New Developments on OPF Problems

Daniel Bienstock, Columbia University

April 2016

Organization:

- 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! Read citations (end of presentation).
- In most cases we present an outline of results with some depth removed.

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 $\boldsymbol{e_k + j f_k}$

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

Basic power flow problem

Find a solution to:

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

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

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

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

Many possible variations/extensions, plus optimization versions

Quadratically constrained, quadratic programming problems (QCQPs)

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

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:

$$W^* \doteq \min \; -\sum_i x_i^2$$

$$\text{s.t.} \quad \sum_i w_i \, x_i = 0,$$

$$-1 \leq x_i \leq 1, \quad 1 \leq i \leq n.$$

Folklore result: QCQP is NP-hard

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

$$W^* \doteq \min \; -\sum_i x_i^2$$

$$\text{s.t.} \quad \sum_i w_i \, x_i = 0,$$

$$-1 \leq x_i \leq 1, \quad 1 \leq i \leq n.$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

so **linearly constrained** QCQP is as hard integer optimization

NO nice approximation algorithms exist for this class of problems

They are called ***strongly*** NP-hard

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"

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 $3v^6 w - v^4 + 7 = 0$.

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

$$\begin{aligned} c^2 &= 1 \\ v^2 - cv_2 &= 0 \\ v_2^2 - cv_4 &= 0 \\ v_2 v_4 - cv_6 &= 0 \\ v_6 w - cy &= 0 \\ 3cy - cv_4 &= -7 \end{aligned}$$

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

Back to general QCQP

Back to general QCQP

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

Back to general QCQP

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

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

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

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

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

Back to general QCQP

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

→ form the **semidefinite relaxation**

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

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

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

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

Back to general QCQP

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

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

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

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

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

Back to general QCQP

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

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

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

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 $(\mathbf{1}, \boldsymbol{x}^T) \begin{pmatrix} \mathbf{1} \\ \boldsymbol{x} \end{pmatrix}$ 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 $(\mathbf{1}, \boldsymbol{x}^T) \begin{pmatrix} \mathbf{1} \\ \boldsymbol{x} \end{pmatrix}$ 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

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

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 can 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
- Or we can work directly with complex quantities

Recall:

- Power injection on line $\boldsymbol{km} = V_k \boldsymbol{I_{km}^*} = \boldsymbol{V_k y_{km}^* (V_k^* - V_m^*)} = \boldsymbol{|V_k|^2 y_{km}^* - y_{km}^* V_k V_m^*}.$
- For systems that are voltagewise tightly constrained, $\boldsymbol{|V_k| \approx 1}$ (p.u.)
- So it is important to have a low-rank matrix with entries $\boldsymbol{V_k V_m^*}$.

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.

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 order $\leq d$, 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

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 $n \times n$ matrix with entries $\mathrm{Re}(V_k V_m^*)$

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 $\boldsymbol{c^*}$ be the value of the SDP relaxation and let $\epsilon > 0$ be a desired tolerance. Suppose we add the constraint

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

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, Andy Sun)

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

Idea 1. For a bus $\boldsymbol{k}$ and line $\boldsymbol{km}$ denote $\boldsymbol{c_{kk} = e_k^2 + f_k^2}$ (square of voltage magnitude), $\boldsymbol{c_{km} = e_k e_m + f_k f_m}$ and $\boldsymbol{s_{km} = e_k f_m - f_k e_m}$.
Then (prior observation by Expósito and Ramos, Jabr) we have

$$\boldsymbol{c_{km}^2 + s_{km}^2 \;=\; c_{kk} c_{mm}},$$

which is nonconvex, but can be relaxed as the SOCP constraint

$$\boldsymbol{c_{km}^2 + s_{km}^2 \;\leq\; c_{kk} c_{mm}},$$

$\rightarrow$ Use a convex formulation that involves these quantities.
Moreover, let $\boldsymbol{\tilde{v} \;=\; (e_1, e_2, \ldots, e_n, f_1, \ldots, f_n)^T}$ and $\boldsymbol{W = vv^T}$. Then the following hold

$$\begin{aligned} c_{km} &= W_{k,m} + W_{k+n,m+n} \\ s_{km} &= W_{k,m+n} - W_{m,k+n} \\ s_{kk} &= W_{k,k} + W_{m+n,m+n} \end{aligned}$$

Given a vector of values $\boldsymbol{c, s}$ we can efficiently check if a positive semidefinite matrix $\boldsymbol{W}$ satisfying these properties exists. And if not: we obtain a *cut* that we can use to strengthen the formulation. This gives rise to an iterative (cutting-plane). algorithm.

Better SOCP Relaxations (Kocuk, Dey, Andy Sun)

Idea 2. The $\boldsymbol{c, s}$ variables can be used to better describe relationships among voltages.

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

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

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

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

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

- Relationship with higher-order moment relaxations?

QC Relaxation (Coffrin, Hijazi, Van Hentenryck)

- Approximates trigonometric relationships and bilinear expressions
- Application to AC-OPF yields a convex relaxation

Given buses k and m, with voltage magnitudes v_k, v_m and phase angles θ_k and θ_m,

$$\boldsymbol{V_k V_m^* \quad = \quad v_k v_k \cos(\theta_k - \theta_m) \; + \; j v_k v_m \sin(\theta_k - \theta_m)}$$

To estimate (relax) expressions of this sort, we use two ideas.
Idea 1. For angle ϕ small enough, we can upper bound

$$\boldsymbol{\sin(\phi) \;\leq\; \cos(\phi/2)(\phi - \phi^u/2) \;+\; \sin(\phi^u/2)},$$

where $\phi \leq \phi^u$, and a similar lower bound can be obtained, and likewise $\cos(\phi)$ can be bounded.
Idea 2. (McCormick relaxation). The convex hull of a set of the form

$$\boldsymbol{\{xy \;:\; x^L \leq x \leq x^U,\; y^L \leq y \leq y^U\}}$$

(where x^L, x^U, y^L, y^U are parameters) is given by four linear inequalities, e.g. $\boldsymbol{xy \geq x^L y + y^L x - x^L y^L}$.

By first applying Idea 1 to the real and imaginary parts of $V_k V_m^*$ and then repeatedly applying Idea 2, we obtain convex approximations to a number of expressions arising in power flow formulae.

- This approach yields the aforementioned convex relaxation
- Initial numerical experiments appear very promising

New developments on Polynomial Optimization

Cut-and-branch for complex QCQP

(Chen, Atamtürk, Oren)

Complex QCQP:

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

Cut-and-branch for complex QCQP

(Chen, Atamtürk, Oren)

SDP relaxation:

$$\text{Min} \;\; < Q_0, X > + \operatorname{Re}(c_0^* x) + b_0$$

s.t.

$$< Q_i, X > + \operatorname{Re}(c_i^* x) + b_i \geq 0, \quad i = 1, \ldots, m$$

$$\text{bounded } x \in \mathbb{C}^n$$

$$\begin{pmatrix} 1 & x^* \\ x & X \end{pmatrix} \succeq 0.$$

Cut-and-branch for complex QCQP

(Chen, Atamtürk, Oren)

SDP relaxation:

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

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.

New LP Hierarchies (Lasserre, Toh, Yang)

Consider the **polynomial optimization problem**

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

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

New LP Hierarchies (Lasserre, Toh, Yang)

Consider the **polynomial optimization problem**

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

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

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

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

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

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

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:

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

Lagrangian relaxation:

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

But for any λ:

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

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 $\boldsymbol{\prod_{j \in J} x_j}$ with a new variable $\boldsymbol{X_J}$

Pros for RLT-POS:

1. It's an LP!
2. Convergence theory related to similar method for **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:

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

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:

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

- 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.

Small set of Citations

1. D. Bienstock and G. Muñoz, LP formulations for mixed-integer polynomial optimization problems, arXiv:1501.00288 (2014).
2. C. Chen, A. Atamtürk and S. Oren, A Spatial Branch-and-Cut Algorithm for Nonconvex QCQP with Bounded Complex Variables, *BCOL* Report 15.04, U. of California, Berkeley (2015) (submitted).
3. C. Coffrin, H. Hijazi, and Pascal Van Hentenryck. The QC Relaxation: A Theoretical and Computational Study on Optimal Power Flow, *IEEE Transactions on Power Systems*, 99, 1–11, September 2015.
4. E. Dalkiran and H. Sherali, RLT-POS: Reformulation-Linearization Technique-based optimization software for solving polynomial programming problems, *Mathematical Programming Computation*, February 2016.
5. B. Kocuk, S. Dey and A. Sun, Strong SOCP Relaxations for the Optimal Power Flow Problem, to appear, *Operations Research*.
6. J.B. Lasserre, K.-C. Toh and S. Yang, A bounded degree SOS hierarchy for polynomial optimization, arXiv: 1501:06126 (2015).
7. M. Laurent, Sum of squares, moment matrices and optimization over polynomials, *IMA*, (2010), 1–147.
8. S. Low, Convex Relaxation of Optimal Power Flow, *IEEE Trans. Control Network Sys.* Part I: Formulations and Equivalence (March 2014), Part II: Exactness (June 2014)
9. R. Madani, S. Sojoudi and Javad Lavaei, Convex Relaxation for Optimal Power Flow Problem: Mesh Networks, *IEEE Trans. Power Sys.*, 30, (2015), 199–211.
10. D.K. Molzahn and I.A. Hiskens, Convex Relaxations of Optimal Power Flow Problems: An Illustrative Example, to appear, *IEEE Trans. Circuits and Systems I: Regular Papers, Special Issue* (2016).

SEMI-INTRO

Repeats a couple of slides

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.

CAN SHOW: $f_0^* = \inf_{\mu} \mathbb{E}_{\mu} f_0(x)$, over all measures μ over $K \doteq \{x \in \mathbb{R}^n \ : \ f_i(x) \geq 0, \ 1 \leq i \leq m\}$.

Higher-order SDP relaxations

Consider the polynomial optimization problem

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

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

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

CAN SHOW $\boldsymbol{f_0^*} \ = \ \inf_\mu \mathbb{E}_\mu\, \boldsymbol{f_0(x)}$, over all measures $\boldsymbol{\mu}$ over $\boldsymbol{K} \doteq \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \boldsymbol{f_i(x) \geq 0, \ 1 \leq i \leq m}\}$.

$$\text{i.e.} \quad \boldsymbol{f_0^*} \ = \ \inf \left\{ \sum_{\pi \in S(0)} \boldsymbol{a_{0,\pi} y_\pi} \ : \ \boldsymbol{y} \text{ is a } \boldsymbol{K}\text{-moment} \right\}$$

Here, $\boldsymbol{y}$ is a $\boldsymbol{K}$-moment if there is a measure $\boldsymbol{\mu}$ over $\boldsymbol{K}$ with $\boldsymbol{y_\pi} \ = \ \mathbb{E}_\mu \boldsymbol{x^\pi}$ for each tuple $\boldsymbol{\pi}$

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.

CAN SHOW $f_0^* = \inf_{\mu} \mathbb{E}_{\mu} f_0(x)$, over all measures μ over $K \doteq \{x \in \mathbb{R}^n \,:\, f_i(x) \geq 0,\ 1 \leq i \leq m\}$.

$$\text{i.e.} \quad f_0^* = \inf\left\{ \sum_{\pi \in S(0)} a_{0,\pi} y_{\pi} \;:\; y \text{ is a } K\text{-moment}\right\}$$

Here, y is a K-moment if there is a measure μ over K with $y_{\pi} = \mathbb{E}_{\mu} x^{\pi}$ for each tuple π

(**Cough!** Here, y is an **infinite-dimensional** vector).

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.

CAN SHOW $f_0^* = \inf_{\mu} \mathbb{E}_{\mu}\, f_0(x)$, over all measures μ over $K \doteq \{x \in \mathbb{R}^n \;:\; f_i(x) \geq 0,\; 1 \leq i \leq m\}$.

$$\text{i.e.} \quad f_0^* = \inf \left\{ \sum_{\pi \in S(0)} a_{0,\pi} y_{\pi} \;:\; y \text{ is a } K\text{-moment} \right\}$$

Here, y is a K-moment if there is a measure μ over K with $y_{\pi} = \mathbb{E}_{\mu} x^{\pi}$ for each tuple π

(**Cough!** Here, y is an **infinite-dimensional** vector). Can we make an easier statement?

Recall:

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

where $f_i(x) \;=\; \sum_{\pi \in S(i)} a_{i,\pi}\, x^\pi$,

Thus, $f_0^* \;=\; \inf_\mu \mathbb{E}_\mu\, f_0(x)$, over all measures μ over $K \doteq \{x \in \mathbb{R}^n \;:\; f_i(x) \geq 0,\; 1 \leq i \leq m\}$.

So, $f_0^* \;=\; \inf_y \sum_\pi a_{0,\pi} y_\pi$, over all K-moment vectors y;

(y is a K-moment if there is a measure μ over K with $y_\pi \;=\; \mathbb{E}_\mu x^\pi$ for each tuple π)

$(K \doteq \{x \in \mathbb{R}^n \;:\; f_i(x) \geq 0,\; 1 \leq i \leq m\})$.

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

where $f_i(x) = \sum_{\pi \in S(i)} a_{i,\pi}\, x^\pi$.

So, $f_0^* = \inf_y \sum_\pi a_{0,\pi} y_\pi$, over all K-moment vectors y;

(y is a K-moment if there is a measure μ over K with $y_\pi = \mathbb{E}_\mu x^\pi$ for each tuple π)

$(K \doteq \{x \in \mathbb{R}^n \;:\; f_i(x) \geq 0,\; 1 \leq i \leq m\})$.

So: $y_0 = 1$. Can we say more?

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

where $f_i(x) = \sum_{\pi \in S(i)} a_{i,\pi}\, x^{\pi}$.

So $f_0^* = \inf_y \sum_{\pi} a_{0,\pi} y_{\pi}$, over all K-moment vectors y;

(y is a K-moment if there is a measure μ over K with $y_{\pi} = \mathbb{E}_{\mu} x^{\pi}$ for each tuple π)

$(K \doteq \{x \in \mathbb{R}^n \;:\; f_i(x) \geq 0, \; 1 \leq i \leq m\})$.

So: $y_0 = 1$. Define $v = (x^{\pi})$ (all monomials).

Also define $M[y] \doteq E_{\mu} v v^T$.

So for any tuples π, ρ, $M[y]_{\pi,\rho} = \mathbb{E}_{\nu}\, x^{\pi} x^{\rho} = E_{\nu} x^{\pi+\rho} = y_{\pi+\rho}$

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

where $f_i(x) = \sum_{\pi \in S(i)} a_{i,\pi}\, x^\pi$.

So $f_0^* = \inf_y \sum_\pi a_{0,\pi} y_\pi$, over all K-moment vectors y;

(y is a K-moment if there is a measure μ over K with $y_\pi = \mathbb{E}_\mu x^\pi$ for each tuple π)

$(K \doteq \{x \in \mathbb{R}^n \,:\, f_i(x) \ge 0,\; 1 \le i \le m\})$.

So: $y_0 = 1$. Can we say more? Define $v = (x^\pi)$ (all monomials). Also define $M[y] \doteq E_\mu v v^T$.

So for any tuples π, ρ, $M[y]_{\pi,\rho} = \mathbb{E}_\nu\, x^\pi x^\rho = E_\nu x^{\pi+\rho} = y_{\pi+\rho}$

So for any (∞-dimensional) vector z, **indexed by tuples**, i.e. with entries z_π for each tuple π,

$$z^T M[y] z = \sum_{\pi,\rho} \mathbb{E}_\mu\, z_\pi x^\pi x^\rho z_\rho = \mathbb{E}_\mu \left(\sum_\pi z_\pi x^\pi\right)^2 \ge 0$$

so $M[y] \succeq 0$!!

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

where $f_i(x) = \sum_{\pi \in S(i)} a_{i,\pi}\, x^\pi$.

So $f_0^* = \inf_y \sum_\pi a_{0,\pi} y_\pi$, over all K-moment vectors y;

(y is a K-moment if there is a measure μ over K with $y_\pi = \mathbb{E}_\mu x^\pi$ for each tuple π)

$(K \doteq \{x \in \mathbb{R}^n \,:\, f_i(x) \geq 0,\ 1 \leq i \leq m\})$.

So: $y_0 = 1$. Can we say more? Define $v = (x^\pi)$ (all monomials). Also define $M[y] \doteq E_\mu v v^T$.

So for any tuples π, ρ, $M[y]_{\pi,\rho} = \mathbb{E}_\nu\, x^\pi x^\rho = E_\nu x^{\pi+\rho} = y_{\pi+\rho}$

So for any (∞-dimensional) vector z, **indexed by tuples**, i.e. with entries z_π for each tuple π,

$$z^T M[y] z = \sum_{\pi,\rho} \mathbb{E}_\mu\, z_\pi x^\pi x^\rho z_\rho = \mathbb{E}_\mu \left(\sum_\pi z_\pi x^\pi\right)^2 \geq 0$$

$$\textbf{so } M[y] \succeq 0 \ !!$$

so

$$\begin{aligned} f_0^* \ \geq \ & \min \sum_\pi a_{0,\pi}\, y_\pi \\ \text{s.t.} \quad & y_0 = 1, \quad M \succeq 0, \quad M_{\pi,\rho} = y_{\pi+\rho}, \quad \text{for all tuples } \pi, \rho \\ & \text{the zeroth row and column of } M \text{ both equal } y \text{ (redundant)} \\ & \sum_\pi a_{i,\pi} y_\pi \geq 0 \quad 1 \leq i \leq m \end{aligned}$$

Cough! An **infinite-dimensional** semidefinite program!!

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

where $\boldsymbol{f_i(x) = \sum_{\pi \in S(i)} a_{i,\pi}\, x^{\pi}}$.

$$\begin{aligned} \boldsymbol{f_0^*} \;\geq\; & \min \sum_{\pi} a_{0,\pi}\, y_{\pi} \\ \text{s.t.} \quad & y_0 = 1, \quad M \succeq 0, \quad M_{\pi,\rho} = y_{\pi+\rho}, \quad \text{for all tuples } \pi, \rho \\ & \sum_{\pi} a_{i,\pi} y_{\pi} \geq 0 \quad 1 \leq i \leq m \end{aligned}$$

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

where $f_i(x) = \sum_{\pi \in S(i)} a_{i,\pi}\, x^{\pi}$.

$$\begin{aligned} f_0^* \ \geq \ & \min \sum_{\pi} a_{0,\pi}\, y_\pi \\ \text{s.t.} \quad & y_0 = 1, \quad M \succeq 0, \quad M_{\pi,\rho} = y_{\pi+\rho}, \quad \text{for all tuples } \pi, \rho \\ & \sum_{\pi} a_{i,\pi} y_\pi \geq 0 \quad 1 \leq i \leq m \end{aligned}$$

Restrict: pick an integer $d \geq 1$. Restrict the SDP to all tuples π with $|\pi| \leq d$.

Example: $d = 8$. So we will consider the monomial $x_1^2\, x_2^4\, x_3$ because $2 + 4 + 1 \leq 8$.

But we will not consider $x_3 x_5^7 x_8$, because $1 + 7 + 1 > 8$.

Restricted (level-d) relaxation (Lasserre):

$$\begin{aligned} & \min \sum_{\pi} a_{0,\pi}\, y_\pi \\ \text{s.t.} \quad & y_0 = 1, \quad M \succeq 0, \quad M_{\pi,\rho} = y_{\pi+\rho}, \quad \text{for all tuples } \pi, \rho \\ & \sum_{\pi} a_{i,\pi} y_\pi \geq 0 \quad 1 \leq i \leq m \\ & \text{the rows and columns of } M \text{, and the entries in } y \text{, indexed by tuples of size } \leq d \end{aligned}$$

A **finite-dimensional** semidefinite program!! But could be very large!!

For $\mathbf{d = 2}$ we get the standard semidefinite relaxation.