $y = f_i(x)$ for $x \in \mathbb{R}^n$, we can work with the graph of each f_i separately, and partition its domain into a union of convex polyhedra. We then replace the graph over each polyhedron by a linear approximation or by an envelope of the graph, and arrive at an approximation or a relaxation that we can solve exactly.

Optimizing over this new domain amounts to solving a disjunctive program, and this can be solved by enumeration (iterating over all domain "pieces") but a better approach is formulating the problem as a mixed integer program (MIP) [7], for which more sophisticated branch-and-bound algorithms can be used. Although a MIP is not necessarily linear, if the disjunctive program is piecewise linear, a mixed integer linear program (MILP) formulation exists under mild assumptions, such as a bounded domain [35]. In that paper, the authors outline some constructions for that purpose, some of which minimize the number of binary variables used, which can be reduced to the theoretical minimum of $O(\log_2(n))$ for a domain partitioned in n polytopes, using big-M formulations.

The numerical instability that big-M formulations may produce and the computation of M can be avoided by employing combinatorial disjunctive constraints (CDCs) [21], which only need to account for the topology of the feasible region. The problem of finding compact formulations for a CDC is more nuanced and has been studied extensively in [21, 31, 36]. For most mesh topologies used to approximate nonlinear functions (n-dimensional grids, 2-d triangulations) we can achieve the same lower bound of $O(\log_2(n))$ [23]. However, the reduction in the number of binary variables comes at the expense of a large number of continuous variables (at least the total number of vertices of the polytopes) and constraints.

On the other hand, the construction of the piecewise linear functions and envelopes has been studied for many types of constraints, such as quadratic [5, 6], bilinear [16, 19] or multilinear [32]. A more general approach is to construct piecewise McCormick relaxations, which use bounds on the variables to construct an envelope for a function over a hyperrectagle $\prod_{i=1}^{n}[l_i, u_i]$. We could also combine both methods to construct a union of convex sets that contain the feasible region of each restriction, as in [34].

For general relaxations such as McCormick applied to ill-behaved functions, any resulting linear program should have at least one variable or constraint for each of the vertices or facets of the polyhedra it contains. However, [6] and [19] show that for certain concrete functions, there are formulations with only $O(\log(n))$ variables and constraints, where n is the number of partitions of the domain.

1.1.2 Optimal Power Flow (OPF)

The problem of transmitting electrical power from generators to consumers over a network, minimizing the generation cost, is of critical importance in any electrical grid system. A brief overview of the history of this problem is provided in [11].

Unlike other transport problems in graphs, which can be solved in polynomial time through linear programming or combinatorial algorithms [15], the OPF problem is NP-hard for general alternating current (AC) circuits [29, 30].

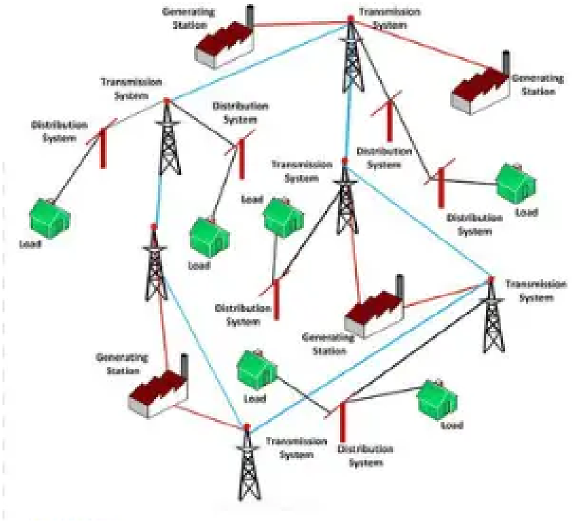


Figure 1.1 : Electrical power grid

This is because in the general AC problem, denoted AC-OPF or ACOPF in the literature, magnitudes like electrical power, voltage, current or impedance are complex numbers, and the laws of physics (Ohm's law and Kirchhoff's law) define nonconvex constraints on their real and imaginary parts. Most practical instances are computationally expensive, with tens of thousands of nonconvex quadratic or trigonometric constraints to model [2].

In practice, this kind of problems are usually solved by local search methods, which are computationally efficient (linear or polynomial in the number of variables and constraints) but offer no guarantee of optimality and may get stuck at a local minimum. One of the most popular solvers of this kind in the literature is IPOPT [37], which is, among others, used by [2, 12], and available for Matpower [39].

The cost of a feasible solution produced by a solver of this type is an upper bound of the true solution. In order to produce a lower bound, we can construct a relaxation of ACOPF, by minimizing across a relaxed feasible set that contains the exact feasible region of ACOPF as a subset. If the relaxation is convex, any local minimum is global, so the solution is exact and gives a lower bound for ACOPF. Although optimality gaps are not too important in practice and, for most applications, we only want a feasible and almost optimal solution that can be computed easily, the solution of a relaxation can also be used as a starting point for a local search algorithm.

Most of the literature focuses on finding additional valid convex constraints for one of three well-known convex relaxations: SOC [10, 28], QC [13] and SDP [3, 4, 26], and/or discuss new algorithms to solve them in their standard form. These are based on second order cone constraints, quadratic constraints and semidefinite programming, respectively.

More recently, Bienstock and Muñoz [9] have proposed a tighter nonconvex formulation that approximates bilinear constraints using a mixed integer linear relaxation. Kocuk et al. also use selective spatial branching in [27] to improve the dual bounds of convex relaxations. This approach can produce optimality gaps arbitrarily close to zero, at the cost of increasing the formulation size, and further decreases in computation costs could make MIP formulations more popular for ACOPF.

1.2 Contributions

We construct relaxed approximations for certain polynomial and trigonometric constraints, both with disjunctive programs and MIL constraints, and show that they are minimal in terms of the number of linear constraints and added variables.

1. Our relaxations for constraints of the form $x_3 \leq \sqrt{x_1^2 + x_2^2}$ and $x_3 = \sqrt{x_1^2 + x_2^2}$

have the smallest error $\varepsilon = \frac{x_3}{\sqrt{x_1^2+x_2^2}} - 1$ among those with the same number of integer variables.

2. They also use $O(log(\varepsilon^{-1}))$ total variables and constraints, which is the best possible asymptotic size, compared with the $O(\varepsilon^{-1})$ variables needed by a CDC formulation.

3. Our relaxations for constraints of the form $x_1 = \cos(x_2)$ also achieve the best possible asymptotic size and beat previous formulations, while using the same number of integer variables.

4. We propose a novel relaxed approximation of the OPF problem based on its polar formulation, using the constructions above.

1.3 Outline of Thesis

Chapter 2 introduces some definitions and concepts already discussed in the literature. Its first section is devoted to disjunctive programs and mixed integer formulations for them, while the second one defines the optimal power flow problem and some of its formulations and relaxations. Chapter 3 introduces our contributions, some related theoretical results, and a general approach for constructing compact disjunctive formulations for certain manifolds. These are applied to the OPF problem in Chapter 4, where we present our relaxation of ACOPF, implement it in Gurobi and discuss its computational results. Finally, in Chapter 5 we draw conclusions and outline future research directions.

Chapter 2

Preliminaries

2.1 Disjunctive Constraints

A disjunctive program can be represented as $\min\left\{f(x) : \bigvee_{i=1}^{d} g_i(x) \leq 0\right\}$ where $x \in \mathbb{R}^n$ and $g_i : \mathbb{R}^n \to \mathbb{R}^m$, meaning x is feasible if $g_i(x) \leq 0$ for some i. If the functions g_i are linear, we have $x \in \cup_{i=1}^{d} P^i$ where each P^i is a polyhedron. For practicality, we will focus on the case where P^i is bounded, that is, a polytope, although the same theory can be developed under some assumptions for unbounded polyhedra [25].

In that case, an obvious way to formulate the problem as a MIP is through a big-M formulation, namely

$$g_1(x) \leq 0 \text{ or } g_2(x) \leq 0 \quad \iff \quad \begin{cases} g_1(x) \leq Mz \\ g_2(x) \leq M(1-z) \\ z \in \{0,1\} \end{cases} \tag{2.1}$$

which is true for a large enough M if $g_i : x \to A_i x - b_i$ are linear functions that delimit polytopes. For general functions, it may not be true since $g_1(x)$ could be unbounded within the set $\{x \in \mathbb{R}^n : g_2(x) \leq 0\}$. Still, we are left with the problem of finding a value of M large enough for the equivalence to hold, but not much more, as bigger values tend to slow down MIP solvers and may cause numerical problems.

By the Minkowsky-Weyl Theorem [14], we can characterize polytopes as convex hulls of points, instead of bounded sets delimited by linear inequalities. Suppose

$$P^i = conv(V^i) = \left\{\sum_{v \in V^i} \lambda_v v : \sum_{v \in V^i} \lambda_v = 1, \lambda \geq 0\right\} \tag{2.2}$$

and the sets of points V^i are not disjoint in general, that is, $V^i = \{V_j : j \in S^i\}$ for $V := \bigcup_i V^i = \{v_1, \dots v_m\}$ and $\mathcal{S} := \{S^i\}$. We can synthesize the relations of shared vertices in the following graph:

Definition 2.1 (Conflict Hypergraph) The hypergraph with vertices $[\![m]\!] = \{1, \dots m\}$ and edges $\{X \subseteq [\![m]\!] : X \nsubseteq S_i \ \ \forall S_i \in \mathcal{S}\}$ is called the *conflict hypergraph* of $\mathcal{S}$.

If the vertices and $\mathcal{S}$ are known, it is possible to avoid the M constant with the following formulation [35]:

$$\begin{cases} x = \sum_{j=1}^{|V|} \lambda_j v_j \\ \sum_{j=1}^{|V|} \lambda_j = 1 \\ \sum_{j \in S} \lambda_j \geq z_S \qquad \forall S \in \mathcal{S} \\ \sum_{S \in \mathcal{S}} z_S = 1 \\ \lambda_j \geq 0 \qquad z_S \in \{0, 1\} \end{cases} \tag{2.3}$$

The region defined by the last 4 constraints in λ-space is a Combinatorial Disjunctive Constraint (CDC), denoted $\lambda \in CDC(\mathcal{S})$. We note that a formulation of the CDC immediately gives a formulation of the original region (the converse is not true), but the CDC does not depend on the geometry of the feasible region, only on the set $\mathcal{S}$.

A natural question is whether we can reduce the number of binary variables in the formulation. Since a set of the form $\{x \in \mathbb{R}^n : \exists z \in \{0, 1\}^m : Ax + Bz \leq b\}$ is the union of 2^m polytopes $P_z = \{x \in \mathbb{R}^n : Ax \leq b - Bz\}$ defined for each $z \in \{0, 1\}^m$, the number of binary variables can't be more than $\lceil \log_2(n) \rceil$ in general.

This bound can always be achieved for a formulation of the original problem, but not necessarily for the CDC. For instance, if we choose n different vectors $\{h^1, h^2, \dots h^n\} \in \{0, 1\}^{\lceil \log_2(n) \rceil}$, the following MIP given in [35] is valid:

$$\begin{cases} x = \sum_{i=1}^{n} \sum_{j \in S^i} \lambda_{i,j} v_j & \\ \sum_{i=1}^{n} \sum_{j \in S^i} \lambda_{i,j} = 1 & \\ \sum_{i:h_l^i=0} \sum_{j \in S^i} \lambda_{i,j} \leq y_l & \forall l \in \{1, \ldots, \lceil \log_2(n) \rceil\} \\ \sum_{i:h_l^i=1} \sum_{j \in S^i} \lambda_{i,j} \leq (1 - y_l) & \forall l \in \{1, \ldots, \lceil \log_2(n) \rceil\} \\ \lambda_{i,j} \geq 0 \;\; \forall S^i \in \mathcal{S}, j \in S^i, & y_l \in \{0,1\} \;\; \forall l \in \{1, \ldots, \lceil \log_2(n) \rceil\} \end{cases} \tag{2.4}$$

However, the reduction in the number of binary variables comes at the expense of a larger number of continuous variables ($\sum_i |S^i|$) and constraints. In [36] it is proposed how to get a MIP formulation out of a pairwise independent branching (IB) decomposition, which amounts to finding polytopes L_k, R_k such that

$$P = \bigcup_{i=1}^{n} P^i = \bigcap_{k=1}^{m} (L_k \cup R_k) \tag{2.5}$$

If V_k^l and V_k^r index the vertices contained in L_k and R_k respectively, so that $V_k^l \cup V_k^r = \{1, \ldots, |V|\}$ for each k, the following formulation for $\lambda \in CDC(\mathcal{S})$ is valid:

$$\begin{cases} \sum_{j=1}^{|V|} \lambda_j = 1 \\ \sum_{j \in V_k^r} \lambda_j \geq z_k \quad \forall k \in [\![m]\!] \\ \sum_{j \in V_k^l} \lambda_j \geq 1 - z_k \quad \forall k \in [\![m]\!] \\ \lambda_j \geq 0 \;\; \forall j \in [\![|V|]\!], \qquad z_k \in \{0,1\} \;\; \forall k \in [\![m]\!] \end{cases} \tag{2.6}$$

In [21] the authors show that the minimal edges of the conflict hypergraph of $\mathcal{S}$ form a graph, the problem of finding a pairwise-IB decomposition of $CDC(\mathcal{S})$ with m disjunctions is equivalent to finding a biclique cover of size m in that conflict graph. Also, the formulations that result from this decomposition are ideal, which means that any extreme point of the linear relaxation is integral in the binary variables, and therefore feasible for the CDC.

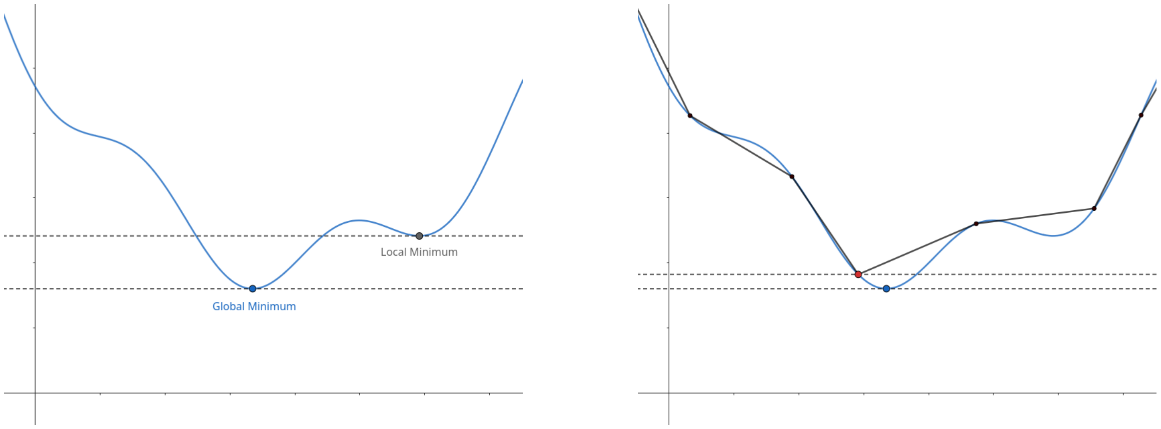
In general, a pairwise-IB decomposition of the problem may not exist, and even if it does, finding a minimal biclique cover of a graph is NP-hard. In practice, however, many special graphs we might encounter in different applications admit this decomposition, or can be made to admit it through an adequate choice of the polytopes P^i. For instance, path graphs and cycle graphs admit a decomposition of size $\lceil \log_2(n) \rceil$, and d-dimensional grids of size n (made of n^d polytopes) admit a formulation of size $d\lceil \log_2(n) \rceil \approx \lceil \log_2(n^d) \rceil$ [23]. The 2-dimensional triangular grid, which is often used for piecewise linear approximations of 3-d surfaces, can be obtained by splitting each square in a 2-d grid in half, and admits a decomposition with $2\lceil \log_2(n) \rceil + 1$ integer variables.

2.2 MIP Relaxations of Nonlinear Constraints

While there are many applications of disjunctive constraints, our focus from now on will be on the approximation and approximate relaxation of nonlinear constraints. For an abstract problem $\min\{f(x) : x \in X \subset \mathbb{R}^n\}$, a relaxation of the problem is a similar problem $\min\{f(x) : x \in X' \subset \mathbb{R}^n\}$ where $X' \supseteq X$. We can also say that the constraint $x \in X'$ is a relaxation of X, or that X' is a relaxation or envelope of X. If we have a sequence $\{X_n\}_{n\in\mathbb{N}}$ such that X_n approaches X in some sense (not necessarily as a limit of sets) as $n \to \infty$, we say $x \in X_n$ is an approximation of $x \in X$. If in addition $X_n \supseteq X$ for every n, we have a relaxed approximation of the problem, constraint, or feasible region.

Given a generic problem of the form $v = min\{f(x) : x \in \mathbb{R}^n, \quad g_i(x) \leq 0 \quad \forall i \in [\![m]\!]\}$, where the functions g_i may not be convex, we would like to obtain something close to a global solution and some guarantees of optimality. The easiest approach is to use a local search algorithm, incorporating the constraints as a penalty in the objective function, to obtain a point $\hat{x}$ that is almost feasible ($g_i(x) < \epsilon$ for some tolerance ϵ) and a local minimum of f. An algorithm of this type scales well with respect to the dimension n and the number of constraints m, but it does not provide guarantees of optimality and depends on the choice of starting point.

On the other hand, if the problem domain is bounded and $\{x \in \mathbb{R}^n : g_i(x) \leq 0 \quad \forall i \in \llbracket m \rrbracket\} \subseteq \prod_{i=1}^{n}[l_i, u_i]$, we can partition the domain $[l_i, u_i]$ of each variable into smaller intervals at some k_i intermediate points $p_0^i = l_i, p_1^i, \ldots, p_{k_i}^i = u_i$. If we then approximate the functions g_i and f at each of the parallelepipeds $[p_{j_1}^1, p_{j_1+1}^1] \times [p_{j_2}^2, p_{j_2+1}^2] \times \cdots \times [p_{j_n}^n, p_{j_n+1}^n]$ with a linear function, we have a piecewise linear approximation of the whole problem that can be formulated as a union of polytopes. We can solve it exactly with an MIP, and expect to obtain an almost-feasible solution with an objective value close to that of the original problem.



(a) An iterative method may get stuck at a local minimum

(b) Approximating the problem gives an approximation of the solution

Figure 2.1 : Nonconvex function and approaches to find its global minimum

Proposition 1 Let $v = \min\{f(x) : x \in \mathbb{R}^n,\ g_i(x) \leq 0\ \forall i \in I\}$*. Then if* $G_i \supseteq \{x \in \mathbb{R}^n : g_i(x) \leq 0\}$*, we have* $\min\{f(x) : x \in \bigcap_{i \in I} G_i\} \leq v$

Proof 2.1 Obviously if the first minimum is attained at $v = f(x^*)$, then $g_i(x^*) \leq 0 \Rightarrow x^* \in G_i\ \ \forall i \in I$ and $\min\{f(x) : x \in \bigcap_{i \in I} G_i\} \leq f(x^*) = v$

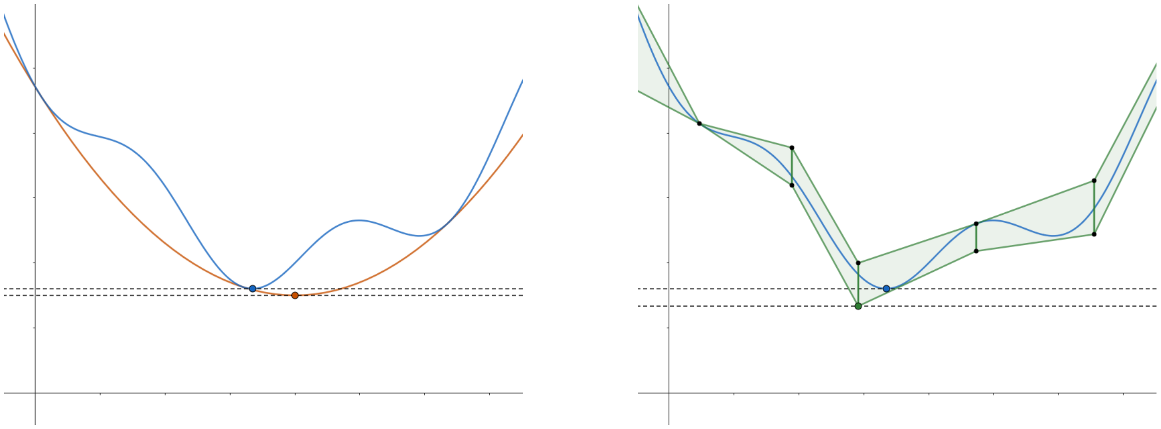
This means that if we find envelopes G_i such that the second problem can be solved exactly, we obtain a dual bound for the first. We may assume f is linear, since $\min\{f(x) : x \in \mathbb{R}^n\} = \min\{x_{n+1} \in \mathbb{R}^n : f(x) - x_{n+1} \leq 0\}$ and any nonlinear objective value can be transformed into a nonlinear constraint. There are two common approaches: choosing G_i to be convex sets (for instance, $G_i = \{g_i'(x) \leq 0\}$ for convex functions $g_i'(\cdot) \leq g_i(\cdot)$) ensures local search or interior point methods find the global minimum. On the other hand, if G_i are polyhedra or unions of polyhedra, we can formulate the relaxation as a disjunctive program and solve it to optimality by enumeration or branch-and-bound. Similarly, piecewise linear functions $g_i'(\cdot) \leq g_i(\cdot)$ can be used to generate the sets $G_i = \{g_i'(x) \leq 0\}$, but by choosing the linear "pieces" small enough, we can make the relaxation arbitrarily tight.

Figure 2.2 shows a convex envelope of the epigraph of a function, and a polyhedral envelope of its graph, although only the lower bound is needed for minimization. Note that if G_i is convex, it can only be as small as the convex hull of $\{g_i(x) \leq 0\}$, but a relaxation by polyhedra can be arbitrarily close to the original feasible set.

Remark 2.1

Although $\min\{a^T x : x \in G \subseteq \mathbb{R}^n\} = \min\{a^T x : x \in conv(G) \subseteq \mathbb{R}^n\}$*, in general* $\min\{a^T x : x \in G_i\ \ \forall i \in I\}$ *is not equal to* $\{a^T x : x \in conv(G_i)\ \ \forall i \in I\}$*.*

An example are the sets $G_1 := \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 = 2\}$, $G_2 := \{(x, y, z) \in \mathbb{R}^3 : y^2 + z^2 = 2\}$ and $G_3 := \{(x, y, z) \in \mathbb{R}^3 : z^2 + x^2 = 2\}$. Since $G_1 \cap G_2 \cap G_3$ contains the



(a) A convex relaxation produces a lower bound

(b) A piecewise linear relaxation also gives a lower bound

Figure 2.2 : Nonconvex function and approaches to find its global minimum

points $\{\pm 1, \pm 1, \pm 1\}$, but the convex hulls are cylinders $conv(G_1) = \{x^2 + y^2 \leq 2\}$, we have $\min\{x : (x, y, z) \in conv(G_1) \cap conv(G_2) \cap conv(G_3)\} = -\sqrt{2} < -1$.

Hence, even if we have ideal MIP formulations for relaxations of each constraint, we may lose precision if we replace any of them with the LP relaxation. In other words, an ideal formulation is useful for the complete problem, not for a constraint.

2.3 Optimal Power Flow

A simplified formulation of ACOPF, such as the one given in [28], assumes that we have a graph $\mathcal{G} = (\mathcal{B}, \mathcal{L})$ with n *buses* $\mathcal{B}$ and m *branches* $\mathcal{L} \subseteq \{\{i, j\} : \mathcal{B} \ni i \neq j \in \mathcal{B}\}$. Following the nomenclature in [2], we denote the voltage at node i by $V_i = e_i + if_i$, the electrical power by $S = p + iq$, the impedance of a line by $Z_{ij} = r_{ij} + ix_{ij}$ and its admittance by $Y_{ij} = \frac{1}{Z_{ij}} = G_{ij} + iB_{ij}$, and the shunt (load) admittance of each bus is Y_i^s. We can control the voltages and the power $S_i^g = p_i^g + iq_i^g$ generated at each node, which has an associated cost function $C_i(p_i^g)$ we want to minimize. We are constrained by Ohm's law, which states that the power flowing from i to j is $S_{ij} = Y_{ij}V_iV_j^*$ and Kirchhoff's law or flow conservation, $S_i^g - S_i^d = Y_i^s|V_i|^2 \sum_{ij\in\mathcal{L}} S_{ij}$ where S_i^d is the power demand at node i. If we replace the complex voltage and power variables with each real and imaginary components, the full problem is then

$$\min \sum_{i\in\mathcal{B}} C_i(\mathbf{p}_i^g)$$

subject to

$$\mathbf{p}_i^g - p_i^d = G_{ii}(\mathbf{e}_i^2 + \mathbf{f}_i^2) + \sum_{j\in\delta(i)} [G_{ij}(\mathbf{e}_i\mathbf{e}_j + \mathbf{f}_i\mathbf{f}_j) - B_{ij}(\mathbf{e}_i\mathbf{f}_j - \mathbf{e}_j\mathbf{f}_i)] \qquad \forall i \in \mathcal{B} \tag{A.1}$$

$$\mathbf{q}_i^g - q_i^d = -B_{ii}(\mathbf{e}_i^2 + \mathbf{f}_i^2) + \sum_{j\in\delta(i)} [-B_{ij}(\mathbf{e}_i\mathbf{e}_j + \mathbf{f}_i\mathbf{f}_j) - G_{ij}(\mathbf{e}_i\mathbf{f}_j - \mathbf{e}_j\mathbf{f}_i)] \qquad \forall i \in \mathcal{B} \tag{A.2}$$

$$\underline{V}_i^2 \leq \mathbf{e}_i^2 + \mathbf{f}_i^2 \leq \overline{V}_i^2 \qquad \forall i \in \mathcal{B} \tag{A.3}$$

$$p_i^{min} \leq \mathbf{p}_i^g \leq p_i^{max} \qquad \forall i \in \mathcal{B} \tag{A.4}$$

$$q_i^{min} \leq \mathbf{q}_i^g \leq q_i^{max} \qquad \forall i \in \mathcal{B} \tag{A.5}$$

This representation of the problem is called the rectangular formulation of ACOPF. Here, the objective function is a sum of convex functions C_i which in practice will be piecewise linear or quadratic, in both cases computationally cheap to model.

Since the only nonlinear terms are $(\mathbf{e}_i^2 + \mathbf{f}_i^2) = \mathbf{V}_i^*\mathbf{V}_i$, $\mathbf{e}_i\mathbf{f}_j - \mathbf{e}_j\mathbf{f}_i = \mathrm{Im}(\mathbf{V}_i^*\mathbf{V}_j)$ and $\mathbf{e}_i\mathbf{e}_j + \mathbf{f}_i\mathbf{f}_j = \mathrm{Re}(\mathbf{V}_i^*\mathbf{V}_j)$, all constraints are linear on the entries of the matrix $X = VV^*$. Finally, a matrix X admits the decomposition $X = VV^*$ if and only if X is semidefinite positive and has rank at most 1 (if the latter is true, the vectors in the outer product must be complex conjugates since X is Hermitian).

If we relax the rank constraint and only require that X be Hermitian and semidefinite positive, we obtain a convex problem, which is usually denoted as the semidefinite program relaxation or SDP relaxation of ACOPF. If we increase our domain beyond the semidefinite cone $X \succeq 0$, and only require that X is Hermitian and the submatrices $\begin{pmatrix} x_{ii} & x_{ij} \\ x_{ji} & x_{jj} \end{pmatrix}$ are semidefinite positive, which amounts to $x_{ii}x_{jj} - x_{ji}x_{ij} \geq 0$, we obtain the second order cone (SOC) relaxation of ACOPF.

In order to frame the SOC relaxation in terms of real variables, we can define $c_{ij} = \mathrm{Re}(x_{ij}) = \mathbf{e}_i\mathbf{e}_j + \mathbf{f}_i\mathbf{f}_j$ and $s_{ij} = \mathrm{Im}(x_{ij}) = \mathbf{e}_i\mathbf{f}_j - \mathbf{e}_j\mathbf{f}_i$. The matrix is Hermitian if $c_{ij} = c_{ji}$ and $s_{ij} = -s_{ji}$ for all i, j, and in particular $s_{ii} = 0 \leq c_{ii}$ in the diagonal. The SOC constraint then becomes $c_{ii}c_{jj} = x_{ii}x_{jj} \geq x_{ij}x_{ji} = c_{ij}^2 + s_{ij}^2$, which is equivalent to the 4-dimensional Lorenz cone $c_{ij}^2 + s_{ij}^2 + \left(\frac{c_{ii} - cjj}{2}\right)^2 \leq \left(\frac{c_{ii} + cjj}{2}\right)^2$.

However, if X has rank at most 1, the determinant of any 2×2 submatrix is zero, and $c_{ii}c_{jj} = c_{ij}^2 + s_{ij}^2$. This quadratic but non-convex constraint is enough to guarantee a feasible solution for radial networks, that is, if the graph $\mathcal{G}$ is a tree [17]. In order to extend it to general graphs, Jabr [24] proved that it suffices to include additional variables $\theta_i = \angle(V_i)$ that represent the angle of each complex voltage (that is, the phase of alternating current at node i) and require $\theta_i - \theta_j = \angle(c_{ij} + is_{ij})$. The full alternative formulation for ACOPF, called the *polar formulation*, is then

$$\min \sum_{i \in \mathcal{B}} C_i(\mathbf{p}_i^g)$$

subject to

$$\mathbf{p}_i^g - p_i^d = G_{ii}\mathbf{c}_{ii} + \sum_{j \in \delta(i)} [G_{ij}\mathbf{c}_{ij} - B_{ij}\mathbf{s}_{ij}] \qquad \forall i \in \mathcal{B} \tag{B.1}$$

$$\mathbf{q}_i^g - q_i^d = -B_{ii}\mathbf{c}_{ii} + \sum_{j \in \delta(i)} [-B_{ij}\mathbf{c}_{ij} - G_{ij}\mathbf{s}_{ij}] \qquad \forall i \in \mathcal{B} \tag{B.2}$$

$$\underline{V}_i^2 \leq \mathbf{c}_{ii} \leq \overline{V}_i^2 \qquad \forall i \in \mathcal{B} \tag{B.3}$$

$$\mathbf{c}_{ij} = \mathbf{c}_{ji}, \quad \mathbf{s}_{ij} = -\mathbf{s}_{ji} \qquad \forall (i,j) \in \mathcal{L} \tag{B.4}$$

$$\mathbf{c}_{ij}^2 + \mathbf{s}_{ij}^2 = \mathbf{c}_{ii}\mathbf{c}_{jj} \qquad \forall (i,j) \in \mathcal{L} \tag{B.5}$$

$$\boldsymbol{\theta}_j - \boldsymbol{\theta}_i = atan2(\mathbf{s}_{ij}, \mathbf{c}_{ij}) := \angle(\mathbf{c}_{ij} + i\mathbf{s}_{ij}) \qquad \forall (i,j) \in \mathcal{L} \tag{B.6}$$

$$p_i^{min} \leq \mathbf{p}_i^g \leq p_i^{max}, \quad q_i^{min} \leq \mathbf{q}_i^g \leq q_i^{max} \qquad \forall i \in \mathcal{B} \tag{B.7}$$

As mentioned above, this model is a simplification, and current instance libraries such as PGLib-OPF [2] add more features to the model. First, there may be several generators in the same bus, and several branches between the same pair of buses (i.e. G is not simple). The branches, which represent power lines or transformers, also have a maximum capacity s^u and may have transmission losses. And finally, there are bounds in the phase difference of adjacent nodes, i.e. $\theta_l^{\Delta l} \leq \angle(V_i^* V_j) \leq \theta_l^{\Delta u}$ for every branch l between i and j.

However, even after these modifications, most constraints are still linear in the real and imaginary parts of V^*V, so the SOC and SDP relaxations can be constructed in the same way. The only exception are the branch capacity constraints $|S_{l,il}| \leq s_l^u$ for each branch l, which are equivalent to the SOC constraints $\text{Re}(S_{l,ij})^2 + \text{Im}(S_{l,ij})^2 \leq s_l^{u2}$. Again, the power $S_{l,ij}$ transmitted through branch l from i to j depends linearly on c_{ij} and s_{ij}, so we only need to add convex constraints to either relaxation.

Chapter 3

Folding Constructions

3.1 Lorentz Cone Surface Constraints

Our starting point is a result of Ben-Tal and Nemirovsky [8]. They show that a second order cone of the form $x_3^2 \geq x_1^2 + x_2^2$ can be approximated by the polyhedron

$$\begin{aligned} &\text{(a)} \begin{cases} \xi_0 \geq |x_1| \\ \eta_0 \geq |x_2| \end{cases} \\ &\text{(b)} \begin{cases} \xi_j = \cos\left(\frac{\pi}{2^{j+1}}\right)\xi_{j-1} + \sin\left(\frac{\pi}{2^{j+1}}\right)\eta_{j-1} \\ \eta_j \geq \left|-\sin\left(\frac{\pi}{2^{j+1}}\right)\xi_{j-1} + \cos\left(\frac{\pi}{2^{j+1}}\right)\eta_{j-1}\right| \end{cases} \qquad \forall j \in [\![v]\!] \\ &\text{(c)} \begin{cases} \xi_v \leq x_3 \\ \eta_v \leq \tan\left(\frac{\pi}{2^{v+1}}\right)\xi_v \end{cases} \end{aligned} \tag{3.1}$$

Let Π_v is the polyhedron defined in 3.1, and let its projection in (x_1, x_2, x_3)-space be $\Pi'_v = \{(x_1, x_2, x_3) : (x_1, x_2, x_3, u) \in \Pi_v\}$,

Theorem 3.1 (Ben-Tal and Nemirovsky)

For $x \in \mathbb{R}^3$ *with* $x_3 \geq 0$, $x_3^2 \geq x_1^2 + x_2^2 \Rightarrow x \in \Pi'_v \Rightarrow x_3^2 \geq (x_1^2 + x_2^2)\cos\left(\frac{\pi}{2^{v+1}}\right)^2$

In other words, Π'_v is a relaxed approximation of the cone. Geometrically, Π'_v is a polyhedral cone, and on the plane $x_3 = r$ it approximates the circle $x_1^2 + x_2^2 \leq r^2$ by a circunscribed regular 2^{v+1}-gon. The approximation error $\varepsilon_v := 1 - \cos\left(\frac{\pi}{2^{v+1}}\right) = O(\frac{1}{4^v})$ decreases exponentially in the number of constraints, so the number of variables and constraints grows logarithmically on the error and the number of facets. In [8], the authors also prove that it takes at least $O(-\log(\varepsilon))$ variables to construct a relaxed approximation with error ε, so the formulation given by 3.1 is asymptotically optimal.

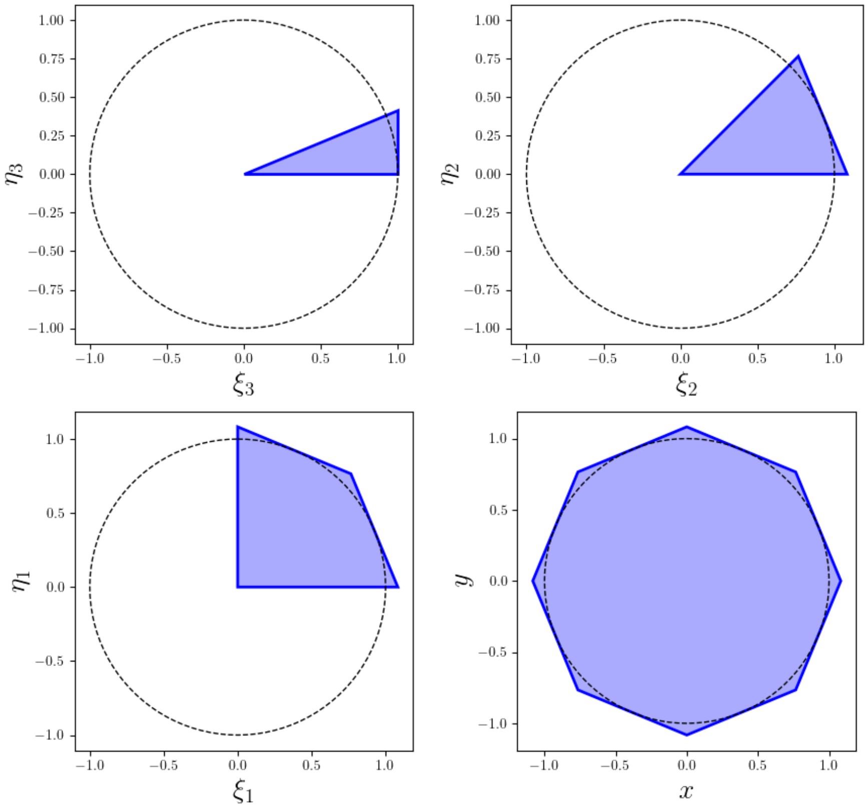


Figure 3.1 : Region delimited by Π_2' for fixed $x_3 = 1$

Replacing $\xi_v \le x_3$ by $\cos\left(\frac{\pi}{2^{v+2}}\right)\xi_v + \sin\left(\frac{\pi}{2^{v+2}}\right)\eta_v \le x_3$ in the original construction produces a regular 2^{v+2}-gon instead of a 2^{v+1}-gon when x_3 is fixed, and decreases the error to $1 - \cos(\frac{\pi}{2^{v+2}})$. We can also include the inequalities from (a) as two more cases of (b) for $j = 0, -1$, rewriting them as $\xi_{-1} = -x_1$, $\eta_{-1} \ge |x_2|$, $\xi_0 = \eta_{-1}$ and $\eta_0 \ge |\xi_{-1}|$. Denoting $\theta_j = \frac{\pi}{2^{j-1}}$, we end up with the equivalent formulation

$$\begin{aligned}
&\text{(a)}\begin{cases} \xi_0 = x_1 \\ \eta_0 = x_2 \end{cases} \\
&\text{(b)}\begin{cases} \xi_j = \cos\left(\theta_j\right)\xi_{j-1} + \sin\left(\theta_j\right)\eta_{j-1} \\ \eta_j \geq \left|-\sin\left(\theta_j\right)\xi_{j-1} + \cos\left(\theta_j\right)\eta_{j-1}\right| \end{cases} \qquad \forall j \in \llbracket v \rrbracket \\
&\text{(c)}\begin{cases} \cos\left(\theta_{v+1}\right)\xi_v + \sin\left(\theta_{v+1}\right)\eta_v \leq x_3 \\ \eta_v \leq \tan\left(\theta_v\right)\xi_v \end{cases}
\end{aligned} \tag{3.2}$$

Similarly, $x_3^2 \leq x_1^2 + x_2^2$ can be approximated by the projection of the set $\Pi_{(v)}$ defined by:

$$\begin{aligned}
&\text{(a)}\begin{cases} \xi_0 = x_1 \\ \eta_0 = x_2 \end{cases} \\
&\text{(b)}\begin{cases} \xi_j = \cos\left(\theta_j\right)\xi_{j-1} + \sin\left(\theta_j\right)\eta_{j-1} \\ 0 \leq \eta_j \leq \left|-\sin\left(\theta_j\right)\xi_{j-1} + \cos\left(\theta_j\right)\eta_{j-1}\right| \end{cases} \qquad \forall j \in \llbracket v \rrbracket \\
&\text{(c)}\begin{cases} \cos\left(\theta_{v+1}\right)\xi_v + \sin\left(\theta_{v+1}\right)\eta_v \geq \cos\left(\theta_{v+1}\right)x_3 \geq 0 \\ \eta_v \leq \tan\left(\theta_v\right)\xi_v \end{cases}
\end{aligned} \tag{3.3}$$

Theorem 3.2

The projection $\Pi'_{(v)}$ of $\Pi_{(v)}$ in (x_1, x_2, x_3) satisfies

$$0 \leq x_3 \leq \sqrt{x_1^2 + x_2^2} \quad \Rightarrow \quad x \in \Pi'_{(v)} \quad \Rightarrow \quad 0 \leq \cos\left(\frac{\pi}{2^v}\right)x_3 \leq \sqrt{x_1^2 + x_2^2} \tag{3.4}$$

Proof 3.1 For the first implication, if $\sqrt{x_1^2 + x_2^2} \geq x_3$, we can define $\xi_0 := x_1$, $\eta_0 := x_2$, and then recursively $\xi_j := \cos\left(\theta_j\right)\xi_{j-1} + \sin\left(\theta_j\right)\eta_{j-1}$ and $\eta_j := \left| -\sin\left(\theta_j\right)\xi_{j-1} + \cos\left(\theta_j\right)\eta_{j-1}\right|$, which is equivalent to rotating the point (ξ_{j-1}, η_{j-1}) an angle of $\theta_j = \frac{\pi}{2^{j-1}}$ clockwise and reflecting it in the x axis if it lies below, which preserves the distance to the origin, i.e. $\sqrt{\xi_j^2 + \eta_j^2} = \sqrt{x_1^2 + x_2^2} \geq x^3$ for every $j \in \llbracket v \rrbracket$. This ensures that (ξ_j, η_j) is in the increasingly smaller polyhedron defined by $\eta_j \geq 0$ and $\eta_j \leq \tan\left(\frac{\pi}{2^{j-1}}\right)\xi_j$, and after v iterations, $0 \leq \eta_v \leq \xi_v \tan\left(\frac{\pi}{2^{v-1}}\right)$. Finally, the points in that polyhedron more than x_3 units away from the origin satisfy the remaining inequality $\cos(\theta_{v+1})\xi_v + \sin(\theta_{v+1})\eta_v \geq \cos(\theta_{v+1})x_3$:

$$2\sin(\theta_{v+1})\cos(\theta_{v+1})\xi_v = \sin(\theta_v)\xi_v \geq \cos(\theta_v)\eta_v = (\cos(\theta_{v+1})^2 - \sin(\theta_{v+1})^2)\eta_v$$

$$(\cos(\theta_{v+1})\xi_v + \sin(\theta_{v+1})\eta_v)^2 = \cos(\theta_{v+1})^2\xi_v^2 + 2\cos(\theta_{v+1})\sin(\theta_{v+1})\xi_v\eta_v + \sin(\theta_{v+1})^2\eta_v^2 \geq$$

$$\geq \cos(\theta_{v+1})\xi_v^2 + (\cos(\theta_{v+1}^2 - \sin(\theta_{v+1})^2)\eta_v^2 + \sin(\theta_{v+1})\eta_v^2 = \cos(\theta_{v+1})^2(\xi_v^2+\eta_v^2) = \cos(\theta_{v+1})^2 x_3^2$$

Since x_3, η_v and ξ_v are positive, we can remove the squares and (c) holds, so the point we constructed is in $\Pi_{(v)}$ and therefore $(x_1, x_2, x_3) \in \Pi'_{(v)}$.

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

$$(\xi_j)^2 + (\eta_j)^2 \leq (\cos(\theta_j)\xi_{j-1} + \sin(\theta_j)\eta_{j-1})^2 + (-\sin(\theta_j)\xi_{j-1} + \cos(\theta_j)\eta_{j-1})^2 = (\xi_{j-1})^2 + (\eta_{j-1})^2$$

So by induction in j, we get $x_1^2 + x_2^2 = (\xi_0)^2 + (\eta_0)^2 \geq \cdots \geq (\xi_v)^2 + (\eta_v)^2$, and by the scalar product inequality and constraints (c),

$$(\xi_v)^2 + (\eta_v)^2 = \|(\xi_v, \eta_v)\|_2^2 \cdot \|(\cos(\theta_{v+1}), \sin(\theta_{v+1}))\|_2^2 \geq (\cos(\theta_{v+1})\xi_v + \sin(\theta_{v+1})\eta_v)^2 \geq (\cos(\theta_{v+1})x_3)^2$$

so $\sqrt{x_1^2 + x_2^2} \geq \cos(\theta_{v+1})x_3 = \cos\left(\frac{\pi}{2^v}\right)x_3$. $\square$

The inequalities in Π_v with an absolute value can be written as pairs of linear inequalities, namely $\eta_j \geq -\sin(\theta_j)\xi_{j-1} + \cos(\theta_j)\eta_{j-1}$ and $\eta_j \geq \sin(\theta_j)\xi_{j-1} - \cos(\theta_j)\eta_{j-1}$, but in our formulation of $\Pi_{(v)}$ this is not possible (since the set is not convex). However, we can write them as disjunctive constraints where either $0 \leq \eta_j \leq -\sin(\theta_j)\xi_{j-1} + \cos(\theta_j)\eta_{j-1}$ or $0 \leq \eta_j \leq \sin(\theta_j)\xi_{j-1} - \cos(\theta_j)\eta_{j-1}$ is satisfied. In fact, we can show that this disjunctive formulation of $\Pi_{(v)}$ is optimal in terms of the number of disjunctions.

Theorem 3.3

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 3.2 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 $\bar{P} = \bigcap_{i=1}^{v}\left(\bar{Q}_i^0 \cup \bar{Q}_i^1\right)$

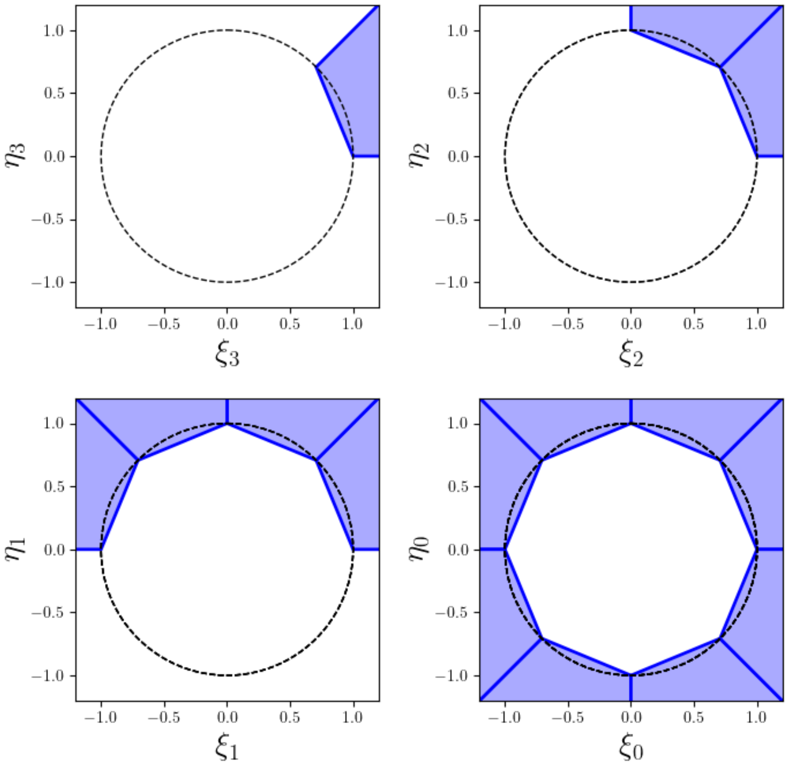


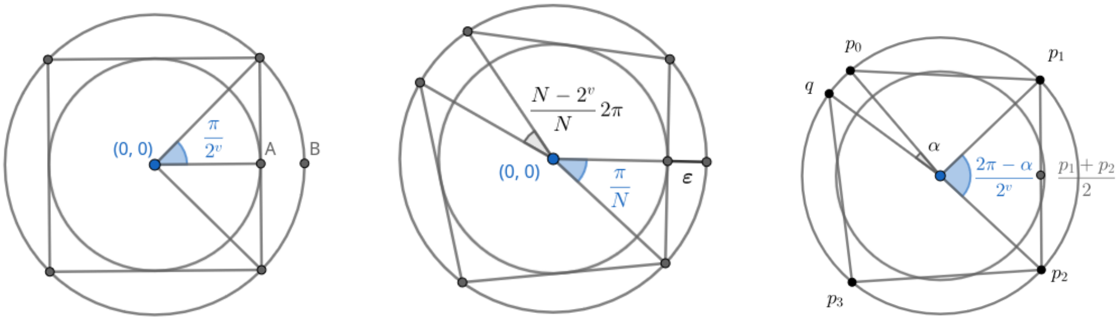
Figure 3.2 : Region delimited by $\Pi'_{(3)}$ for fixed $x_3 = 1$

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 $j - 1 \in \{0, 1, \dots 2^v\}$. 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 closer than a certain distance 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 $\frac{N-2^v}{N}2\pi$, i.e. $p_0 = (\cos(\phi), \sin(\phi)), q = (\cos(\phi+\alpha), \sin(\phi+\alpha))$, 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 $0 \le i < j \le 2^v - 1$, so their midpoint is at most $\cos(\frac{\pi}{N}) = 1 - \varepsilon$ units away from the origin, and therefore not in P'.



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

(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

Figure 3.3 : Proof sketch for $v = 2$

This means no two points in $\{p_i : 0 \le i \le 2^v - 1\}$ lie in the same polyhedron P^j, and each of them must be in exactly one of the polyhedra. The same is true for $\{p_i : 1 \le i \le 2^v - 1\} \cup \{q\}$, which is the same as the previous set rotated by $\frac{2\pi-\alpha}{2^v}$ radians. 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. □

This shows that our formulation is the most precise among those with v disjunctions and, therefore, that we cannot use less than $O(-\log(\varepsilon))$ disjunctions to achieve an approximation error of ε. To understand the construction, it will be helpful to combine both inequalities in the following relaxed approximation to $S := \{x \in \mathbb{R}^3 : x_3 = \sqrt{x_1^2 + x_2^2}\}$:

$$\begin{array}{lll}
\text{(a)} \left\{ \begin{array}{l} \xi_0 = x_1 \\[6pt] \eta_0 = x_2 \end{array} \right. & & \\[10pt]
\text{(b)} \left\{ \begin{array}{l} \xi_j = \cos\left(\theta_j\right) \xi_{j-1} + \sin\left(\theta_j\right) \eta_{j-1} \\[6pt] \eta_j = \left|-\sin\left(\theta_j\right) \xi_{j-1} + \cos\left(\theta_j\right) \eta_{j-1}\right| \end{array} \right. & \forall j \in [\![v]\!] & \\[10pt]
\text{(c)} \left\{ \begin{array}{l} \cos\left(\theta_{j+1}\right) x_3 \leq \cos\left(\theta_{j+1}\right) \xi_v + \sin\left(\theta_{j+1}\right) \eta_v \leq x_3 \\[6pt] \eta_v \leq \tan\left(\theta_j\right) \xi_v \end{array} \right. & &
\end{array} \tag{3.5}$$

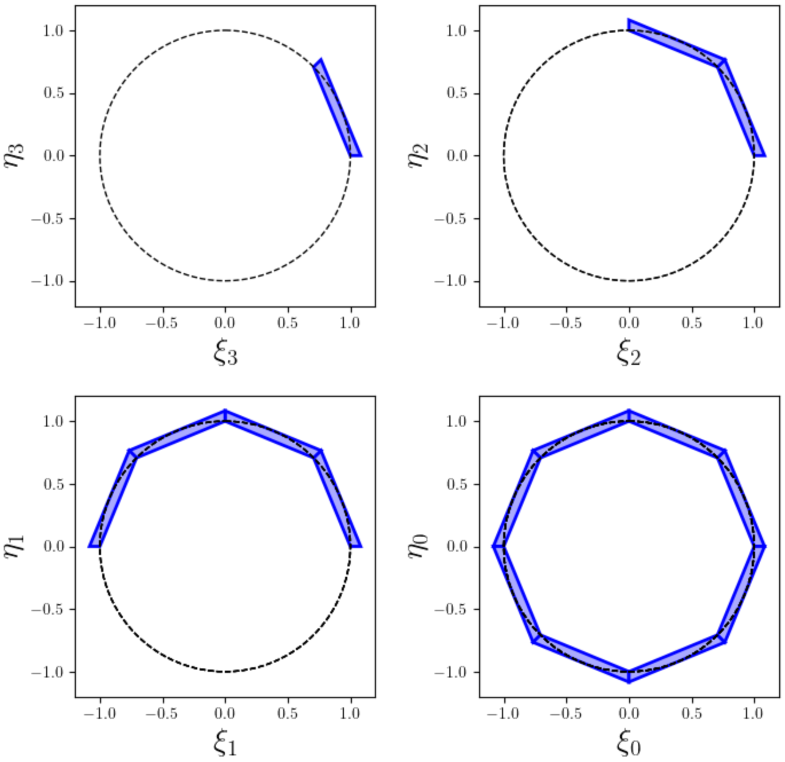


Figure 3.4 : Region delimited by the system 3.5 for $v = 3$ and fixed $x_3 = 1$

The set defined by 3.5 is obviously contained in both 3.2 and 3.3, so its projection in the space of x_1, x_2 and x_3 satisfies $\cos(\frac{\pi}{2^{v+2}})^2 x_3^2 \leq x_1^2 + x_2^2 \leq \cos(\frac{\pi}{2^{v+2}})^{-2} x_3^2$. However, now the constraints in (b) can be understood as the combination of a rotation by an angle of $\frac{\pi}{2^{j+1}}$ clockwise, followed by a "fold" along the x axis (or rather, the ξ_j axis). After v repetitions, the plane has been mapped to the points at most $\frac{\pi}{2^{v+1}}$ degrees from the horizontal axis, and in fact, the last constraint in (c) is redundant. Since for any fixed x_3, the region defined by (c) in the (ξ_v, η_v) plane contains the arc $(x_3 \cos(t), x_3 \sin(t))$ for $t \in [0, \frac{\pi}{2^{v+1}}]$, and satisfies $\sqrt{x_1^2 + x_2^2} \in [\cos(\frac{\pi}{2^{v+2}})x_3, \cos^{-1}(\frac{\pi}{2^{v+2}})x_3]$, we have a relaxed approximation for the conical surface $\sqrt{x_1^2 + x_2^2} = x_3$.

It is well known that any polynomial constraint can be reduced to quadratic constraints and linear inequalities [18], and a quadratic constraint $y = x^2$ is equivalent to the cone surface constraint $(y+1)^2 = (y-1)^2 + (2x)^2$. Therefore, we can construct a relaxed approximation to any polynomial optimization problem with sets of disjunctive constraints of the form 3.5 and linear inequalities. For instance, in the case of conic curves like the parabola $y = x^2$ a single conical surface is required, and for a n-dimensional conical surface $x_n^2 = \sum_{i=1}^{n-1} x_i^2$ we need $n-2$ three-dimensional constraints, as per the construction in [8].

If our feasible region is not the whole surface, but just the projection of an arc $(x_1, x_2) = (x_3 \cos(\alpha), x_3 \sin(\alpha))$ for $\alpha \in [\alpha^-, \alpha^+]$, we can gain precision if we instead define θ_j as $2^{-j}(\alpha^+ - \alpha^-)$ and replace constraints (a) by $\xi_0 = \cos(\alpha^-)x_1 + \sin(\alpha^-)x_2$ and $\eta_0 = \cos(\alpha^-)x_2 - \sin(\alpha^-)x_1$. This ensures that all 2^v polytopes are used in the feasible region, which does not happen if we just add linear cuts to 3.5.

3.1.1 MILP formulations

Although in theory a disjunctive program can be solved by branch-and-bound, most solvers are designed to work with mixed integer linear programs. However, we cannot construct a relaxed approximation of S with a MILP without bounding the domain of some variables.

Theorem 3.4

A set of the form $X := \{x \in \mathbb{R}^n : \exists (y, z) \in \mathbb{R}^m \times \{0,1\}^v, \quad Ax + A'y + A''z \leq b\}$ can be expressed as $C + \bigcup_i^{2^v} Q_i$ where C is a polyhedral cone and Q_i are polytopes.

Proof 3.3 We can decompose X as the union of polyhedra $X = \bigcup_{z \in \{0,1\}^v} \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^m : Ax + A'y \leq b - A''z\}$ and each $P_z = \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^m : Ax + A'y \leq b - A''z\}$ has the same recession cone $C := rec(P_z) = \{d \in \mathbb{R}^n : Ad \leq 0\}$. Using the Decomposition Theorem for polyhedra, we can find 2^v polytopes Q_z such that $P_z = Q_z + C$ and $X = \bigcup_{z \in \{0,1\}^v} (Q_z + C) = C + \bigcup_{z \in \{0,1\}^v} Q_z$. $\square$

Since our cone surface contains the half-lines $\{(\lambda, 0, \lambda) : \lambda \in \mathbb{R}^+\}$ and $\{(-\lambda, 0, \lambda) : \lambda \in \mathbb{R}^+\}$, any MILP relaxation must contain both in some polyhedron, and therefore in C, which is convex and also contains the x_3 axis. This implies that the whole x_3 axis must be in that relaxation, because the origin is in S, and a similar argument shows that the whole convex cone $x_3 \geq \sqrt{x_1^2 + x_2^2}$ would be in any MILP relaxation.

However, the set $S_K := \{x \in S : x_3 \leq K\}$ is bounded, and therefore it can be approximated with a union of polytopes, which can be defined by mixed integer linear constraints, for instance by as we saw in Chapter 2. Instead, we can replace the disjunctions between $\eta_j \leq -\sin(\theta_j)\xi_{j-1} + \cos(\theta_j)\eta_{j-1}$ and $\eta_j \leq \sin(\theta_j)\xi_{j-1} - \cos(\theta_j)\eta_{j-1}$ by the constraints $\eta_j \leq -\sin(\theta_j)\xi_{j-1} + \cos(\theta_j)\eta_{j-1} + M_j\lambda_j$ and $\eta_j \leq \sin(\theta_j)\xi_{j-1} - \cos(\theta_j)\eta_{j-1} + M_j(1 - \lambda_j)$, where $\lambda_j \in \{0, 1\}$ is a binary variable, and M_j is a constant.

If the two big-M constraints above hold, it is easy to see that the disjunctive constraint also holds. In order to ensure that every point in S_k is in the set defined by the big-M constraints, we recall that the point constructed in the proof of satisfies $\eta_j \leq \xi_j^2 + \eta_j^2 = x_1^2 + x_2^2 = x_3^2 \leq K^2$ so if $\eta_j = |\sin(\theta_j)\xi_{j-1} - \cos(\theta_j)\eta_{j-1}|$, we have $|\eta_j \pm (\sin(\theta_j)\xi_{j-1} + \cos(\theta_j)\eta_{j-1})| \leq |\eta_j| + \eta_j \leq 2K$ so $M_j = 2K$ is big enough.

Better still, we saw in that proof that the angle of (ξ_j, η_j) is between 0 and $\theta_j = \frac{\pi}{2^{j-1}}$, so $\eta_j \leq \sin(\theta_j)\sqrt{\xi_j^2 + \eta_j^2} \leq \sin(\theta_j)K$. This allows us to use $M_j = 2\sin(\theta_j)K \approx \frac{\pi K}{2^{j-2}}$.

3.2 Helix Construction

We consider the three-dimensional helix $H := \{(x_3\cos(\alpha), x_3\sin(\alpha), \alpha) : \alpha \in [0, 2\pi]\}$. An approximation to this set can be constructed by the same approach as before, with the following disjunctive program:

$$\begin{array}{ll}
\text{(a)} \begin{cases} \xi_0 = x_1 \\ \eta_0 = x_2 \\ \mu_0 = \alpha \end{cases} & \\
\text{(b)} \begin{cases} \xi_j = \cos(\theta_j)\,\xi_{j-1} + \sin(\theta_j)\,\eta_{j-1} \\ \eta_j = -\sin(\theta_j)\,\xi_{j-1} + \cos(\theta_j)\,\eta_{j-1} \\ \mu_j = \mu_{j-1} - \theta_j \end{cases} \text{or} & \begin{array}{l} \xi_j = \cos(\theta_j)\,\xi_{j-1} + \sin(\theta_j)\,\eta_{j-1} \\ \eta_j = \sin(\theta_j)\,\xi_{j-1} - \cos(\theta_j)\,\eta_{j-1} \\ \mu_j = -\mu_{j-1} + \theta_j \end{array} \\
\text{(c)} \begin{cases} \cos(\theta_v)\,x_3 \le \cos(\theta_{v+1})\,\xi_v + \sin(\theta_{v+1})\,\eta_v \le x_3 \\ 0 \le \eta_v \le \tan(\theta_v)\,\xi_v \\ 0 \le \mu_v \le \theta_v \end{cases} &
\end{array} \tag{3.6}$$

The construction is the same as for the circumference $x_1^2 + x_2^2 = x_3^2$ for fixed x_3, but now besides rotating the (ξ_j, η_j) plane an angle of θ_j, we add a translation of θ_j in the μ_j axis, and instead of reflecting in the ξ_j axis we rotate the space (ξ_j, η_j, μ_j) an angle of π along the direction ξ_j. Since both transformations map H to itself, we are "unfolding" a relaxation of the helix for $\alpha \in [0, \theta_j]$ to get a relaxation in the domain $[0, 2\pi]$.

Theorem 3.5

The projection in (x_1, x_2, α) space of the set delimited by 3.6 contains the helix H and satisfies $|\angle(x_1 + x_2 i) - \alpha| \le \frac{\pi}{2^{v-1}}$ and $x_3^2\cos\left(\frac{\pi}{2^v}\right)^2 \le x_1^2 + x_2^2 \le x_3^2\cos\left(\frac{\pi}{2^v}\right)^{-2}$

Proof 3.4 Given any point $(x_1, x_2, \alpha) \in H$, we can define another point in 3.6 by choosing $(\xi_0, \eta_0, \mu_0) := (x_1, x_2, \alpha)$ and then recursively setting $\xi_j := \cos(\theta_j)\,\xi_{j-1} + \sin(\theta_j)\,\eta_{j-1}$, $\eta_j := |-\sin(\theta_j)\xi_{j-1} + \cos(\theta_j)\eta_{j-1}|$, and $\mu_j := |\mu_{j-1} - \theta_j|$. We claim that $\mu_j = \angle(\xi_j, \eta_j)$ at every step show it by induction: for $j = 0$ it follows from the definition of H, and if it holds for $j-1$, then since $\angle(\cos(\theta_j)\,\xi_{j-1}+\sin(\theta_j)\,\eta_{j-1}, -\sin(\theta_j)\xi_{j-1}+\cos(\theta_j)\eta_{j-1}) = \mu_{j-1}$, we have two cases: if $\mu_{j-1} \geq \theta_j$, we are done, since the arguments of the absolute values are both non-negative, and if $\mu_{j-1} \leq \theta_j$, they are both negative, and we only need to use $\angle(a,-b) = -\angle(a,b)$.

This implies that $\mu_j - \theta$ and $-\sin(\theta_j)\,\xi_{j-1} + \cos(\theta_j)\,\eta_{j-1}$ have the same sign, and the point satisfies the constraints in (b). Now, we already know that the first two constraints in (c) hold from theorems 3.1 and 3.2 and it is easy to see that $0 \leq \mu_j \leq \theta_j$ for every j, so the final constraint holds. The bounds on $x_1^2 + x_2^2$ also follow from 3.1.1.

Finally, if a point is feasible in 3.6, $\angle(\xi_j, \eta_j)$ and μ_j can be obtained from $\angle(\xi_v, \eta_v)$ and μ_v respectively via a composition of the same sequence of functions $x \mapsto x+\theta_j$ or $x \mapsto \theta_j - 1$ (depending on which clause of each disjunction is true). Since these maps preserve distances, $|\angle(x_1, x_2) - \alpha| = |\angle(\xi_0, \eta_0) - \mu_0| = |\angle(\xi_v, \eta_v) - \mu_v|$ and that last quantity is at most θ_v, because (c) implies $\angle(\xi_v, \eta_v), \mu_v \in [0, \theta_v]$. □

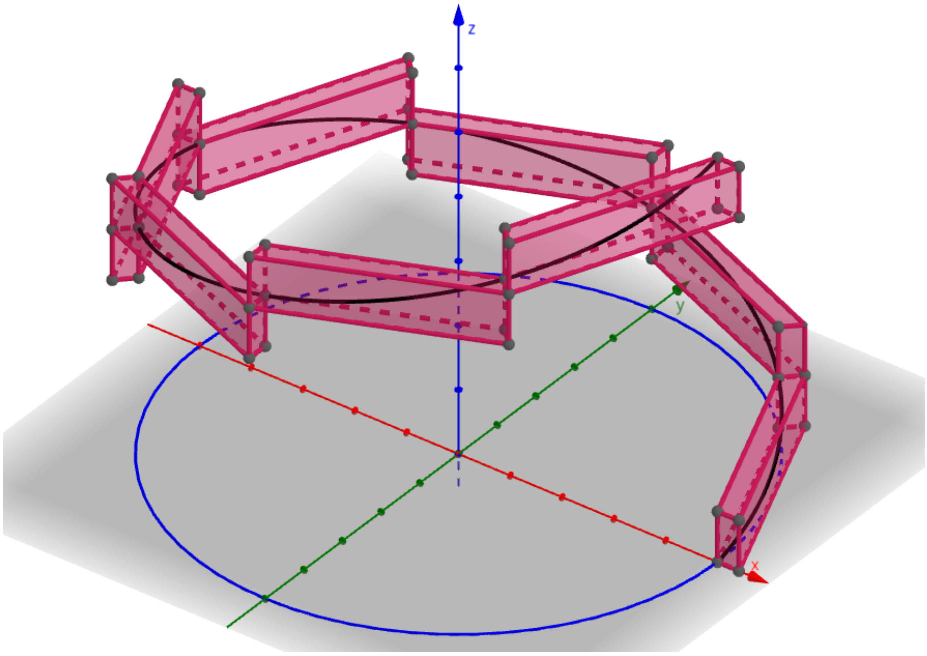


Figure 3.5 : Region delimited by 3.6 for fixed $x_3 = 1$

Although for the sake of simplicity and easier visualization of H we assumed that x_3 was constant, making it a variable still results in a mixed integer linear program, so in effect we have constructed an approximation to $H^4 := \{(x_3 \sin(\alpha), x_3 \cos(\alpha), x_3, \alpha) : x_3 \in \mathbb{R}^+, \alpha \in [0, 2\pi]\}$. The projection of H^4 in (x_1, x_2, x_3) is just our original conical surface S, so we have modified 3.5 to obtain an approximation of $\angle(x_1, x_2)$ with error at most $\theta_v = O(2^{-v})$ without adding more disjunctions.

3.2.1 Sine and cosine constraints

If we fix $x_3 = 1$ in 3.6 and in the definition of H, its projection in the (α, x_1) plane is the graph of $x_1 = \cos(\alpha)$ for $\alpha \in [0, 2\pi]$, so the projection of 3.6 in that plane will approximate the constraint $x_1 = \cos(\alpha)$. The same is true of $x_2 = \sin(\alpha)$ in the plane (x_2, α).

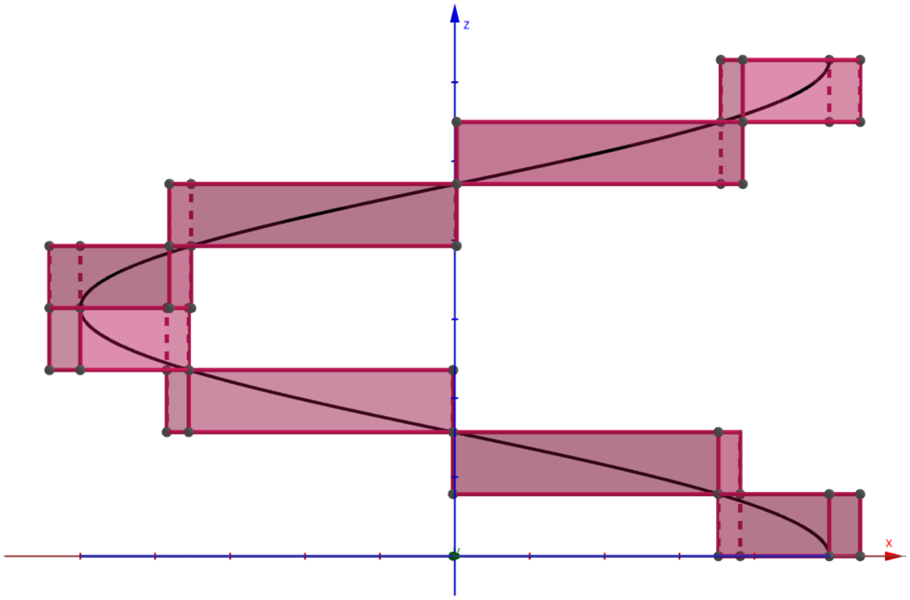


Figure 3.6 : Region delimited by 3.6 for fixed $x_3 = 1$ in the plane (α, x_1)

The approximation error $|x_1 - \cos(\alpha)|$ can be bounded by

$$\varepsilon_v = |x_1 - \cos(\alpha)| \leq |x_1 - \cos(\angle(x_1, x_2))| + |\cos(\angle(x_1, x_2)) - \cos(\alpha)| =$$

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

using the Mean Value Theorem and theorem 3.5. Since $\cos(\theta_v)^{-1} - 1 \approx \frac{1}{2}(\theta_v)^2 = O(4^{-v})$, the error is dominated by the second term $\theta_v = \frac{\pi}{2^{v-1}} = O(2^{-v})$, and therefore the disjunctions needed to achieve an error of v are at most $v = O(-\log(\varepsilon))$. We can prove that, in fact, this is the most compact formulation in terms of binary variables up to a constant factor.

Theorem 3.6

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 $|x_1 - \cos(\alpha)| \leq \varepsilon$ for all $(x_1, \alpha, u) \in P$, then $v \geq -C \log(\varepsilon)$ for some positive constant C.

Proof 3.5 We argue by contradiction, and assume $P_\varepsilon(x, y, u)$ approximates $y = \cos(x)$ in 2π 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 $\{x_1^2 + x_2^2 = 1\}$ 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 $(\alpha, x_2, u_2) \in Q_\varepsilon$. Conversely, 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 $|x_1^2 + x_2^2 - 1| \leq |x_1^2 - \cos(\alpha)^2| + |x_2^2 - \sin(\alpha)^2| \leq \varepsilon|\cos(\alpha) + x_1| + \varepsilon|\sin(\alpha) + x_2| \leq \varepsilon(2\cos(\alpha) + \varepsilon + 2\sin(\alpha) + \varepsilon) \leq 2\varepsilon^2 + 4\varepsilon$ as we wanted.

Finally, we showed in theorem 3.1.1 that a relaxation of the circumference $x_1^2 + x_2^2 = 1$ with w disjunctions can only achieve an error bound of $1 - \sqrt{x_1^2 - x_2^2} \leq 1 - \cos\left(\frac{\pi}{2^{w-1}}\right)$, which translates to $x_1^2 + x_2^2 \geq \cos\left(\frac{\pi}{2^{w-1}}\right)^2$. 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

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

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

which contradicts our hypothesis. □

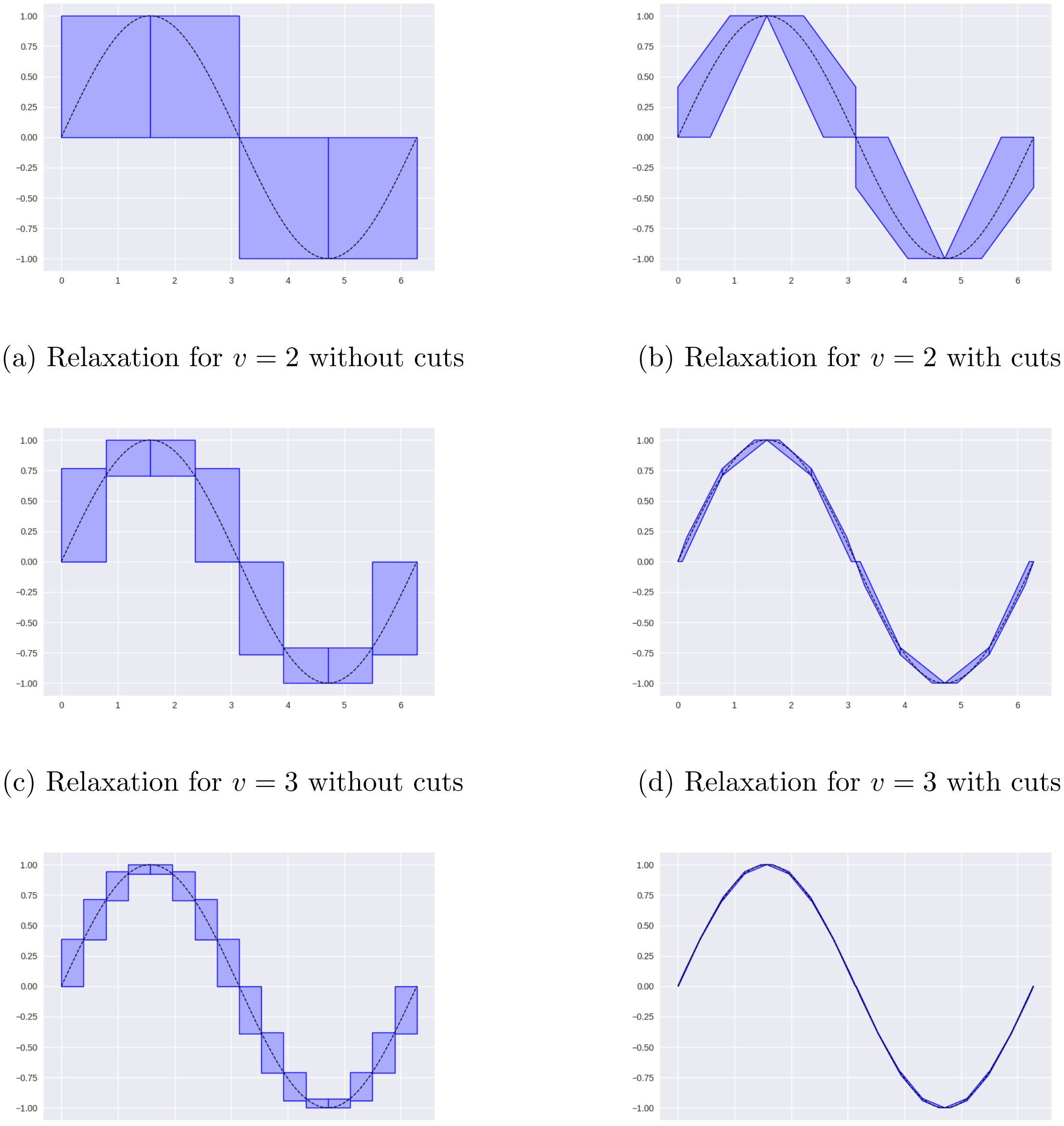
3.2.2 MILP formulations

If x_3 is bounded above by some constant K, we can translate the formulation to a MILP:

$$\begin{aligned}
&(a)\begin{cases} \xi_0 = x_1 \\ \eta_0 = x_2 \\ \mu_0 = \alpha \end{cases} \\
&(b)\begin{cases} \xi_j = \cos(\theta_j)\,\xi_{j-1} + \sin(\theta_j)\,\eta_{j-1} \\ \eta_j \geq -\sin(\theta_j)\,\xi_{j-1} + \cos(\theta_j)\,\eta_{j-1} \\ \eta_j \leq -\sin(\theta_j)\,\xi_{j-1} + \cos(\theta_j)\,\eta_{j-1} + M_j\lambda_j \\ \eta_j \geq \sin(\theta_j)\,\xi_{j-1} - \cos(\theta_j)\,\eta_{j-1} \\ \eta_j \leq \sin(\theta_j)\,\xi_{j-1} - \cos(\theta_j)\,\eta_{j-1} + M_j(1-\lambda_j) \\ \mu_j \geq \mu_{j-1} - \theta_j \\ \mu_j \leq \mu_{j-1} - \theta_j + \theta_{j-1}\lambda_j \\ \mu_j \geq -\mu_{j-1} + \theta_j \\ \mu_j \leq -\mu_{j-1} + \theta_j + \theta_{j-1}(1-\lambda_j) \\ \lambda_j \in \{0,1\} \end{cases} \\
&(c)\begin{cases} \cos\left(\frac{\pi}{2^v}\right) x_3 \leq \cos\left(\frac{\pi}{2^v}\right)\xi_v + \sin\left(\frac{\pi}{2^v}\right)\eta_v \leq x_3 \\ \eta_v \leq \tan(\theta_v)\,\xi_v \\ 0 \leq \mu_v \leq \theta_v \end{cases}
\end{aligned} \tag{3.7}$$

where we can choose the constants $M_j = 2\sin(\theta_j)K$, just as in the conical surface. The feasible point constructed in the proof of Theorem 3.5 is also feasible in 3.7, and the reasoning is the same as in Section 3.1.1.

However, we can see in the figures 3.5 and 3.6 that the relaxation 3.7 is a very crude envelope of the helix. The approximation error can be improved by adding cuts to the prism



(a) Relaxation for $v = 2$ without cuts

(b) Relaxation for $v = 2$ with cuts

(c) Relaxation for $v = 3$ without cuts

(d) Relaxation for $v = 3$ with cuts

(e) Relaxation for $v = 4$ without cuts

(f) Relaxation for $v = 4$ with cuts

Figure 3.7 : Relaxations of the sine function produced by 3.7 with and without the cuts $\eta_v \leq \mu_v$ and $\eta_v \geq \theta_v^{-1}\sin(\theta_v)\mu_v$

(c) that are still valid for the helix arc contained in that prism.

For example, since $\mu_v \geq 0$, our upper bound K ensures $\eta_v = x_3 \sin(\mu_v) \leq x_3\mu_v \leq K\mu_v$. If we also have a lower bound $x_3 \geq K'$, we can use the fact that the sine function is concave in $[0, \theta_v]$, and hence its graph is above the segment between its endpoints, so $\eta_v = x_3 \sin(\mu_v) \geq x_3\mu_v \frac{\sin(\theta_v)}{\theta_v} \geq K'\theta_v^{-1} \sin(\theta_v)\mu_v$.

If we rotate the figure θ_v degrees in the x_1, x_2 plane, we obtain $\cos(\theta_v)\eta_v + \sin(\theta_v)\xi_v = -\sin(\theta_v - \mu_v)$ and the same bounds of the sine function in $[-\theta_v, 0]$ produce cuts in different directions. Finally, we can use a tangent to the sine function at any point $\theta \in [0, \theta_v]$ for an upper bound, i.e. $\eta_v \leq \sin(\theta) + \cos(\theta)(\mu_v - \theta)$.

In the case of constraints $x_1 = \cos(\alpha)$ or $x_2 = \sin(\alpha)$, we have $x_3 = 1$, and just adding the first two cuts $\eta_v \leq \mu_v$ and $\theta_v\eta_v \geq \sin(\theta_v)\mu_v$ reduces the error from $O(2^{-v})$ to $O(4^{-v})$, the same asymptotic bound we had for the conical surface:

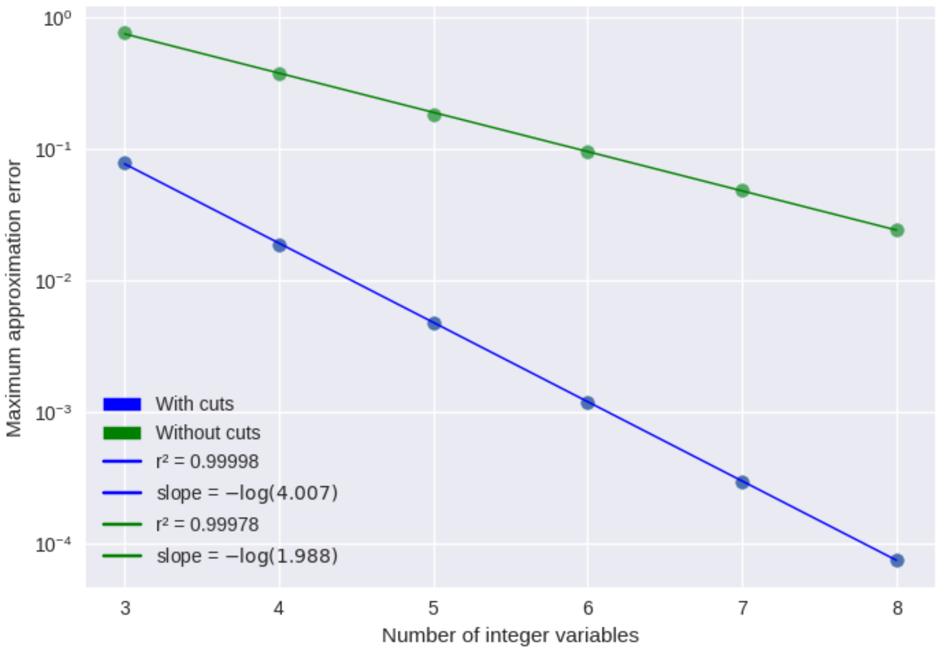


Figure 3.8 : Empirical measure of maximum approximation error $|sin(\alpha) - x_2|$ for 3.7

3.3 A General Approach

The obvious question is whether these constructions can be generalized to other types of constraints and, if they can, which families of constraints are appropriate. In order to answer that question, we take a step back to analyze the "folding" construction more abstractly. In Sections 3.1 and 3.2, we worked with a feasible region that can be parametrized as $x \in \{f(t, y) : t \in [0, T], y \in Y\} =: f([0, T], Y)$.

All of the constructions above have been divided into three sets of inequalities (a), (b), and (c). The first set is purely cosmetic, or a simple affine transformation to change the domain of the constraint, not its shape. The constraints in (b) "fold" the domain of a feasible region onto itself, mapping $f([0, \frac{T}{2}], Y)$ to $f([\frac{T}{2}, T], Y)$. Finally, (c) is just a polytope that acts as a relaxed approximation of $f([0, 2^{-v}T], Y)$.

The fact that we can represent this as a MILP is possible because the "folding" map is an affine transformation, that is, because f has some sort of affine self-similarity that allows us to recover any piece of its domain from any other, as $f(t+a, y) = A(a)f(t, y) + b(a)$. For example, the parabola in [6], the conical surface, the helix, and other sets like the graph of the exponential function in $\mathbb{R}$ have that property:

Given that representation, it is easier to find a compact relaxed approximation for the exponential $y = \exp(x)$ in a bounded domain $x \in [0, T]$, for instance

$$
\begin{aligned}
&(\mathrm{a}) \begin{cases} \xi_0 = x_1 \\ \eta_0 = x_2 \end{cases} \\
&(\mathrm{b}) \begin{cases} \xi_j = \xi_{j-1} - \frac{T}{2^j} \\ \eta_j = \exp\left(-\frac{T}{2^j}\right) \eta_{j-1} \end{cases} \quad \text{or} \quad \begin{matrix} \xi_j = \xi_{j-1} \\ \eta_j = \eta_{j-1} \end{matrix} \qquad \forall j \in [\![v]\!] \\
&(\mathrm{c}) \begin{cases} \eta_v \geq 1 + \xi_v \\ \eta_v - \exp\left(\frac{T}{2^v}\right) \geq \exp\left(\frac{T}{2^v}\right)\left(\xi_v - \frac{T}{2^v}\right) \\ \frac{T}{2^v}(\eta_v - 1) \leq \left(\exp\left(\frac{T}{2^v}\right) - 1\right)\xi_v \end{cases}
\end{aligned}
\tag{3.8}
$$

Constraint	Implicit function f	Folding property
$x_2 = x_1^2$	$f(t) = (t, t^2)$	$f(t+a) = \begin{pmatrix} 1 & 0 \\ 2a & 1 \end{pmatrix} f(t) + \begin{pmatrix} a \\ a^2 \end{pmatrix}$
$x_1^2 + x_2^2 = x_3^2$	$f(t,y) = (y\sin(t), y\cos(t), y)$	$f(t+a) = \begin{pmatrix} \cos(a) & -\sin(a) & 0 \\ \sin(a) & \cos(a) & 0 \\ 0 & 0 & 1 \end{pmatrix} f(t,y)$
$x_1 + ix_2 = x_3\exp(ix_4)$	$f(t,y) = (y\sin(t), y\cos(t), y, t)$	Similar as above
$x_2 = \exp(x_1)$	$f(t) = (t, \exp(t))$	$f(t+a) = \begin{pmatrix} 1 & 0 \\ 0 & \exp(a) \end{pmatrix} f(t) + \begin{pmatrix} a \\ 0 \end{pmatrix}$

Table 3.1 : Some constraints that admit folding constructions

3.4 Comparison with CDC Constructions

We have already established the improvement in memory use of folding constructions over CDC formulations. In the latter, any representation of the constraint $\lambda \in CDC(\mathcal{S})$ necessarily uses as many variables λ_v as the number of total extreme points of the polytopes used. In the case of the construction 3.5, which produces 2^{v+1} extreme points in total at depth v, we only need $5v+5$ linear inequalities and $3v$ additional variables, including v binary variables. This number can be reduced even more to $2v$ variables, as detailed in [8].

On the other hand, our formulation is no longer ideal, in contrast to the CDC-based formulations in [35]. This may not be so useful when dealing with the intersection of many approximated constraints, since in that case relaxing each of them to each convex hull is not correct. However, it obviously incurs a huge computational cost when we only have a single constraint, as the integer variables cannot be relaxed.

Another important question is how the branching algorithm behaves in each case. Since the folding maps in Section 3.1 and Section 3.2 are reflections in the (x_1, x_2) plane, the binary variable values corresponding to each polytope are ordered following the binary reflected Gray code described in [23], and it has the same branching behavior as the approximations of the parabola in [6] or the annulus in [22].

However, if we use a different folding transformation that involves a rotation instead of a reflection, we could achieve the same branching behavior as for the zig-zag formulation proposed by Huchette and Vielma in [23], using the following disjunctive formulation:

$$\begin{cases} \xi_j = \cos(\theta_j)\,\xi_{j-1} + \sin(\theta_j)\,\eta_{j-1} & \qquad \xi_j = \xi_{j-1} \\ \eta_j = -\sin(\theta_j)\,\xi_{j-1} + \cos(\theta_j)\,\eta_{j-1} & \text{or} \quad \eta_j = \eta_{j-1} \\ \mu_j = \mu_{j-1} - \theta_j & \qquad \mu_j = \mu_{j-1} \end{cases} \tag{3.9}$$

If we encode the disjunctions with binary variables z_j, and force the first clause of 3.9(b) when $z_j = 0$ and the second one when $z_j = 1$, we have $z_j = 1 \Rightarrow \angle(\xi_j + i\eta_j) = \angle(\xi_{j-1} + i\eta_{j-i}) - \theta_j$. If $z_j = 0$, the angle is the same, so $\angle(\xi_j + i\eta_j) = \angle(\xi_{j-1} + i\eta_{j-i}) - \theta_j z_j$, and by induction in j, $\angle(x_1 + ix_2) = \angle(\xi_0 + i\eta_0) = \angle(\xi_v + i\eta_v) + \sum_{j=1}^{v} \theta_j z_j$. The final angle $\angle(\xi_v + i\eta_v)$ is between 0 and θ_v, so any feasible point (x_1, x_2) with an angle between $n\theta_v$ and $(n+1)\theta_v$ can only be represented by choosing the binary vector z to be the binary representation of n, as in the zig-zag formulations reference above.

Chapter 4

Computational Results

4.1 Integer Programming Relaxations of OPF

The difficulty of solving OPF to optimality resides in the non-convex constraints, be it the bilinear constraints of the rectangular formulation or the quadratic and arctangent constraints in the polar formulation. Bienstock and Muñoz [9] discretize the variables $\mathbf{e}_i$ or $\mathbf{f}_i$ to approximate the bilinear constraints, using Glover's construction [16] and we will do the same for the polar formulation.

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 $(c_{ij}, s_{ij}, z, \theta_j - \theta_i) \in H^4(0, 2\pi) = \{(x_3\cos(\alpha), x_3\sin(\alpha), x_3, \alpha) : x_3 \in \mathbb{R}^+, \alpha \in [-\pi, \pi]\}$. In PGLib instances, there are also bounds on $\theta_j - \theta_i$, so we only have to approximate the helix in a small interval $\alpha \in [\theta_l^{\Delta l}, \theta_l^{\Delta u}]$. Our proposed reformulation is as follows:

$$\min \sum_{i \in \mathcal{B}} C_i(\mathbf{p}_i^g)$$

subject to

$$\mathbf{p}_i^g - p_i^d = G_{ii}\mathbf{c}_{ii} + \sum_{j \in \delta(i)} [G_{ij}\mathbf{c}_{ij} - B_{ij}\mathbf{s}_{ij}] \qquad \forall i \in \mathcal{B} \tag{C.1}$$

$$\mathbf{q}_i^g - q_i^d = -B_{ii}\mathbf{c}_{ii} + \sum_{j \in \delta(i)} [-B_{ij}\mathbf{c}_{ij} - G_{ij}\mathbf{s}_{ij}] \qquad \forall i \in \mathcal{B} \tag{C.2}$$

$$\underline{V}_i^2 \leq \mathbf{c}_{ii} \leq \overline{V}_i^2 \qquad \forall i \in \mathcal{B} \tag{C.3}$$

$$\mathbf{c}_{ij} = \mathbf{c}_{ji}, \quad \mathbf{s}_{ij} = -\mathbf{s}_{ji} \qquad \forall (i,j) \in \mathcal{L} \tag{C.4}$$

$$(\mathbf{c}_{ij}, \mathbf{s}_{ij}, \mathbf{z}_{ij}, \boldsymbol{\theta_j} - \boldsymbol{\theta_i}) \in H^4,\ \ \theta_{ij}^{\Delta l} \leq \boldsymbol{\theta_j} - \boldsymbol{\theta_i} \leq \theta_{ij}^{\Delta u} \qquad \forall (i,j) \in \mathcal{L} \tag{C.5}$$

$$(2\mathbf{z}_{ij})^2 + (\mathbf{c}_i - \mathbf{c}_j)^2 = (\mathbf{c}_i + \mathbf{c}_j)^2 \qquad \forall (i,j) \in \mathcal{L} \tag{C.6}$$

$$p_i^{min} \leq \mathbf{p}_i^g \leq p_i^{max}, \quad q_i^{min} \leq \mathbf{q}_i^g \leq q_i^{max} \qquad \forall i \in \mathcal{B} \tag{C.7}$$

We can use the constructions in Chapter 3 to get relaxed approximations of the helix constraint (C.5) and the conical surface constraint (C.6), which produces a relaxed approximation to the overall problem. In total, we would need $2|\mathcal{L}|$ discretizations to solve the full problem, compared to the $|\mathcal{B}|$ discretizations proposed in [9], but electrical grid graphs are typically sparse, with $|\mathcal{L}| \leq 1.6|\mathcal{B}|$ for most networks in PGLib.

However, our formulation has the advantage that we can relax each of the constraints (C.6) and (C.5) for each branch independently. If we approximate the bilinear constraints using Glover's construction, but only discretize a subset of variables $\{\mathbf{e}_j : j \in J \subseteq \mathcal{B}\}$, there is no constraint on the products $\mathbf{e}_j\mathbf{e}_i$ or $\mathbf{e}_j\mathbf{f}_i$ for $j \notin J$ anymore. In contrast, we could start by solving the problem with (C.6) relaxed to SOC constraints, and only replace them by the conical surface relaxation if the solution is in the interior of the cone, which may happen only in a fraction of the constraints.

This also applies to the constraints (C.5) if the branch ij is a bridge of the graph $\mathcal{G}$, that is, if $\mathcal{G} - ij$ is not connected. Since angle constraints are only needed to ensure that the angles $\angle(\mathbf{c}_{ij} + i\mathbf{s}_{ij})$ add up to zero over cycles in $\mathcal{G}$ [28], the angle can always be recovered if an edge ij is a bridge and (C.5) can be replaced by another conical surface constraint.

In PGLib instances, the angles of the points $V_iV_j^* = c_{ij} + is_{ij})$ also have bounds, in particular $|\theta_i - \theta_j| \leq \frac{\pi}{6}$ (30 degrees) for the "typical operating conditions" and "congested operating conditions" cases, and less than 2 degrees for some of the "small angle difference" cases. This means that we only need to model a fraction of the domain of each helix. Finally, since $\mathbf{z}_{ij}$ is non-negative, the domain of (C.6) is also reduced to 180 degrees.

4.2 Results

We implement our formulation in Gurobi using the Python API for small PGLib cases, and run it for up to 20 minutes on a computer with an Intel Core i7 CPU, 4.8GHz processor, and 32 GB RAM. For comparison, we solve the same cases with the PowerModels implementation of the standard relaxations, using IPOPT as the solver for the full problem (used as the upper bound for all relaxations) as well as the SOC and QC relaxations, and MOSEK to solve the SDP relaxation. The full results are presented in Appendix A.

Tables 4.1, 4.2 and 4.3 show the optimality gaps of the MIP formulation with fixed approximation depth v set to 7, 8 and 9, meaning that all nonconvex constraints are approximated by MILP relaxations with v variables. All times are in seconds, unless otherwise specified. The gap is between the IPOPT solution for the full problem and each relaxation. If Gurobi does not terminate in 20 minutes, we use the best lower bound produced by branch and bound at that point, which is also a valid lower bound for the solution of the original problem. The best gap is highlighted in bold.

Case	SOC Gap	SDP Gap	SDP Time	Gap (7)	Time (7)	Gap (8)	Time (8)	Gap (9)	Time (9)
3_lmbd	1.32%	0.38%	0.01	0.02%	0.21	0.01%	0.21	**0.01%**	0.31
5_pjm	14.54%	5.22%	0.09	2.49%	1.11	1.15%	4.11	**0.69%**	13.02
14_ieee	0.11%	**0.00%**	0.15	0.01%	16.12	0.00%	60.55	0.00%	112.89
24_ieee_rts	0.01%	**0.00%**	0.36	0.02%	20 min	0.01%	20 min	0.01%	20 min
30_as	0.06%	**0.00%**	0.92	0.03%	84.36	0.00%	239.09	0.00%	806.42
30_ieee	18.84%	**0.01%**	0.71	0.36%	20 min	0.52%	20 min	0.57%	20 min
39_epri	0.55%	**0.01%**	2.93	0.25%	20 min	0.19%	20 min	0.24%	20 min
57_ieee	0.16%	**0.00%**	13.58	0.15%	20 min	0.15%	20 min	0.14%	20 min
60_c	0.06%	**0.02%**	21.91	0.04%	20 min	0.04%	20 min	0.04%	20 min
73_ieee_rts	0.03%	**0.00%**	60.15	0.04%	20 min	0.03%	20 min	0.03%	20 min
89_pegase	0.75%	1.83%	247.80	0.81%	20 min	0.76%	20 min	**0.75%**	20 min
118_ieee	0.90%	**0.07%**	928.54	0.81%	20 min	0.86%	20 min	0.89%	20 min

Table 4.1 : Time and gaps for PGLib instances under typical operating conditions

Case	SOC Gap	SDP Gap	SDP Time	Gap (7)	Time (7)	Gap (8)	Time (8)	Gap (9)	Time (9)
3_lmbd__api	9.32%	4.99%	0.01	0.05%	0.20	**0.02%**	0.30	0.06%	0.21
5_pjm__api	1.75%	0.20%	0.09	0.67%	0.61	0.46%	0.61	**0.18%**	1.11
14_ieee__api	5.13%	**0.02%**	0.17	0.47%	114.43	0.16%	142.72	0.10%	693.12
24_ieee_rts__api	7.48%	**0.31%**	0.32	6.34%	20 min	6.57%	20 min	4.21%	20 min
30_as__api	44.60%	**20.81%**	0.82	44.02%	20 min	44.39%	20 min	44.54%	20 min
30_ieee__api	5.43%	**0.56%**	0.89	0.66%	20 min	0.89%	20 min	0.84%	20 min
39_epri__api	1.42%	**0.38%**	2.51	0.65%	20 min	0.82%	20 min	0.77%	20 min
57_ieee__api	8.20%	**0.04%**	12.79	5.21%	20 min	5.33%	20 min	5.46%	20 min
60_c__api	2.06%	**1.30%**	17.90	1.86%	20 min	1.82%	20 min	1.84%	20 min
73_ieee_rts__api	4.20%	**0.56%**	84.84	3.06%	20 min	3.05%	20 min	4.19%	20 min
89_pegase__api	**12.50%**	12.95%	286.44	12.55%	20 min	12.51%	20 min	12.50%	20 min
118_ieee__api	26.16%	**10.36%**	968.40	26.10%	20 min	26.08%	20 min	26.15%	20 min

Table 4.2 : Time and gaps for PGLib instances under congested operating conditions

For small instances in which the MIP can be solved exactly, the optimality gap is reduced with every increase in binary variables because the feasible region is reduced with each increase in depth. However, in larger cases, adding more depth to the formulation can be counterproductive. We also improve some duality gaps produced by the SDP relaxation in very small cases, especially if those gaps are large. In contrast, for larger networks the MIP barely improves the gap of the SOC relaxation.

Despite achieving the lowest gaps overall, MOSEK is also more unstable than Gurobi or IPOPT and sometimes performs worse than the SOC relaxation (notably in case 89_pegase), despite being tighter in theory. It also fails to solve case 57_ieee__sad in table 4.3 entirely, and for bigger PGLib cases, it runs out of memory.

Case	SOC Gap	SDP Gap	SDP Time	Gap (7)	Time (7)	Gap (8)	Time (8)	Gap (9)	Time (9)
3_lmbd__sad	3.74%	0.62%	0.01	0.04%	0.20	0.01%	0.20	**0.01%**	0.20
5_pjm__sad	3.62%	**0.00%**	0.09	0.28%	0.21	0.56%	0.21	0.21%	0.21
14_ieee__sad	21.52%	**0.09%**	0.17	1.23%	16.73	0.54%	59.95	0.20%	173.35
24_ieee_rts__sad	9.54%	**2.53%**	0.38	4.69%	20 min	6.24%	20 min	5.25%	20 min
30_as__sad	7.87%	**0.18%**	0.68	0.33%	330.57	0.39%	20 min	0.40%	20 min
30_ieee__sad	9.70%	**0.02%**	0.79	0.26%	367.66	0.26%	20 min	0.43%	20 min
39_epri__sad	0.66%	0.04%	2.73	**0.03%**	644.21	0.15%	20 min	0.42%	20 min
57_ieee__sad	0.70%	99.96%	19.91	0.23%	20 min	0.22%	20 min	**0.21%**	20 min
60_c__sad	4.37%	**2.36%**	18.92	2.47%	20 min	2.46%	20 min	3.00%	20 min
73_ieee_rts__sad	6.73%	**1.48%**	57.00	4.34%	20 min	4.10%	20 min	4.65%	20 min
89_pegase__sad	0.72%	1.04%	254.26	0.80%	20 min	0.73%	20 min	**0.72%**	20 min
118_ieee__sad	8.16%	**1.84%**	1016.05	7.79%	20 min	7.90%	20 min	8.12%	20 min

Table 4.3 : Time and gaps for PGLib instances with small angle differences

Chapter 5

Conclusions and Future Work

Our constructions for approximating some families of nonconvex constraints reveal that there is still room to reduce the size of piecewise linear formulations, which are typically computed by storing each polyhedral piece separately in memory. For conical surfaces and trigonometric constraints, we also showed that an approximation error of at most ε can be achieved with $v = O(-\log(\varepsilon))$ integer variables, and that this bound is asymptotically optimal.

We also compared these geometric formulations with previously proposed programs based on combinatorial disjunctive constraints, showing a theoretical equivalence between their branching behaviors, and how different Gray codes in the latter can be emulated by choosing different "folding" maps in the former.

We showed that mixed integer programming relaxations using these constructions can be used to improve on the dual bounds than standard convex relaxations of ACOPF produce, at the expense of much worse scalability. However, we used only out-of-the-box solvers for comparison, when the state-of-the-art involves much faster cutting plane methods [10].

In future work, we should determine if $\varepsilon \leq O(4^{-v})$ is indeed the best possible asymptotic bound for trigonometric and other constraints, which seems likely. Approximating a one-variable function by its piecewise linear interpolation produces errors proportional to the square of the piece sizes, under mild assumptions such as a bounded second derivative [1]. Since we can split the domain into at most 2^v pieces with v disjunctions or binary variables, a piece size proportional to 2^{-v} would produce errors proportional to $O(4^{-v})$. However, the choice of pieces can reduce the error even further [20].

Another potential improvement could be to find constructions that approximate other convex constraints besides the second order cone, for which a linear program (without integer variables) could be enough. For example, the constructions in Chapter 3 could be modified to approximate $y \geq \exp(x)$ and $y \leq \sin(x), x \in [0, \pi]$ with similarly behaved errors. This would prove that problems with this type of constraints are also "polynomially reducible" to linear programs, the same as [8] does for Lorentz cones. In practice, however, these formulations are superseded by cutting-plane and interior point methods.

Finally, in order to ascertain how quickly MIP approximations can be competitive with convex relaxations for the ACOPF problem, we should pursue a more refined implementation of our formulation. Instead of using an all-purpose solver to execute branch and bound on a fixed MIP, it would be more efficient to start with convex relaxations of the conical surface constraints and branch only when the solution lies in the interior of those cones.

One way to accomplish this would be to find ways to warm-start our formulation, provided we have a solution for a lower depth v. Since our folding constructions can be truncated at any depth by adding the constraints (c), increasing the depth just amounts to adding additional variables ξ_j and η_j and modifying (c). A more ambitious approach could involve enforcing all convex inequalities through cutting planes, including planes derived from the SDP constraint on the matrix with entries $c_{ij} + is_{ij}$. Since execution times for the SDP relaxation have been significantly reduced by new approaches due to Lavaei [38], an MIP extension of them could reduce the gaps even more for small and medium cases.

In conclusion, mixed integer programming is a new approach to tighten the standard formulations of ACOPF, and at the moment they are only competitive against the standard techniques for the smallest instances. However, at the expense of significant computational resources, they can be used to reduce optimality gaps indefinitely. Future implementations, combined with other convex relaxations and potential improvements in integer programming solvers, could eventually place MIP relaxations among the standard techniques for this problem.

Appendix A

Complete numerical results

Case	IPOPT Value	IPOPT Time	QC Value	QC Time	SOC Value	SOC Time	SDP Value	SDP Time
3_lmbd	5812.64	0.00	5742.08	0.01	5736.17	0.00	5790.54	0.01
5_pjm	17551.89	0.11	14999.72	0.10	14999.71	0.09	16635.78	0.09
14_ieee	2178.08	0.21	2175.70	0.13	2175.70	0.11	2178.08	0.15
24_ieee_rts	63352.20	0.03	63344.58	0.06	63344.57	0.02	63352.18	0.36
30_as	803.13	0.01	802.68	0.04	802.68	0.02	803.13	0.92
30_ieee	8208.52	0.02	6665.22	0.05	6662.15	0.02	8207.79	0.71
39_epri	138415.56	0.03	137664.02	0.10	137654.06	0.03	138394.88	2.93
57_ieee	37589.34	0.03	37529.72	0.09	37529.71	0.04	37588.04	13.58
60_c	92693.67	0.16	92639.15	0.23	92637.04	0.13	92676.14	21.91
73_ieee_rts	189764.08	0.06	189706.17	0.16	189706.06	0.05	189761.68	60.15
89_pegase	107285.67	0.17	106484.57	0.45	106481.70	0.14	105322.43	247.80
118_ieee	97213.61	0.12	96450.42	0.28	96335.84	0.11	97141.28	928.54

Table A.1 : Time and objectives for IPOPT and MOSEK and typical operating conditions

Case	Value (6)	Time (6)	Value (7)	Time (7)	Value (8)	Time (8)	Value (9)	Time (9)
3_lmbd	5804.74	0.10	5809.96	0.20	5811.72	0.21	5812.19	0.31
5_pjm	16446.05	0.21	16645.38	0.31	17113.50	1.21	17350.26	5.11
14_ieee	2168.47	4.21	2174.99	6.41	2177.48	7.91	2177.95	27.03
24_ieee_rts	63098.62	22.82	63300.42	32.43	63342.30	110.79	63342.71	20 min
30_as	786.22	15.82	800.24	118.19	802.47	178.33	802.98	683.31
30_ieee	7965.73	24.33	8150.82	271.72	8180.87	20 min	8153.68	20 min
39_epri	137916.48	117.99	138309.87	815.93	138037.38	20 min	138181.13	20 min
57_ieee	37109.61	20 min	37443.21	20 min	37514.45	20 min	37534.96	20 min
60_c	92581.32	20 min	92645.55	20 min	92656.65	20 min	92656.82	20 min
73_ieee_rts	188946.83	20 min	189541.05	20 min	189676.08	20 min	189698.93	20 min
89_pegase	102304.86	20 min	105435.13	20 min	106249.71	20 min	106434.17	20 min
118_ieee	95555.36	20 min	96229.39	20 min	96425.43	20 min	96341.90	20 min

Table A.2 : Time and objectives for Gurobi and typical operating conditions

Case	IPOPT Value	IPOPT Time	QC Value	QC Time	SOC Value	SOC Time	SDP Value	SDP Time
3_lmbd__api	11242.12	0.01	10609.23	0.01	10194.91	0.00	10680.90	0.01
5_pjm__api	78949.91	0.11	77571.36	0.11	77571.36	0.08	78790.09	0.09
14_ieee__api	5999.36	0.17	5691.80	0.16	5691.80	0.10	5998.11	0.17
24_ieee_rts__api	161222.58	0.03	150015.21	0.06	149168.25	0.02	160717.11	0.32
30_as__api	4996.20	0.03	2767.85	0.05	2767.85	0.02	3956.62	0.82
30_ieee__api	18036.59	0.02	17057.85	0.04	17057.84	0.02	17936.39	0.89
39_epri__api	256769.34	0.03	253220.36	0.09	253130.37	0.05	255791.42	2.51
57_ieee__api	36242.46	0.03	33370.20	0.09	33271.61	0.04	36228.76	12.79
60_c__api	185002.89	0.18	181212.97	0.25	181189.17	0.14	182596.49	17.90
73_ieee_rts__api	509847.89	0.10	490117.76	0.22	488422.63	0.06	507015.71	84.84
89_pegase__api	129568.36	0.23	113412.21	0.42	113369.28	0.15	112789.71	286.44
118_ieee__api	249614.52	0.14	184543.41	0.30	184307.64	0.09	223750.72	968.40

Table A.3 : Time and objectives for IPOPT and MOSEK and congested operating conditions

Case	Value (6)	Time (6)	Value (7)	Time (7)	Value (8)	Time (8)	Value (9)	Time (9)
3_lmbd__api	11189.62	0.10	11230.43	0.20	11235.75	0.21	11238.61	0.21
5_pjm__api	77490.16	0.11	77939.35	0.21	78394.64	0.31	78515.24	0.51
14_ieee__api	5872.39	4.11	5967.92	10.11	5969.54	35.13	5989.25	199.67
24_ieee_rts__api	156622.85	20 min	154811.91	20 min	149990.09	20 min	151853.43	20 min
30_as__api	2893.20	20 min	2796.79	20 min	2793.65	20 min	2780.43	20 min
30_ieee__api	17712.59	16.32	17860.54	81.07	17896.33	20 min	17906.00	20 min
39_epri__api	255464.27	330.75	255282.55	20 min	255039.53	20 min	254836.23	20 min
57_ieee__api	34301.83	20 min	34546.71	20 min	34473.65	20 min	34403.35	20 min
60_c__api	181439.11	20 min	181632.81	20 min	181617.50	20 min	181643.22	20 min
73_ieee_rts__api	496473.16	20 min	497894.74	20 min	494585.39	20 min	488719.93	20 min
89_pegase__api	108869.96	20 min	112333.01	20 min	113132.02	20 min	113316.74	20 min
118_ieee__api	183227.85	20 min	184087.78	20 min	184364.71	20 min	184460.82	20 min

Table A.4 : Time and objectives for Gurobi and congested operating conditions

Case	IPOPT Value	IPOPT Time	QC Value	QC Time	SOC Value	SOC Time	SDP Value	SDP Time
3_lmbd__sad	5959.31	0.00	5874.77	0.01	5736.17	0.00	5922.14	0.01
5_pjm__sad	26108.84	0.11	25851.04	0.10	25164.93	0.09	26108.85	0.09
14_ieee__sad	2776.79	0.18	2180.37	0.14	2179.18	0.14	2774.22	0.17
24_ieee_rts__sad	76917.96	0.03	74667.00	0.08	69578.86	0.02	74970.12	0.38
30_as__sad	897.35	0.02	876.64	0.06	826.73	0.02	895.74	0.68
30_ieee__sad	8208.52	0.02	7721.70	0.06	7412.59	0.02	8206.99	0.79
39_epri__sad	148340.50	0.03	148030.68	0.09	147360.52	0.03	148275.70	2.73
57_ieee__sad	38663.28	0.03	38529.48	0.11	38392.20	0.04	14.48	19.91
60_c__sad	113498.76	0.16	110916.39	0.25	108541.85	0.16	110821.91	18.92
73_ieee_rts__sad	227603.74	0.09	221836.16	0.23	212294.72	0.06	224240.54	57.00
89_pegase__sad	107285.67	0.16	106531.21	0.40	106510.31	0.14	106173.95	254.26
118_ieee__sad	105155.05	0.11	98017.80	0.34	96571.52	0.09	103222.74	1016.05

Table A.5 : Time and objectives for IPOPT and MOSEK and small angle differences

Case	Value (6)	Time (6)	Value (7)	Time (7)	Value (8)	Time (8)	Value (9)	Time (9)
3_lmbd__sad	5951.06	0.10	5956.71	0.20	5958.82	0.21	5958.91	0.21
5_pjm__sad	26046.29	0.21	26086.15	0.21	26095.25	0.31	26102.81	0.51
14_ieee__sad	2710.31	8.11	2752.92	26.93	2769.34	65.65	2774.91	202.77
24_ieee_rts__sad	73177.87	20 min	72219.98	20 min	71916.09	20 min	72491.17	20 min
30_as__sad	877.74	20 min	890.90	20 min	892.53	20 min	892.98	20 min
30_ieee__sad	8006.91	354.46	8158.48	20 min	8165.03	20 min	8161.71	20 min
39_epri__sad	147364.85	20 min	148094.35	20 min	147869.58	20 min	148075.63	20 min
57_ieee__sad	38171.07	20 min	38498.70	20 min	38562.43	20 min	38572.29	20 min
60_c__sad	111222.02	20 min	110867.25	20 min	110095.12	20 min	110260.01	20 min
73_ieee_rts__sad	218098.12	20 min	218339.08	20 min	217576.63	20 min	217484.41	20 min
89_pegase__sad	102608.83	20 min	105532.13	20 min	106299.91	20 min	106468.38	20 min
118_ieee__sad	96027.33	20 min	96756.53	20 min	96966.58	20 min	96627.01	20 min

Table A.6 : Time and objectives for Gurobi and small angle differences

Bibliography

[1] Kendall Atkinson. “Interpolation Theory”. In: *An Introduction to Numerical Analysis*. John Wiley, 1991, pp. 131–196. ISBN: 978-0-471-62489-9.

[2] Sogol Babaeinejadsarookolaee et al. “The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms”. In: *ArXiv* abs/1908.02788 (2019). URL: https://api.semanticscholar.org/CorpusID:199501662.

[3] Shahab Bahrami et al. “Semidefinite Relaxation of Optimal Power Flow for AC–DC Grids”. In: *IEEE Transactions on Power Systems* 32.1 (2017), pp. 289–304. DOI: 10.1109/TPWRS.2016.2543726.

[4] Xiaoqing Bai et al. “Semidefinite programming for optimal power flow problems”. In: *International Journal of Electrical Power Energy Systems* 30.6 (2008), pp. 383–392. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2007.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0142061507001378.

[5] Xiaowei Bao, Nikolaos V. Sahinidis, and Mohit Tawarmalani. “Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs”. In: *Optimization Methods and Software* 24.4-5 (2009), pp. 485–504. DOI: 10.1080/10556780902883184. URL: https://doi.org/10.1080/10556780902883184.

[6] Benjamin Beach, Robert Hildebrand, and Joey Huchette. “Compact mixed-integer programming formulations in quadratic optimization”. In: *Journal of*

Global Optimization 4.84 (2022), pp. 869–912. DOI: 10.1007/s10898-022-01184-6.

[7] Pietro Belotti et al. “Disjunctive Inequalities: Applications And Extensions”. In: *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley Sons, Ltd, 2011. ISBN: 9780470400531. DOI: https://doi.org/10.1002/9780470400531.eorms0537. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470400531.eorms0537.

[8] Aharon Ben-Tal and Arkadi Nemirovski. “On Polyhedral Approximations of the Second-Order Cone”. In: *Mathematics of Operations Research* 26.2 (2001), pp. 195–205.

[9] Daniel Bienstock and Gonzalo Muñoz. “On linear relaxations of OPF problems”. In: *arXiv: Optimization and Control* (2014). URL: https://api.semanticscholar.org/CorpusID:117479734.

[10] Daniel Bienstock and Matías Villagra. “Accurate and Warm-Startable Linear Cutting-Plane Relaxations for ACOPF”. In: *2024 IEEE 63rd Conference on Decision and Control (CDC)*. 2024, pp. 5024–5031. DOI: 10.1109/CDC56724.2024.10886304.

[11] Mary B. Cain, Richard P. O’Neill, and Anya Castillo. “History of optimal power flow and formulations”. In: *Fed. Energy Regul. Comm.* 1 (Jan. 2012), pp. 1–36.

[12] Carleton Coffrin, Dan Gordon, and Paul Scott. “NESTA, The NICTA Energy System Test Case Archive”. In: *ArXiv* abs/1411.0359 (2014). URL: https://api.semanticscholar.org/CorpusID:8719091.

[13] Carleton Coffrin, Hassan L. Hijazi, and Pascal Van Hentenryck. "The QC Relaxation: A Theoretical and Computational Study on Optimal Power Flow". In: *IEEE Transactions on Power Systems* 31.4 (2016), pp. 3008–3018. DOI: 10.1109/TPWRS.2015.2463111.

[14] Michele Conforti, Gerard Cornuejols, and Giacomo Zambelli. "Linear Inequalities and Polyhedra". In: *Integer Programming*. Cham: Springer International Publishing, 2014, pp. 85–128. ISBN: 978-3-319-11008-0. DOI: 10.1007/978-3-319-11008-0_3. URL: https://doi.org/10.1007/978-3-319-11008-0_3.

[15] William J. Cook et al. *Combinatorial Optimization*. Series in Discrete Mathematics and Optimization. Wiley-Interscience, 1998. ISBN: 047155894X.

[16] Fred Glover. "Improved linear integer programming formulations of nonlinear integer problems". In: *Management science* 22.4 (1975), pp. 455–460.

[17] Antonio Gómez Expósito and Esther Romero Ramos. "Reliable load flow technique for radial distribution networks". In: *IEEE Transactions on Power Systems* 14.3 (1999), pp. 1063–1069. DOI: 10.1109/59.780924.

[18] Brais González-Rodríguez and Joe Naoum-Sawaya. "Degree reduction techniques for polynomial optimization problems". In: *European Journal of Operational Research* 322.2 (2025), pp. 401–413. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2024.12.021. URL: https://www.sciencedirect.com/science/article/pii/S0377221724009639.

[19] Akshay Gupte et al. "Solving Mixed Integer Bilinear Problems Using MILP Formulations". In: *SIAM Journal on Optimization* 23.2 (Jan. 2013), pp. 721–744. ISSN: 1052-6234. DOI: 10.1137/110836183. URL: https://doi.org/10.1137/110836183.

[20] Bernd Hamann and Jiann-Liang Chen. “Data point selection for piecewise linear curve approximation”. In: *Computer Aided Geometric Design* 11.3 (1994), pp. 289–301.

[21] Joey Huchette and Juan Pablo Vielma. “A Combinatorial Approach for Small and Strong Formulations of Disjunctive Constraints”. In: *Mathematics of Operations Research* 44(3) (2019), pp. 793–820.

[22] Joey Huchette and Juan Pablo Vielma. “A geometric way to build strong mixed-integer programming formulations”. In: *Operations Research Letters* 47.6 (2019), pp. 601–606.

[23] Joey Huchette and Juan Pablo Vielma. “Nonconvex Piecewise Linear Functions: Advanced Formulations and Simple Modeling Tools”. In: *Operations Research* 71.5 (2022), pp. 1835–1856. DOI: 10.1287/opre.2019.1973. URL: https://doi.org/10.1287/opre.2019.1973.

[24] Rabih A. Jabr. “A Conic Quadratic Format for the Load Flow Equations of Meshed Networks”. In: *IEEE Transactions on Power Systems* 22.4 (2007), pp. 2285–2286. DOI: 10.1109/TPWRS.2007.907590.

[25] Robert G. Jeroslow. “Representability in mixed integer programmiing, I: Characterization results”. In: *Discrete Applied Mathematics* 17.3 (1987), pp. 223–243. ISSN: 0166-218X. DOI: https://doi.org/10.1016/0166-218X(87)90026-6. URL: https://www.sciencedirect.com/science/article/pii/0166218X87900266.

[26] Burak Kocuk, Santanu S. Dey, and X. Andy Sun. “Inexactness of SDP Relaxation and Valid Inequalities for Optimal Power Flow”. In: *IEEE Transactions on Power Systems* 31.1 (2016), pp. 642–651. DOI: 10.1109/TPWRS.2015.2402640.

[27] Burak Kocuk, Santanu S. Dey, and X. Andy Sun. “Matrix minor reformulation and SOCP-based spatial branch-and-cut method for the AC optimal power flow problem”. In: *Mathematical Programming Computation* 10.4 (2018), pp. 557–596.

[28] Burak Kocuk, Santanu S. Dey, and X. Andy Sun. “Strong SOCP Relaxations for the Optimal Power Flow Problem”. In: *Operations Research* 64.6 (2016), pp. 1177–1196. DOI: https://doi.org/10.1287/opre.2016.1489.

[29] Javad Lavaei and Steven H. Low. “Zero Duality Gap in Optimal Power Flow Problem”. In: *IEEE Transactions on Power Systems* 27.1 (2012), pp. 92–107. DOI: 10.1109/TPWRS.2011.2160974.

[30] Karsten Lehmann, Alban Grastien, and Pascal Van Hentenryck. “AC-feasibility on tree networks is NP-hard”. In: *IEEE Transactions on Power Systems* 31.1 (2015), pp. 798–801.

[31] Bochuan Lyu, Illya V. Hicks, and Joey Huchette. “Modeling combinatorial disjunctive constraints via junction trees”. In: *Mathematical Programming* 204 (2024), pp. 385–413. DOI: https://doi.org/10.1007/s10107-023-01955-3.

[32] Anatoliy D. Rikun. “A Convex Envelope Formula for Multilinear Functions”. In: *Journal of Global Optimization* 10.4 (1997), pp. 425–437. ISSN: 0925-5001. DOI: 10.1023/A:1008217604285. URL: https://doi.org/10.1023/A:1008217604285.

[33] Kaarthik Sundar, Sujeevraja Sanjeevi, and Harsha Nagarajan. “Sequence of polyhedral relaxations for nonlinear univariate functions”. In: *Optimization and Engineering* 23 (Apr. 2021), pp. 877–894. DOI: 10.1007/s11081-021-09609-z.

[34] Mohit Tawarmalani, Jean-Phillipe P. Richard, and Chuanhui Xiong. "Explicit convex and concave envelopes through polyhedral subdivisions". In: *Mathematical Programming* 138 (2013), pp. 531–577. DOI: `10.1007/s10107-012-0581-4`.

[35] Juan Pablo Vielma, Shabbir Ahmed, and George Nemhauser. "Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions". In: *Operations Research* 2.58 (2009), pp. 303–315.

[36] Juan Pablo Vielma and George L. Nemhauser. "Modeling disjunctive constraints with a logarithmic number of binary variables and constraints". In: *Mathematical Programming* 128 (2011), pp. 49–72. DOI: `10.1007/s10107-009-0295-4`. URL: `https://doi.org/10.1007/s10107-009-0295-4`.

[37] Andreas Wächter and Lorenz Biegler. "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming". In: *Mathematical programming* 106 (Mar. 2006), pp. 25–57. DOI: `10.1007/s10107-004-0559-y`.

[38] Richard Y. Zhang and Javad Lavaei. "Sparse semidefinite programs with guaranteed near-linear time complexity via dualized clique tree conversion". In: *Mathematical programming* 188 (2021), pp. 351–393.

[39] Ray Zimmerman, Carlos Murillo-Sanchez, and Defang Gan. "MATPOWER-A MATLAB Power System Simulation Package: User"s Manual". In: (Dec. 1997).