Part II: Fundamentals and New Algorithms

Daniel Bienstock, Columbia University

Organization of this part of the tutorial:

- Intro to OPF and basic modern mathematics related to OPF
- Newer results, including basic methodologies for polynomial optimization
- Slower introduction to underlying mathematics (runs past end of webinar)

Remarks

- One hour is not enough!
- In most cases we present an outline of results with some detail removed.
- This is an enhanced version of a webinar I gave a year ago

INTRO

Power flow problem in its simplest form

Power flow problem in its simplest form

Parameters:

- For each line $\boldsymbol{km}$, its admittance $\boldsymbol{b_{km} + j g_{km} = b_{mk} + j g_{mk}}$

- For each bus $\boldsymbol{k}$, voltage limits $\boldsymbol{V_k^{\min}}$ and $\boldsymbol{V_k^{\max}}$

- For each bus $\boldsymbol{k}$, active and reactive net power limits

  $\boldsymbol{P_k^{\min}}$, $\boldsymbol{P_k^{\max}}$, $\boldsymbol{Q_k^{\min}}$, and $\boldsymbol{Q_k^{\max}}$

Variables to compute:

- For each bus $\boldsymbol{k}$, complex voltage $\boldsymbol{e_k + j f_k}$

- Or, complex voltage $\boldsymbol{|V_k|\, e^{j\theta_k}}$

Notation: For a bus $\boldsymbol{k}$, $\boldsymbol{\delta(k)}$ = set of lines incident with $\boldsymbol{k}$

Basic power flow problem

Find a solution to:

$$\boldsymbol{P_k^{\min}} \;\leq\; \sum_{km \in \delta(k)} \left[\, \boldsymbol{g_{km}}(e_k^2 + f_k^2) - \boldsymbol{g_{km}}(e_k e_m + f_k f_m) + \boldsymbol{b_{km}}(e_k f_m - f_k e_m) \right] \;\leq\; \boldsymbol{P_k^{\max}}$$

$$\boldsymbol{Q_k^{\min}} \;\leq\; \sum_{km \in \delta(k)} \left[\, -\boldsymbol{b_{km}}(e_k^2 + f_k^2) + \boldsymbol{b_{km}}(e_k e_m + f_k f_m) + \boldsymbol{g_{km}}(e_k f_m - f_k e_m) \right] \;\leq\; \boldsymbol{Q_k^{\max}}$$

$$\boldsymbol{(V_k^{\min})^2} \;\leq\; e_k^2 + f_k^2 \;\leq\; \boldsymbol{(V_k^{\max})^2},$$

for each bus $\boldsymbol{k = 1, 2, \ldots}$

Many possible variations/extensions, plus optimization versions

Notation: For a bus $\boldsymbol{k}$, $\boldsymbol{\delta(k)}$ = set of lines incident with $\boldsymbol{k}$

In polar representation

Find a solution to:

$$\boldsymbol{P_k^{\min}} \leq \sum_{km \in \delta(k)} \left[\boldsymbol{g_{km}}(|V_k|^2) - \boldsymbol{g_{km}}|V_k||V_m|\cos(\theta_k - \theta_m) - \boldsymbol{b_{km}}|V_k||V_m|\sin(\theta_k - \theta_m)\right] \leq \boldsymbol{P_k^{\max}}$$

$$\boldsymbol{Q_k^{\min}} \leq \sum_{km \in \delta(k)} \left[-\boldsymbol{b_{km}}(|V_k|^2) + \boldsymbol{b_{km}}|V_k||V_m|\cos(\theta_k - \theta_m) - \boldsymbol{g_{km}}|V_k||V_m|\sin(\theta_k - \theta_m)\right] \leq \boldsymbol{Q_k^{\max}}$$

$$\boldsymbol{(V_k^{\min})^2} \leq |V_k|^2 \leq \boldsymbol{(V_k^{\max})^2},$$

for each bus $\boldsymbol{k = 1, 2, \ldots}$

Quadratically constrained, quadratic programming problems (QCQPs)

$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_i(x) \ \geq \ 0, \quad 1 \leq i \leq m \\ & x \in \mathbb{R}^n \end{array}$$

Here,

$$f_i(x) \ = \ x^T M_i x + c_i^T x + d_i$$

is a general quadratic (each M_i is $n \times n$, wlog symmetric)

Folklore result: QCQP is NP-hard

... and in practice QCQP can be quite hard

Folklore result: QCQP is NP-hard

Let $w_1, w_2, \ldots, w_n$ be **integers**, and consider:

$$
\begin{aligned}
W^* \doteq \min \quad & -\sum_i x_i^2 \\
\text{s.t.} \quad & \sum_i w_i \, x_i = 0, \\
& -1 \le x_i \le 1, \quad 1 \le i \le n.
\end{aligned}
$$

Folklore result: QCQP is NP-hard

Let $w_1, w_2, \ldots, w_n$ be **integers**, and consider:

$$\begin{aligned} W^* \doteq \min \ & -\sum_i x_i^2 \\ \text{s.t.} \ & \sum_i w_i x_i = 0, \\ & -1 \leq x_i \leq 1, \ \ 1 \leq i \leq n. \end{aligned}$$

we have that $\boldsymbol{W^* = -n}$, iff there exists a subset $J \subseteq \{1, \ldots, n\}$ with

$$\sum_{j \in J} w_j = \sum_{j \notin J} w_j$$

Folklore result: QCQP is NP-hard

Let $w_1, w_2, \ldots, w_n$ be **integers**, and consider:

$$\begin{aligned} W^* \doteq \min \; & -\sum_i x_i^2 \\ \text{s.t.} \; & \sum_i w_i \, x_i = 0, \\ & -1 \le x_i \le 1, \quad 1 \le i \le n. \end{aligned}$$

we have that $\boldsymbol{W^* = -n}$, iff there exists a subset $J \subseteq \{1, \ldots, n\}$ with

$$\sum_{j \in J} w_j = \sum_{j \notin J} w_j$$

This is called the "integer partition" (or "subset sum") problem.

It is **NP-hard** when the w_i are large. It is, thus, **weakly** NP-hard.

Folklore result: QCQP is NP-hard

Let $w_1, w_2, \ldots, w_n$ be **integers**, and consider:

$$\begin{aligned} W^* \doteq \min \; & -\sum_i x_i^2 \\ \text{s.t.} \; & \sum_i w_i \, x_i = 0, \\ & -1 \le x_i \le 1, \quad 1 \le i \le n. \end{aligned}$$

we have that $\boldsymbol{W^* = -n}$, iff there exists a subset $J \subseteq \{1, \ldots, n\}$ with

$$\sum_{j \in J} w_j = \sum_{j \notin J} w_j$$

This is called the "integer partition" (or "subset sum") problem.

It is **NP-hard** when the w_i are large. It is, thus, **weakly** NP-hard. But can be **approximately** solved.

Take any $\{-1,1\}$-linear program

$$\begin{aligned} \min\ & c^T x \\ \text{s.t.}\ & Ax = b \\ & x \in \{-1,1\}^n. \end{aligned}$$

Take any $\{-1, 1\}$-linear program

$$\min \ c^T x$$

$$\text{s.t.} \quad Ax = b$$

$$x \in \{-1, 1\}^n.$$

this is the same as, for **big M**,

$$\min \ c^T x \ - \ \boldsymbol{M} \sum_j x_j^2$$

$$\text{s.t.} \quad Ax = b$$

$$-1 \leq x_j \leq 1, \quad 1 \leq j \leq n.$$

Take any $\{-1,1\}$-linear program

$$\min \; c^T x$$

$$\text{s.t.} \quad Ax = b \quad \text{and} \quad x \in \{-1,1\}^n.$$

this is the same as, for **big M**,

$$\min \; c^T x \; - \; \boldsymbol{M} \sum_j x_j^2$$

$$\text{s.t.} \quad Ax = b$$

$$-1 \le x_j \le 1, \quad 1 \le j \le n.$$

so **linearly constrained** QCQP is as hard as general integer programming

And how about AC-OPF – a special case of QCQP?

- Lavaei & Low (2011), van Hentenryck & Coffrin (2014): AC-OPF is weakly NP-hard on trees

- Bienstock and Verma (2008): AC-OPF is strongly NP-hard on general networks

- Bienstock and Muñoz (2014): AC-OPF can be ***approximated*** on trees, and more generally on networks of small "tree-width"

Special case of AC power flows

- Zero resistances ("lossless")
- Unit voltage magnitues
- No constraints on reactive power flows
- Active power flow on line $\boldsymbol{km}$:

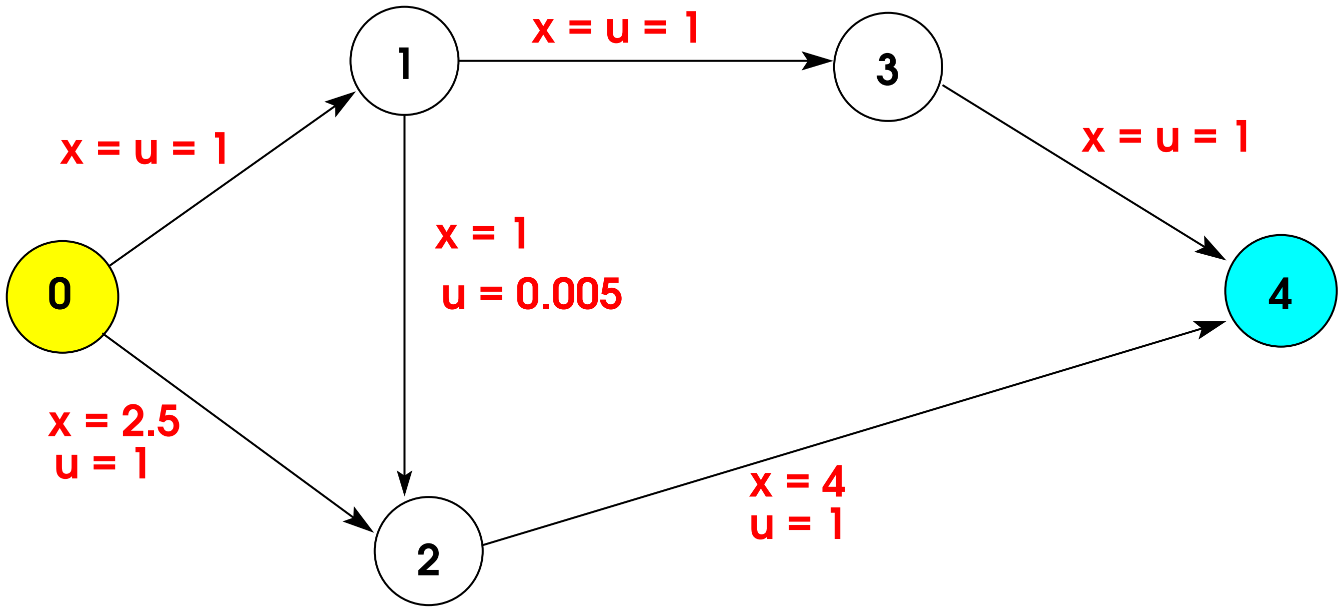
$$\sin(\theta_k - \theta_m)/x_{km}$$

- Line limits

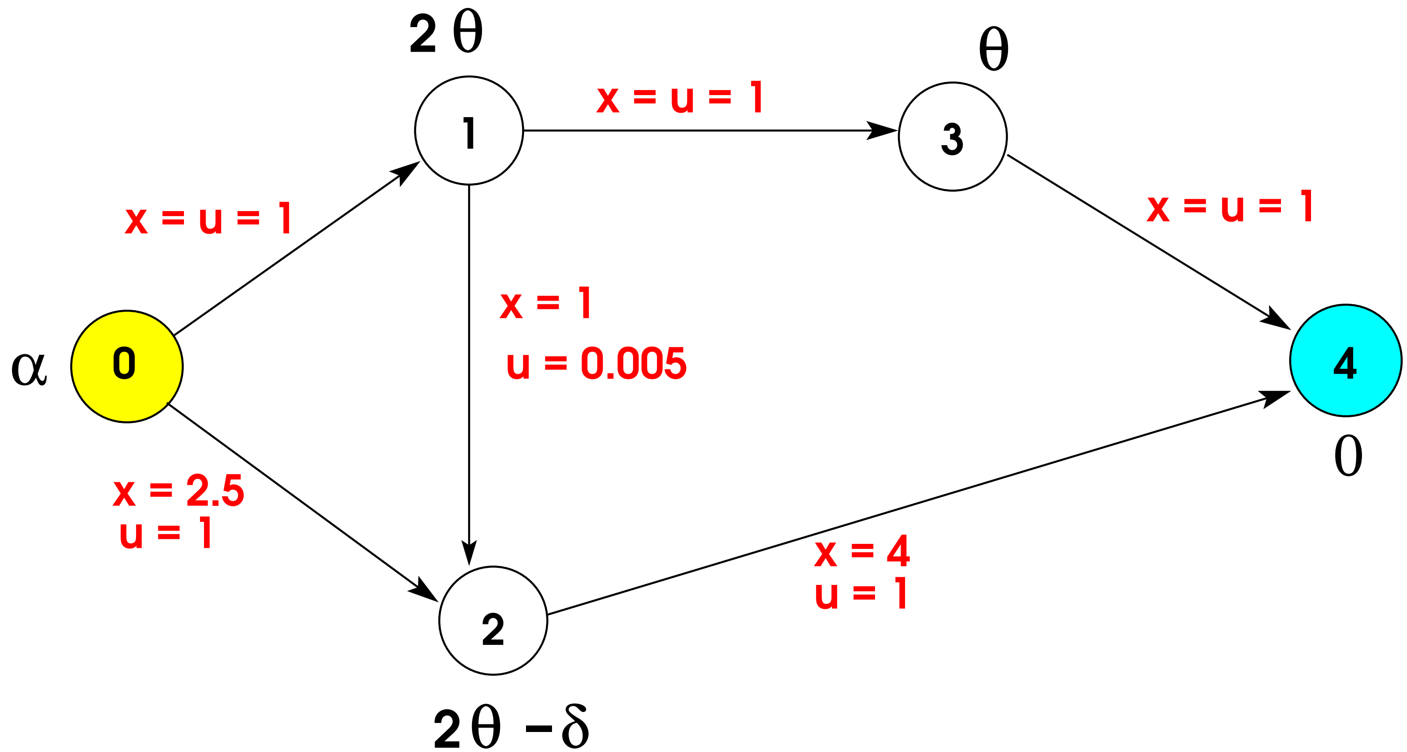
$$|\sin(\theta_k - \theta_m)|/x_{km} \ \leq \ u_{km}$$

- Choose phase angles so that flow balance is attained, subject to line limits

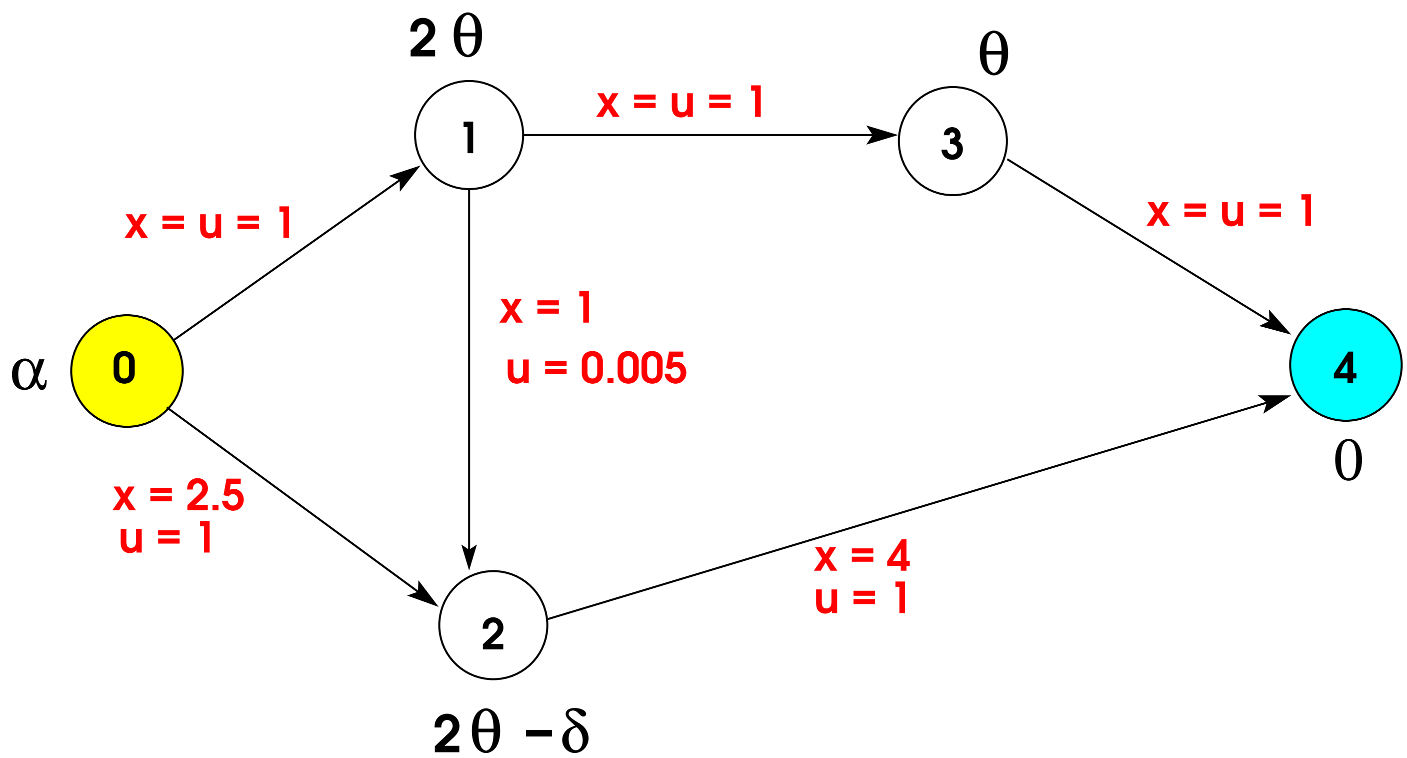
- Bus **0** is a generator
- Bus **4** is a load



- Bus **0** is a generator
- Bus **4** is a load



- Bus $\mathbf{0}$ is a generator
- Bus $\mathbf{4}$ is a load



- $\boldsymbol{\delta} \geq \mathbf{0}$ and very small. Assume $= \mathbf{0}$.
- Flow conservation at node $\mathbf{1}$: $\mathbf{\sin(\boldsymbol{\alpha} - 2\boldsymbol{\theta}) = \sin(\boldsymbol{\theta})}$
- Flow conservation at node $\mathbf{2}$: $\mathbf{\sin(\boldsymbol{\alpha} - 2\boldsymbol{\theta}) = \frac{5}{8}\sin(2\boldsymbol{\theta})}$
- So $\mathbf{\sin(\boldsymbol{\theta}) = \frac{5}{8}\sin(2\boldsymbol{\theta})}$. So $\boldsymbol{\theta} = \mathbf{0}$ or $\boldsymbol{\theta} = \mathbf{\cos^{-1}(4/5)}$

Even more general than QCQP:

Polynomially-constrained problems.

Problem: given polynomials $p_i : \mathbb{R}^n \to \mathbb{R}$, for $1 \leq i \leq m$
find $x \in \mathbb{R}^n$ s.t. $p_i(x) = 0, \forall\, i$

Observation. Can be reduced to QCQP.

Example: find a solution for $3v^6w - v^4 + 7 = 0$.

Even more general than QCQP:

Polynomially-constrained problems.

Problem: given polynomials $p_i : \mathbb{R}^n \to \mathbb{R}$, for $1 \leq i \leq m$
find $x \in \mathbb{R}^n$ s.t. $p_i(x) = 0, \forall\, i$

Observation. Can be reduced to QCQP.

Example: find a root for $3v^6w - v^4 + 7 = 0$.

Equivalent to the system on variables v, v_2, v_4, v_6, w, y and c:

$$\begin{aligned} c^2 &= 1 \\ v^2 - cv_2 &= 0 \\ v_2^2 - cv_4 &= 0 \\ v_2v_4 - cv_6 &= 0 \\ v_6w - cy &= 0 \\ 3cy - cv_4 &= -7 \end{aligned}$$

This is an "efficient" (polynomial-time) reduction

Back to general QCQP

Back to general QCQP

$$\begin{array}{rll} \textbf{(QCQP):} & \min\ x^T Q x + 2c^T x & \\ \text{s.t.} & x^T A_i x + 2b_i^T x + r_i \geq 0 & i = 1, \ldots, m \\ & x \in \mathbb{R}^n. & \end{array}$$

Back to general QCQP

$$
\textbf{(QCQP):} \quad \begin{array}{ll} \min & x^T Q x \, + \, 2c^T x \\ \text{s.t.} & x^T A_i x \, + \, 2b_i^T x \, + \, r_i \; \geq 0 \qquad i = 1, \ldots, m \\ & x \in \mathbb{R}^n. \end{array}
$$

$\rightarrow$ form the **semidefinite relaxation**

$$
\textbf{(SR):} \quad \begin{array}{ll} \min & \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix} \bullet X \\ \text{s.t.} & \begin{pmatrix} r_i & b_i^T \\ b_i & A^i \end{pmatrix} \bullet X \; \geq 0 \qquad i = 1, \ldots, m \\ & X \succeq 0, \quad X_{11} = 1. \end{array}
$$

Here, for symmetric matrices $\boldsymbol{M}$, $\boldsymbol{N}$,

$$
M \bullet N \; = \; \sum_{h,k} M_{hk} N_{hk}
$$

Back to general QCQP

$$
\begin{array}{rll}
\textbf{(QCQP):} & \min\ x^T Q x \,+\, 2c^T x & \\
\text{s.t.} & x^T A_i x \,+\, 2b_i^T x \,+\, r_i \ \geq 0 & \quad i = 1, \ldots, m \\
 & x \in \mathbb{R}^n. &
\end{array}
$$

$\rightarrow$ form the **semidefinite relaxation**

$$
\begin{array}{rll}
\textbf{(SR):} & \min \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix} \bullet X & \\
\text{s.t.} & \begin{pmatrix} r_i & b_i^T \\ b_i & A^i \end{pmatrix} \bullet X \ \geq 0 & \quad i = 1, \ldots, m \\
 & X \succeq 0, \quad X_{11} = 1. &
\end{array}
$$

Here, for symmetric matrices $\boldsymbol{M}$, $\boldsymbol{N}$,

$$M \bullet N \ = \ \sum_{h,k} M_{hk} N_{hk}$$

Why do we call it a relaxation?

Back to general QCQP

$$
\begin{array}{rll}
\textbf{(QCQP):} & \min\ x^TQx + 2c^Tx & \\
\text{s.t.} & x^TA_ix + 2b_i^Tx + r_i \ \geq 0 & i = 1,\ldots,m \\
 & x \in \mathbb{R}^n. &
\end{array}
$$

$\rightarrow$ form the **semidefinite relaxation**

$$
\begin{array}{rll}
\textbf{(SR):} & \min \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix} \bullet X & \\
\text{s.t.} & \begin{pmatrix} r_i & b_i^T \\ b_i & A^i \end{pmatrix} \bullet X \ \geq 0 & i = 1,\ldots,m \\
 & X \succeq 0, \quad X_{11} = 1. &
\end{array}
$$

Here, for symmetric matrices $\boldsymbol{M}$, $\boldsymbol{N}$,

$$
M \bullet N \ = \ \sum_{h,k} M_{hk}N_{hk}
$$

Why do we call it a relaxation?

Given $\boldsymbol{x}$ feasible for **QCQP**, the matrix $\begin{pmatrix} \mathbf{1} \\ \boldsymbol{x} \end{pmatrix} (\mathbf{1}, \boldsymbol{x}^T)$ feasible for **SR** and with the same value

Back to general QCQP

$$\textbf{(QCQP):} \quad \begin{array}{ll} \min & x^T Q x + 2c^T x \\ \text{s.t.} & x^T A_i x + 2b_i^T x + r_i \geq 0 \qquad i = 1, \ldots, m \\ & x \in \mathbb{R}^n. \end{array}$$

$\longrightarrow$ form the **semidefinite relaxation**

$$\textbf{(SR):} \quad \begin{array}{ll} \min & \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix} \bullet X \\ \text{s.t.} & \begin{pmatrix} r_i & b_i^T \\ b_i & A^i \end{pmatrix} \bullet X \geq 0 \qquad i = 1, \ldots, m \\ & X \succeq 0, \quad X_{11} = 1. \end{array}$$

Here, for symmetric matrices $\boldsymbol{M}$, $\boldsymbol{N}$,

$$M \bullet N = \sum_{h,k} M_{hk} N_{hk}$$

Why do we call it a relaxation?

Given $\boldsymbol{x}$ feasible for **QCQP**, the matrix $\begin{pmatrix} \mathbf{1} \\ \boldsymbol{x} \end{pmatrix} (\mathbf{1}, \boldsymbol{x}^T)$ feasible for **SR** and with the same value

So the value of problem **SR** is a **lower bound** for **QCQP**

Back to general QCQP

$$\textbf{(QCQP):} \quad \begin{array}{ll} \min & x^T Q x + 2c^T x \\ \text{s.t.} & x^T A_i x + 2b_i^T x + r_i \ \geq 0 \qquad i = 1, \ldots, m \\ & x \in \mathbb{R}^n. \end{array}$$

$\rightarrow$ form the **semidefinite relaxation**

$$\textbf{(SR):} \quad \begin{array}{ll} \min & \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix} \bullet X \\ \text{s.t.} & \begin{pmatrix} r_i & b_i^T \\ b_i & A^i \end{pmatrix} \bullet X \ \geq 0 \qquad i = 1, \ldots, m \\ & X \succeq 0, \quad X_{11} = 1. \end{array}$$

Here, for symmetric matrices $\boldsymbol{M}$, $\boldsymbol{N}$,

$$M \bullet N \ = \ \sum_{h,k} M_{hk} N_{hk}$$

Why do we call it a relaxation?

Given $\boldsymbol{x}$ feasible for **QCQP**, the matrix $\begin{pmatrix} \mathbf{1} \\ \boldsymbol{x} \end{pmatrix} (\mathbf{1}, \boldsymbol{x}^T)$ feasible for **SR** and with the same value

So the value of problem **SR** is a **lower bound** for **QCQP**

So if **SR** has a **rank-1 solution**, the lower bound is **exact**.

Back to general QCQP

$$\textbf{(QCQP):} \quad \begin{array}{ll} & \min \ x^T Q x + 2c^T x \\ \text{s.t.} & x^T A_i x + 2b_i^T x + r_i \geq 0 \qquad i = 1, \ldots, m \\ & x \in \mathbb{R}^n. \end{array}$$

$\rightarrow$ form the **semidefinite relaxation**

$$\textbf{(SR):} \quad \begin{array}{ll} & \min \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix} \bullet X \\ \text{s.t.} & \begin{pmatrix} r_i & b_i^T \\ b_i & A^i \end{pmatrix} \bullet X \geq 0 \qquad i = 1, \ldots, m \\ & X \succeq 0, \quad X_{11} = 1. \end{array}$$

Here, for symmetric matrices $\boldsymbol{M}$, $\boldsymbol{N}$,

$$M \bullet N = \sum_{h,k} M_{hk} N_{hk}$$

Why do we call it a relaxation?

Given $\boldsymbol{x}$ feasible for **QCQP**, the matrix $\begin{pmatrix} \mathbf{1} \\ \boldsymbol{x} \end{pmatrix} (\mathbf{1}, \boldsymbol{x}^T)$ feasible for **SR** and with the same value

So the value of problem **SR** is a **lower bound** for **QCQP**

So if **SR** has a **rank-1 solution**, the lower bound is **exact**.

Unfortunately, **SR typically does not** have a rank-1 solution.

Theorem (Pataki, 1998):

An SDP

$$
\begin{array}{rll}
\textbf{(SR):} & \min \; M \bullet X & \\
\text{s.t.} & N^i \bullet X \; \geq b_i & i = 1, \ldots, m \\
& X \succeq 0, \quad X \text{ an } n \times n \text{ matrix,} &
\end{array}
$$

always has a solution of rank $\boldsymbol{O(m^{1/2})}$, and there exist examples where this condition is attained.

Theorem (Pataki, 1998):

An SDP

$$\begin{aligned} \textbf{(SR):} \quad & \min \; M \bullet X \\ \text{s.t.} \quad & N^i \bullet X \; \geq b_i \qquad i = 1, \ldots, m \\ & X \succeq 0, \quad X \text{ an } n \times n \text{ matrix,} \end{aligned}$$

always has a solution of rank $\boldsymbol{O(m^{1/2})}$, and there exist examples where this condition is attained.

Observation (Lavaei and Low):

The SDP relaxation of practical AC-OPF instances **can** have a rank-1 solution, or the solution **may** be relatively easy to massage into rank-1 solutions (also see earlier work of Bai et al)

Can we leverage this observation into practical, globally optimal algorithms for AC-OPF?

In the context of AC-OPF

Recall: in AC-OPF we denote the voltage of bus k as $\boldsymbol{e_k + j f_k}$

Power flow basic equations:

$$\boldsymbol{P_k^{\min}} \leq \sum_{km \in \delta(k)} \left[\boldsymbol{g_{km}}(e_k^2 + f_k^2) - \boldsymbol{g_{km}}(e_k e_m + f_k f_m) + \boldsymbol{b_{km}}(e_k f_m - f_k e_m) \right] \leq \boldsymbol{P_k^{\max}}$$

$$\boldsymbol{Q_k^{\min}} \leq \sum_{km \in \delta(k)} \left[-\boldsymbol{b_{km}}(e_k^2 + f_k^2) + \boldsymbol{b_{km}}(e_k e_m + f_k f_m) + \boldsymbol{g_{km}}(e_k f_m - f_k e_m) \right] \leq \boldsymbol{Q_k^{\max}}$$

$$\boldsymbol{(V_k^{\min})^2} \leq e_k^2 + f_k^2 \leq \boldsymbol{(V_k^{\max})^2},$$

for each bus $\boldsymbol{k = 1, 2, \ldots, n}$

- A direct SDP relaxation will produce a $2n \times 2n$ matrix $\boldsymbol{W}$ that approximates $\begin{pmatrix} \boldsymbol{1} \\ \boldsymbol{e} \\ \boldsymbol{f} \end{pmatrix} \boldsymbol{(1, e^T, f^T)}$

  (Or we can work directly with complex quantities)
- SDP-based lower-bounding algorithms also seek to produce low-rank $\boldsymbol{W}$.

Higher-order SDP relaxations

Consider the polynomial optimization problem

$$f_0^* \quad \doteq \quad \min \{\, f_0(x) \ : \ f_i(x) \geq 0, \quad 1 \leq i \leq m, \quad x \in \mathbb{R}^n \},$$

where each $f_i(x)$ is a **polynomial** i.e. $f_i(x) \ = \ \sum_{\pi \in S(i)} a_{i,\pi}\, x^{\pi}$.

- Each π is a tuple $\pi_1, \pi_2, \ldots, \pi_n$ of **nonnegative integers**, and $x^{\pi} \ \doteq \ x_1^{\pi_1}\, x_2^{\pi_2} \ \ldots \ x_n^{\pi_n}$
- Each $S(i)$ is a finite set of **tuples**, and the $a_{i,\pi}$ are reals.

Higher-order SDP relaxations

Consider the polynomial optimization problem

$$f_0^* \doteq \min \{ f_0(x) \ : \ f_i(x) \geq 0, \quad 1 \leq i \leq m, \quad x \in \mathbb{R}^n \},$$

where each $f_i(x)$ is a **polynomial** i.e. $f_i(x) = \sum_{\pi \in S(i)} a_{i,\pi} \, x^{\pi}$.

- Each π is a tuple $\pi_1, \pi_2, \ldots, \pi_n$ of **nonnegative integers**, and $x^{\pi} \doteq x_1^{\pi_1} x_2^{\pi_2} \ldots x_n^{\pi_n}$

- Each $S(i)$ is a finite set of **tuples**, and the $a_{i,\pi}$ are reals.

Moment Relaxations

- Introduce a variable X_{π} used to represent each monomial x^{π} of degree $\leq d$, for some integer d.
- This set of monomials includes all of those appearing in the polynomial optimization problem as well as $x^0 = 1$.
- If we replace each x^{π} in the formulation with the corresponding X_{π} we obtain a *linear* relaxation.
- Let X denote the vector of all such monomials. Then $X X^T \succeq 0$ and of rank one. The semidefinite constraint strengthens the formulation.
- Further semidefinite constraints are obtained from the constraints.

Challenges and opportunities

- Semidefinite programs can be very difficult to solve, **especially** large ones. Poor numerical conditioning can also engender difficulties.

- Even for $\mathbf{d = 2}$, an AC-OPF instance on a large grid can yield a large SDP, and problematic values for physical parameters (impedances) can yield difficult numerics.

- However, practical AC-OPF instances tend to arise on networks with ***structured sparsity***: low tree-width.

- Low tree-width naturally translates into structured sparsity of the matrices encountered in the solution of the SDPs

CAUTION

CAUTION

sparsity $\neq$ small tree-width

e.g. a $\boldsymbol{k} \times \boldsymbol{k}$ grid (max degree 4) is sparse but has treewidth $\boldsymbol{k}$

most authors write "sparsity" but mean *structured sparsity*

Challenges and opportunities

- Semidefinite programs can be very difficult to solve, **especially** large ones. Poor numerical conditioning can also engender difficulties.

- Even for $\mathbf{d = 2}$, an AC-OPF instance on a large grid can yield a large SDP, and problematic values for physical parameters (impedances) can yield difficult numerics.

- However, practical AC-OPF instances tend to arise on networks with ***structured sparsity***: low tree-width.

- Low tree-width naturally translates into structured sparsity of the matrices encountered in the solution of the SDPs

- This feature can be exploited by SDP algorithms: the **matrix completion** theorem

- This point has been leveraged by several researchers: Lavaei and Low, Hiskens and Molzahn, and others

Newer Results on OPF

1. Heuristic methods using SDP relaxations, and related
2. Algorithms using MINLP- and IP-inspired techniques

Obtaining low-rank near-optimal solutions to SDP relaxations

(Madani, Sojoudi, Lavaei)

Key points:

- Optimal solution to SDP relaxation of OPF may have high rank – even if optimal or near-optimal solutions have low rank, or even rank 1.

  Remark. Interior point algorithms for SDP tend to find highest rank optimal solutions.

- We need efficient procedures to find such solutions.

Obtaining low-rank near-optimal solutions to SDP relaxations

(Madani, Sojoudi, Lavaei)

Key points:

- Optimal solution to SDP relaxation of OPF may have high rank – even if optimal or near-optimal solutions have low rank, or even rank 1.

  Remark. Interior point algorithms for SDP tend to find highest rank optimal solutions.

- We need efficient procedures to find such solutions.

Typical objective of AC-OPF: minimize cost of active power generation

$$\min \sum_{k \in \mathbb{G}} f_k(P_k)$$

$\mathbb{G}$ = set of generators, P_k = active power generation at bus k
f_k = a convex quadratic
$\rightarrow$ potentially, many solutions to SDP attain same $\sum_{k \in \mathbb{G}} f_k(P_k)$

Obtaining low-rank near-optimal solutions to SDP relaxations

(Madani, Sojoudi, Lavaei)

Perturbed objective for AC-OPF:

$$\min \sum_{k \in \mathbb{G}} f_k(P_k) \; + \; \epsilon \sum_{k \in \mathbb{G}} Q_k$$

Q_k = reactive power generation at bus k

Why:

- ϵ small does not change problem "much"
- penalization tends to select a subset of (near) optimal solutions which additionally incur low reactive power generation
- can be argued that the penalization should decrease the rank of the solution to SDP

Obtaining low-rank near-optimal solutions to SDP relaxations

(Madani, Sojoudi, Lavaei)

Perturbed objective for AC-OPF:

$$\min \sum_{k \in \mathbb{G}} f_k(P_k) \; + \; \epsilon \sum_{k \in \mathbb{G}} Q_k$$

Q_k = reactive power generation at bus k

Why:

- ϵ small does not change problem "much"
- penalization tends to select a subset of (near) optimal solutions which additionally incur low reactive power generation
- reactive power on $k \rightarrow m$: $\approx |V_k|^2 g + V_k V_m^* b$
- But $|V|^2 \approx e$
- And $W_{kk} W_{mm} \geq W_{km} W_{mk}^*$ (positive-semidefiniteness) and $b \leq 0$

Improving SDP relaxations of AC-OPF

Molzahn, Hiskens, and Molzahn, Josz, Hiskens, Panciatici

1. SDP relaxation can sometimes fail; relying on the (full) higher moment relaxations can yield tighter convex relaxations but at huge computational cost.

An alternative: selectively use higher-order relaxations at different buses, in order to (locally) better represent the power flow equations at such buses.

HEURISTIC

(a) Construct a set of "bags" (sets of nodes) such that each line has both ends in at least one bag, and such that the largest such bag is as small as we can make it (**remark:** this is an estimate of treewidth).

(a.1) Initially we use as the monomials for the moment relaxation the set of all pairs of nodes that appear in each bag.

(b) Solve relaxation of OPF and construct *nearest* rank-1 matrix to the solution to the SDP (Ekart-Young metric).

c) This solution implies a vector of voltages and power injections. For each "bag", consider the bus with the highest infeasibility (e.g. power flow mismatch); use a heuristic rule that parameterizes this infeasibility to add further momonials chosen from subsets of that bag (more infeasible $\Rightarrow$ higher-order moments). **Repeat.**

Improving SDP relaxations of AC-OPF

Molzahn, Hiskens, and Molzahn, Josz, Hiskens, Panciatici

2. SDP (or moment) relaxation relaxation often prove tight lower bounds on AC-OPF; but how do we recover near-optimal rank-1 solutions?

IDEA:

(a) First, let c^* be the value of the SDP relaxation and let $\epsilon > 0$ be a desired tolerance. Suppose we add the constraint

$$\textbf{OPF cost} \quad \leq \quad \boldsymbol{c^*(1+\epsilon)}$$

to the constraints in the relaxation.

(b) Assuming (as one hope) there is a feasible solution to AC-OPF of cost $\leq \boldsymbol{c^*(1+\epsilon)}$ this constraint is not limiting. But we need to find a rank-1 solution that has this cost.

(c) The final ingredient: modify the objective in AC-OPF so as to more naturally produce rank-1 solutions. The authors propose a function that better accounts for reactive power injections.

Note: Step (a) makes it more likely that the objective modification in (c) does not produce much more expensive solutions.

Improving SDP relaxations of AC-OPF

Molzahn, Hiskens, and Molzahn, Josz, Hiskens, Panciatici

3. SDP (or moment) relaxation relaxation often prove tight lower bounds on AC-OPF; but not always. A conjecture was (is?) that this behavior is related to the particular physical characteristics of the example at hand.

For example, an early idea was to perturb resistances so that they are all positive and large enough.

However, the authors provide a class of 3-bus examples where two equivalent reformulations give rise to SDP relaxations of very different strength.

Remark. In the traditional 0-1 integer programming world, the idea that a problem can be reformulated so as to better leverage the strength of a particular solution technique is well-known; and general principles have been derived. An interesting question is whether such thinking can be extended to the AC-OPF setting (or to polynomial optimization in general).

Better SOCP Relaxations (Kocuk, Dey, Sun)

- Use SOCP instead of SDP to obtain tight relaxations that are (much) easier to solve
- Several observations lead to interesting inequalities.

Idea 1. Exploit connection to polar representation.
Recall that for a bus $\boldsymbol{k}$ the voltage satisfies

$$\boldsymbol{V_k} \;=\; |\boldsymbol{V_k}|\boldsymbol{e^{j\theta_k}} \;=\; \boldsymbol{e_k} + \boldsymbol{j f_k}$$

and basic power flow constraints include, e.g.:

$$P_k^{\min} \le \sum_{km \in \delta(k)} \left[g_{km}\,(\boldsymbol{e_k^2} + \boldsymbol{f_k^2}) - g_{km}\,(\boldsymbol{e_k e_m} + \boldsymbol{f_k f_m}) + b_{km}\,(\boldsymbol{e_k f_m} - \boldsymbol{f_k e_m}) \right] \le P_k^{\max}$$

Here,

$$\begin{aligned}
\boldsymbol{c_{kk}} &= \boldsymbol{e_k^2} + \boldsymbol{f_k^2} = |\boldsymbol{V_k}|^2 \\
\boldsymbol{c_{km}} &= \boldsymbol{e_k e_m} + \boldsymbol{f_k f_m} = |\boldsymbol{V_k}||\boldsymbol{V_m}|\cos(\boldsymbol{\theta_k} - \boldsymbol{\theta_m}) \\
\boldsymbol{s_{km}} &= \boldsymbol{e_k f_m} - \boldsymbol{f_k e_m} = -|\boldsymbol{V_k}||\boldsymbol{V_m}|\sin(\boldsymbol{\theta_k} - \boldsymbol{\theta_m})
\end{aligned}$$

So $\boldsymbol{c_{kk} c_{mm}} = \boldsymbol{c_{km}^2} + \boldsymbol{s_{km}^2}$ for any line $\boldsymbol{km}$

Reformulation to power flow equations(?):

$$P_k^{\min} \;\leq\; \sum_{km\,\in\,\delta(k)} \left[g_{km}(e_k^2+f_k^2) - g_{km}(e_k e_m + f_k f_m) + b_{km}(e_k f_m - f_k e_m)\right] \;\leq\; P_k^{\max}$$

$$Q_k^{\min} \;\leq\; \sum_{km\,\in\,\delta(k)} \left[-b_{km}(e_k^2+f_k^2) + b_{km}(e_k e_m + f_k f_m) + g_{km}(e_k f_m - f_k e_m)\right] \;\leq\; Q_k^{\max}$$

$$(V_k^{\min})^2 \quad \leq \quad e_k^2 + f_k^2 \quad \leq \quad (V_k^{\max})^2,$$

for each bus $\boldsymbol{k = 1, 2, \ldots}$, and

$$\boldsymbol{c_{km}^2 + s_{km}^2} = \boldsymbol{c_{kk}c_{mm},\; c_{km} = c_{mk}},\; \boldsymbol{s_{km} = -s_{mk}} \quad \text{for any line } \boldsymbol{km}$$

Reformulation to power flow equations(?):

$$P_k^{\min} \;\leq\; \sum_{km \in \delta(k)} [g_{km}\; \boldsymbol{c_{kk}} - g_{km}\; \boldsymbol{c_{km}} + b_{km}\; \boldsymbol{s_{km}}] \;\leq\; P_k^{\max}$$

$$Q_k^{\min} \;\leq\; \sum_{km \in \delta(k)} [-b_{km}\; \boldsymbol{c_{kk}} + b_{km}\; \boldsymbol{c_{km}} + g_{km}\; \boldsymbol{s_{km}}] \;\leq\; Q_k^{\max}$$

$$(V_k^{\min})^2 \quad \leq \quad \boldsymbol{c_{kk}} \quad \leq \quad (V_k^{\max})^2,$$

for each bus $\boldsymbol{k = 1, 2, \ldots}$, and

$$\boldsymbol{c_{km}^2 + s_{km}^2} = \boldsymbol{c_{kk}c_{mm},\; c_{km} = c_{mk},\; s_{km} = -s_{mk}} \quad \text{for any line } \boldsymbol{km}$$

(Expósito and Ramos (1999), Jabr (2006), others ?)

Reformulation to power flow equations as an SOCP:

$$P_k^{\min} \;\leq\; \sum_{km\,\in\,\delta(k)} \left[g_{km}\,\boldsymbol{c_{kk}} - g_{km}\,\boldsymbol{c_{km}} + b_{km}\,\boldsymbol{s_{km}}\right] \;\leq\; P_k^{\max}$$

$$Q_k^{\min} \;\leq\; \sum_{km\,\in\,\delta(k)} \left[-b_{km}\,\boldsymbol{c_{kk}} + b_{km}\,\boldsymbol{c_{km}} + g_{km}\,\boldsymbol{s_{km}}\right] \;\leq\; Q_k^{\max}$$

$$(V_k^{\min})^2 \quad \leq \quad \boldsymbol{c_{kk}} \quad \leq \quad (V_k^{\max})^2,$$

for each bus $\boldsymbol{k = 1, 2, \ldots}$, and

$$\boldsymbol{c_{km}^2 + s_{km}^2} \;\leq\; \boldsymbol{c_{kk}c_{mm},}\;\; \boldsymbol{c_{km} = c_{mk}},\;\; \boldsymbol{s_{km} = -s_{mk}} \quad \text{for any line } \boldsymbol{km}$$

(Expósito and Ramos (1999), Jabr (2006), others ?)

Reformulation to power flow equations as an SOCP:

$$P_k^{\min} \;\leq\; \sum_{km\,\in\,\delta(k)} \left[g_{km}\; \boldsymbol{c_{kk}} - g_{km}\; \boldsymbol{c_{km}} + b_{km}\; \boldsymbol{s_{km}}\right] \;\leq\; P_k^{\max}$$

$$Q_k^{\min} \;\leq\; \sum_{km\,\in\,\delta(k)} \left[-b_{km}\; \boldsymbol{c_{kk}} + b_{km}\; \boldsymbol{c_{km}} + g_{km}\; \boldsymbol{s_{km}}\right] \;\leq\; Q_k^{\max}$$

$$(V_k^{\min})^2 \quad \leq \quad \boldsymbol{c_{kk}} \quad \leq \quad (V_k^{\max})^2,$$

for each bus $\boldsymbol{k = 1, 2, \ldots}$, and

$$\boldsymbol{c_{km}^2 + s_{km}^2} \;{\color{red}\leq}\; \boldsymbol{c_{kk}c_{mm},\;\; c_{km} = c_{mk},\;\; s_{km} = -s_{mk}} \quad \text{for any line } \boldsymbol{km}$$

(Expósito and Ramos (1999), Jabr (2006), others ?)

Is the relationship invertible?

$$\begin{aligned}
{\color{red}\boldsymbol{c_{kk}}} &= \boldsymbol{e_k^2 + f_k^2} &&= \;\; |\boldsymbol{V_k}|^{\mathbf{2}} \\
{\color{red}\boldsymbol{c_{km}}} &= \boldsymbol{e_k e_m + f_k f_m} &&= \;\; \boldsymbol{|V_k||V_m|\cos(\theta_k - \theta_m)} \\
{\color{red}\boldsymbol{s_{km}}} &= \boldsymbol{e_k f_m - f_k e_m} &&= \;\; \boldsymbol{-|V_k||V_m|\sin(\theta_k - \theta_m)}
\end{aligned}$$

Stronger SOCP relaxation?

Exploit further relationships among $\boldsymbol{c}, \boldsymbol{s}$ variables.

Given a cycle $\mathcal{C}$, we must have $\sum_{km \in \mathcal{C}} \theta_{km} = 0$ (here $\theta_{km} = \theta_k - \theta_m$)

This can be relaxed into the condition $\cos\left(\sum_{km \in \mathcal{C}} \theta_{km}\right) = 1$.

But note that e.g. $\cos(\theta_{km}) = \frac{c_{km}}{\sqrt{c_{kk} c_{mm}}}$ (and likewise with $\sin(\theta_{km})$).

Furthermore, given a cycle $\mathcal{C}$, we can expand $\sum_{km \in \mathcal{C}} \cos(\theta_{km})$ into a polynomial in the quantities $\cos(\theta_{km})$ and $\sin(\theta_{km})$ (over all $km \in \mathcal{C}$).

- This yields a degree- $|\mathcal{C}|$ homogeneous polynomial equation in the quantities c_{km} and s_{km}.
- This equation can be approximately convexified (linearized!) using the McCormick reformulation trick.

- **Which** cycles? **How many** cycles? How to use McCormick?

Even stronger than SOCP relaxation?

Exploit stronger relationships among $\boldsymbol{c}, \boldsymbol{s}$ variables.

Let $\tilde{\boldsymbol{v}} = (\boldsymbol{e_1, e_2, \ldots, e_n, f_1, \ldots, f_n})^{\boldsymbol{T}}$ and $\boldsymbol{W = vv^T}$. Then the following hold ($\boldsymbol{n}$ = number of buses)

Procedure

$$c_{kk} = e_k^2 + f_k^2 \; = \; W_{k,k} + W_{k+n,k+n} \tag{8a}$$

$$c_{km} = e_k e_m + f_k f_m \; = \; W_{k,m} + W_{k+n,m+n} \tag{8b}$$

$$s_{km} = e_k f_m - f_k e_m \; = \; W_{k,m+n} - W_{m,k+n} \tag{8c}$$

- For any cycle $\mathcal{C}$, check if there exist $W \succeq 0$ satisfying (8) for every node and edge of $\mathcal{C}$.
- A "small" SDP if $\mathcal{C}$ is small
- Do this for each cycle of a *cycle basis*

Outline of computational results

- Several relaxations compared

- Here: basic (c,s)-SOCP, and (c,s)SDP (separation variant)

Outline of computational results

- Standard problem instances: (c,s)-SOCP already very strong and runs in seconds
- (c,s)-SDP more accurate but significantly more expensive
- Hard instances: (c,s)-SOCP can have gaps of around **5%**
- Hard instances: (c,s)-SDP can have gaps of around **4%**
- Hard instances: QC-relaxation can be competitive
- Variable bound tightening is **very** important
- Upper bounds found using IPOPT – not sensitive to starting point

Cut-and-branch for complex QCQP

(Chen, Atamtürk, Oren)

Complex QCQP:

$$
\begin{array}{ll}
\text{Min} & x^* Q_0 x + \text{Re}(c_0^* x) + b_0 \\
\text{s.t.} & \\
& x^* Q_i x + \text{Re}(c_i^* x) + b_i \geq 0, \quad i = 1, \ldots, m \\
& \text{bounded } x \in \mathbb{C}^n
\end{array}
$$

Cut-and-branch for complex QCQP

(Chen, Atamtürk, Oren)

SDP relaxation:

$$
\begin{aligned}
\text{Min} \quad & < Q_0, X > + \mathrm{Re}(c_0^* x) + b_0 \\
\text{s.t.} \quad & \\
& < Q_i, X > + \mathrm{Re}(c_i^* x) + b_i \geq 0, \quad i = 1, \ldots, m \\
& \text{bounded } x \in \mathbb{C}^n \\
& \begin{pmatrix} 1 & x^* \\ x & X \end{pmatrix} \succeq 0.
\end{aligned}
$$

Cut-and-branch for complex QCQP

(Chen, Atamtürk, Oren)

SDP relaxation:

$$\begin{aligned}
&\text{Min} \;\; < Q_0, X > + \operatorname{Re}(c_0^* x) + b_0 \\
&\text{s.t.} \\
&\qquad < Q_i, X > + \operatorname{Re}(c_i^* x) + b_i \ge 0, \quad i = 1, \ldots, m \\
&\qquad \text{bounded } x \in \mathbb{C}^n \\
&\qquad \begin{pmatrix} 1 & x^* \\ x & X \end{pmatrix} \succeq 0.
\end{aligned}$$

Theorem. An $\boldsymbol{n} \times \boldsymbol{n}$ matrix has rank 1 if and only if all of its $\mathbf{2} \times \mathbf{2}$ principal minors are zero.

- Provides a venue for finding violated inequalities
- **Algorithm:** solve current relaxation (starting with SDP relaxation) then if rank > 1, then either *cut* or *branch* (spatial branching) using the Theorem to identify a matrix entry to work with. Repeat.

Cut-and-branch for complex QCQP

(Chen, Atamtürk, Oren)

Cutting. Given parameters $\boldsymbol{L_{11}, U_{11}, L_{12}, U_{12}, L_{22}, U_{22}}$, consider the set of **Hermitian** matrices

$$\begin{pmatrix} \boldsymbol{W_{11}} & \boldsymbol{W_{12}} \\ \boldsymbol{W_{12}} & \boldsymbol{W_{22}} \end{pmatrix}$$

where $W_{pq} = W_{pq} + jT_{pq}$ that satisfy

$$\begin{aligned} & L_{11} \leq W_{11} \leq U_{11}, \quad L_{22} \leq W_{22} \leq U_{22} \\ & L_{12}W_{12} \leq T_{12} \leq U_{12}W_{12} \\ & W_{11}W_{22} = W_{12}^2 + T_{12}^2 \end{aligned}$$

This represents a relaxation of the (positive-semidefinite, rank ≤ 1) condition.

The authors provide a description of the *convex hull* of the set of such matrices. Any inequality valid for the convex hull can be applied to any $\mathbf{2} \times \mathbf{2}$ principal submatrix of the matrix $\boldsymbol{X}$ in the formulation.

More Kocuk, Dey, Sun (2017)

- Branch-and-cut algorithm: able to get $< \mathbf{1\%}$ gap on average in $<$ half hour even on harder instances

- SOCP relaxation combining $\boldsymbol{c}, \boldsymbol{s}$ variables and $\boldsymbol{\theta}$ variables

More Kocuk, Dey, Sun (2017)

- Branch-and-cut algorithm: able to get $< \mathbf{1\%}$ gap on average in $<$ half hour even on harder instances
- Basic relaxation: SOCP
- Some new ideas

Theorem. A Hermitian $\boldsymbol{n} \times \boldsymbol{n}$ matrix $\boldsymbol{X}$ is positive-semidefinite iff

- The diagonal is real and nonnegative
- Every $\mathbf{2} \times \mathbf{2}$ minor equals zero

$$\begin{vmatrix} \boldsymbol{X_{ij}} & \boldsymbol{X_{ik}} \\ \boldsymbol{X_{lj}} & \boldsymbol{X_{lk}} \end{vmatrix} = \mathbf{0} \quad \forall\, \boldsymbol{i, j, k, l}$$

Apply this idea when $\boldsymbol{X}$ is the (approximation) to $\boldsymbol{VV^*}$

First example: for all $\boldsymbol{k, m}$

$$\begin{vmatrix} V_k V_k^* & V_k V_m^* \\ V_m V{*}_k & V_m V_m^* \end{vmatrix} = 0$$

What does this mean?

- Recall $V_i = e_i + j f_i$ for all buses i
- So $V_k V_k^* = e_k^2 + f_k^2 = c_{kk}$ and $V_m V_m^* = c_{mm}$
- And $V_m V_k^* = e_k e_m + f_k f_m - j(e_k f_m - f_k e_m) = c_{km}$
  and $V_k V_m^* = c_{km} + j s_{km}$
- So above condition states: $c_{kk} c_{mm} = c_{km}^2 + s_{km}^2$
- $\{c_{kk} c_{mm} \geq c_{km}^2 + s_{km}^2\} \cap \{c_{kk} c_{mm} \leq c_{km}^2 + s_{km}^2\}$
- Linear **underestimator** for $\sqrt{c_{kk} c_{mm}}$
- Linear **overestimator** for $\sqrt{c_{km}^2 + s_{km}^2}$

- Other minors similarly handled over $\boldsymbol{c, s}$ variables

Phase angles

Recall that

$$\begin{aligned} c_{km} &= e_k e_m + f_k f_m &= |V_k||V_m|\cos(\theta_k - \theta_m) \\ s_{km} &= e_k f_m - f_k e_m &= -|V_k||V_m|\sin(\theta_k - \theta_m) \end{aligned}$$

So:

$$s_{km}/c_{km} = \tan(\theta_k - \theta_m)$$

- Can get bounds on s_{km}, c_{km} based on bounds on $\theta_k - \theta_m$
- Can also get bounds on s_{km}, c_{km} based on bounds for $|V_k|$, $|V_m|$.
- But also valid inequalities for the convex hull of (box-constrained) $\{(c, s, \theta) \, : \, c/s = \arctan \theta\}$

New developments on Polynomial Optimization

Approximate reformulation as 0,1 IP (Bienstock and Muñoz)

Bounded variable QCQP:

$$\begin{aligned} & \min \; x^T Q x \, + \, 2c^T x \\ \text{s.t.} \quad & x^T A_i x \, + \, 2b_i^T x \, + \, r_i \; \geq 0 \qquad i = 1, \ldots, m \\ & 0 \leq x_j \leq 1, \quad \forall j. \end{aligned}$$

Approximate reformulation as 0,1 IP (Bienstock and Muñoz)

Bounded variable QCQP:

$$\begin{aligned} & \min \ x^T Q x + 2c^T x \\ \text{s.t.} \quad & x^T A_i x + 2b_i^T x + r_i \geq 0 \qquad i = 1, \ldots, m \\ & 0 \leq x_j \leq 1, \quad \forall j. \end{aligned}$$

Main technique: approximate representation using binary variables:

$$\boldsymbol{x_j \approx \sum_{k=1}^{L} 2^{-k} y_k}, \quad \text{each } \boldsymbol{y_k = 0} \text{ or } \mathbf{1}$$

- Error $\leq \mathbf{2^{-L}}$.
- Apply parsimoniously
- If an $\boldsymbol{x_j}$ is approximated this way then a bilinear form $\boldsymbol{x_j x_i}$ can be *represented* within error $\mathbf{2^{-L}}$ using McCormick
- Other bilinear forms approximated using standard McCormick for continuous variables
- **Main advantage:** can leverage robust, modern linear 0,1 solvers.

New LP Hierarchies (Lasserre, Toh, Yang)

Consider the **polynomial optimization problem**

$$f^* \doteq \operatorname{Min} \; f(x) \quad \text{s.t.} \quad g_j(x) \geq 0, \quad j = 1, \ldots, m$$

where $f(x)$ and the $g_j(x)$ are polynomials.

New LP Hierarchies (Lasserre, Toh, Yang)

Consider the **polynomial optimization problem**

$$\begin{aligned} \boldsymbol{f^*} \ \doteq \ & \text{Min} \ \ f(x) \\ & \text{s.t.} \\ & \qquad\qquad g_j(x) \geq 0, \quad j = 1, \ldots, m \end{aligned}$$

where $f(x)$ and the $g_j(x)$ are polynomials.

Let $\boldsymbol{d \geq 1}$ integral. Then

$$\begin{aligned} \boldsymbol{f^*} \ = \ & \text{Min} \ \ f(x) \\ & \text{s.t.} \\ & \qquad \prod_{j=1}^{m} g_j(x)^{\alpha_j}(1 - g_j(x))^{\beta_j} \ \geq \ 0 \quad \forall (\alpha, \beta) \in \mathbb{N}_d^{2m} \end{aligned}$$

Here, $\mathbb{N}_{\boldsymbol{d}}^{\boldsymbol{2m}}$ is the set of nonnegative integer vectors $\boldsymbol{\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m}$ with

$$\boldsymbol{\textstyle\sum_{j=1}^{m} \alpha_j \ \geq \ d, \ \sum_{j=1}^{m} \beta_j \ \geq \ d}$$

New LP Hierarchies (Lasserre, Toh, Yang)

Consider the **polynomial optimization problem**

$$\boldsymbol{f^*} \doteq \text{Min } f(x)$$
$$\text{s.t.}$$
$$g_j(x) \geq 0, \quad j = 1, \ldots, m$$

where $f(x)$ and the $g_j(x)$ are polynomials.

Let $\boldsymbol{d \geq 1}$ integral. Then

$$\boldsymbol{f^*} = \text{Min } f(x)$$
$$\text{s.t.}$$
$$\prod_{j=1}^{m} g_j(x)^{\alpha_j}(1 - g_j(x))^{\beta_j} \geq 0 \quad \forall (\alpha, \beta) \in \mathbb{N}_d^{2m}$$

Here, $\mathbb{N}_{\boldsymbol{d}}^{\boldsymbol{2m}}$ is the set of nonnegative integer vectors $\boldsymbol{\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m}$ with

$$\boldsymbol{\sum_{j=1}^{m} \alpha_j \geq d, \; \sum_{j=1}^{m} \beta_j \geq d}$$

Lagrangian relaxation:

$$\boldsymbol{f^*} \geq \quad \mathbf{sup}_{\boldsymbol{\lambda \geq 0}} \;\; \mathbf{inf}_{\boldsymbol{x}} \;\; \left[\boldsymbol{f_0(x) - \sum_{(\alpha,\beta)\in\mathbb{N}_d^{2m}} \lambda_{\alpha,\beta} \prod_{j=1}^{m} g_j(x)^{\alpha_j}(1 - g_j(x))^{\beta_j}}\right]$$

Lagrangian relaxation:

$$f^* \geq \quad \sup_{\lambda \geq 0} \; \inf_x \; \underbrace{\left[f_0(x) - \sum_{(\alpha,\beta)\in\mathbb{N}_d^{2m}} \lambda_{\alpha,\beta} \prod_{j=1}^{m} g_j(x)^{\alpha_j} (1 - g_j(x))^{\beta_j} \right]}_{L(x,\lambda)}$$

Lagrangian relaxation:

$$\boldsymbol{f^* \ \geq \ \ \sup_{\lambda \geq 0} \ \inf_x \ \underbrace{\left[f_0(x) - \sum_{(\alpha,\beta) \in \mathbb{N}_d^{2m}} \lambda_{\alpha,\beta} \prod_{j=1}^{m} g_j(x)^{\alpha_j} (1 - g_j(x))^{\beta_j} \right]}_{L(x,\lambda)}}$$

But for any $\boldsymbol{\lambda}$:

$$\boldsymbol{\inf_x L(x,\lambda) \ \geq \ \inf\{\, t \ : L(x,\lambda) - t} \text{ is SOS} \,\boldsymbol{\}}$$

SOS: sum-of-squares polynomials

- Can restrict to polynomials of bounded degree
- Resulting formulation can be solved using SDP
- SDPs can leverage structured sparsity (e.g. low treewidth)

RLT-POS (Dalkiran-Sherali, Sherali et al)

$$
\begin{aligned}
\text{Min} \quad & \phi_0(x) \\
\text{s.t.} \quad & \\
& \phi_r(x) \geq \beta_r, \quad r = 1, \ldots, R \\
& Ax = b \\
& 0 \leq l_j \leq x_j \leq u_j < \infty, \ \forall j
\end{aligned}
$$

where

$$
\phi_r(x) \doteq \sum_{t \in T_r} \alpha_{rt} \Big[\prod_{j \in J_{rt}} x_j \Big], \quad r = 0, \ldots, R.
$$

RLT-POS (Dalkiran-Sherali, Sherali et al)

$$\begin{aligned} \text{Min} \quad & \phi_0(x) \\ \text{s.t.} \quad & \\ & \phi_r(x) \geq \beta_r, \quad r = 1, \ldots, R \\ & Ax = b \\ & 0 \leq l_j \leq x_j \leq u_j < \infty, \; \forall j \end{aligned}$$

where

$$\phi_r(x) \doteq \sum_{t \in T_r} \alpha_{rt} \Big[\prod_{j \in J_{rt}} x_j \Big], \quad r = 0, \ldots, R.$$

REFORMULATION-LINEARIZATION

The **RLT** procedure (Sherali-Adams) linearizes the formulation by replacing each monomial with a new variable (underlying mathematical foundation related to moment relaxation)

$$\begin{aligned} \textbf{RLT:} \quad \text{Min} \quad & [\phi_0(x)]_L \\ \text{s.t.} \quad & \\ & [\phi_r(x)]_L \geq \beta_r, \; r = 1, \ldots, R \\ & Ax = b \\ & \left[\prod_{j \in J_1} (x_j - l_j) \prod_{j \in J_2} (u_j - x_j) \right]_L \geq 0, \; \forall \text{ appropriate } J_1, \; J_2 \\ & 0 \leq l_j \leq x_j \leq u_j < \infty, \; \forall j \end{aligned}$$

Here, the "L" operator

substitutes each monomial $\prod_{j \in J} x_j$ with a new variable X_J

Pros for RLT-POS:

1. It's an LP!
2. Convergence theory related to similar method for $\mathbf{0,1}$-integer programming.

Cons against RLT-POS:

1. It's a **BIG** LP! If we want to be guaranteed exactness.

Other technical details:

- Linearize monomials $\prod_{j\in J} x_j$ in a restricted fashion in order to keep LP small (e.g. use nonbasic variables from LP)
- Use SDP cuts
- Use *branching* (careful enumeration)

Approximate reformulation as 0,1 IP (Bienstock and Muñoz)

Bounded variable QCQP:

$$
\begin{array}{ll}
 & \min \; x^T Q x + 2c^T x \\
\text{s.t.} & x^T A_i x + 2b_i^T x + r_i \geq 0 \qquad i = 1, \ldots, m \\
 & 0 \leq x_j \leq 1, \quad \forall j.
\end{array}
$$

Approximate reformulation as 0,1 IP (Bienstock and Muñoz)

Bounded variable QCQP:

$$\begin{aligned} & \min \ x^T Q x \, + \, 2c^T x \\ \text{s.t.} \quad & x^T A_i x \, + \, 2b_i^T x \, + \, r_i \, \geq 0 \qquad i = 1, \ldots, m \\ & 0 \leq x_j \leq 1, \quad \forall j. \end{aligned}$$

Main technique: approximate representation using binary variables:

$$\boldsymbol{x_j \, \approx \, \sum_{k=1}^{L} 2^{-k} y_k,} \quad \text{each} \ \ \boldsymbol{y_k = 0} \text{ or} \boldsymbol{1}$$

- Error $\leq \mathbf{2^{-L}}$.
- Apply parsimoniously
- If an $\boldsymbol{x_j}$ is approximated this way then a bilinear form $\boldsymbol{x_j x_i}$ can be *represented* within error $\mathbf{2^{-L}}$ using McCormick
- Other bilinear forms approximated using standard McCormick for continuous variables
- **Main advantage:** can leverage robust, modern linear 0,1 solvers.