Optimization Fundamentals of OPF Problems

Daniel Bienstock, Columbia University

January 2015
Power flow problem in its simplest form
Power flow problem in its simplest form

Parameters:

• For each line $km$, its admittance $b_{km} + jg_{km} = b_{mk} + jg_{mk}$

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

• For each bus $k$, active and reactive net power limits $P_k^{\text{min}}, P_k^{\text{max}}, Q_k^{\text{min}}, \text{ and } Q_k^{\text{max}}$

Variables:

• For each bus $k$, complex voltage $e_k + jf_k$

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

Basic problem
Find a solution to:

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

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

\[(V_k^{\text{min}})^2 \leq e_k^2 + f_k^2 \leq (V_k^{\text{max}})^2,\]

for each bus \( k = 1, 2, \ldots \)

Many possible variations

- Line limits
- Various optimization versions
Quadratically constrained, quadratic programming problems (QCQPs):

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

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
Quadratically constrained, quadratic programming problems (QCQPs):

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

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

\[
x^T M x = x^T M^T x, \text{ so } x^T M x = \frac{1}{2}(x^T M x + x^T M^T x) = x^T (M + M^T) x
\]
Quadratically constrained, quadratic programming problems (QCQPs):

\[
\min \ f_0(x) \\
\text{s.t.} \quad f_i(x) \geq 0, \quad 1 \leq i \leq m \\
\quad x \in \mathbb{R}^n
\]

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

**Special case: Linear Programming**

\[
\begin{align*}
\min \quad & c^T x \\
\text{s.t.} \quad & Ax \geq b, \\
\quad & x \in \mathbb{R}^n
\end{align*}
\]
Quadratically constrained, quadratic programming problems (QCQPs):

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

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

Special case: Convex Quadratic Programming:

\[ M_0 \succeq 0, \quad M_i \preceq 0, \quad 1 \leq i \leq m \]
Folklore result: QCQP is NP-hard
Folklore result: QCQP is NP-hard

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

$$W^* \doteq \min - \sum_{i} x_i^2$$

s.t. $\sum_{i} w_i x_i = 0$,

$$-1 \leq x_i \leq 1, \quad 1 \leq i \leq n.$$
Folklore result: QCQP is NP-hard

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

\[
W^* \triangleq \min - \sum_{i} x_i^2 \\
\text{s.t. } \sum_{i} w_i x_i = 0, \\
-1 \leq x_i \leq 1, \; 1 \leq i \leq n.
\]

\( W^* = -n \), 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 \textbf{integers}, and consider:

$$W^* = \min \left\{ -\sum_i x_i^2 \quad \right. \left. \begin{array}{l} \text{s.t.} \sum_i w_i x_i = 0, \\ -1 \leq x_i \leq 1, \ 1 \leq i \leq n. \end{array} \right. $$

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

$$\sum_{j \in J} w_j = \sum_{j \notin J} w_j$$
• Lavaei & Low (2011), van Hentenryck & Coffrin (2014): AC-OPF is NP-hard on trees
Folklore result: QCQP is NP-hard

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

$$W^* \equiv \min \; - \sum_{i} x_i^2$$

s.t. $\sum_{i} w_i x_i = 0,$

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

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

$$\sum_{j \in J} w_j = \sum_{j \notin J} w_j$$
Actually, exactly what are “NP-hard” problems?
Folklore result: QCQP is NP-hard

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

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

s.t. $\sum_i w_i x_i = 0$, $-1 \leq x_i \leq 1, \ 1 \leq i \leq n$.

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

$$\sum_{j \in J} w_j = \sum_{j \notin J} w_j$$
Actually, exactly what are “NP-hard” problems?

• Really, really hard problems?
Folklore result: QCQP is NP-hard

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

$$
W^* \triangleq \min - \sum_{i} x_i^2
$$

s.t. \hspace{1em}
$$
\sum_{i} w_i x_i = 0,
$$

$$
-1 \leq x_i \leq 1, \hspace{0.5em} 1 \leq i \leq n.
$$

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

$$
\sum_{j \in J} w_j = \sum_{j \notin J} w_j
$$
Actually, exactly what are “NP-hard” problems?

- Really, really hard problems?

- But it is really, really hard to say exactly how they are hard?
Digression: NP-hardness

“Turing Machine” model (bit model) of computing

Programs = algorithms use 1-dimensional memory ("tape") where 0s and 1s are stored

• In one step, head reads/writes and/or moves one unit
Digression: NP-hardness

“Turing Machine” model (bit model) of computing

Programs = algorithms use 1-dimensional memory (“tape”) where 0s and 1s are stored

• In one step, head reads/writes and/or moves one unit
• Yes/no problems (like integer partition)
Digression: NP-hardness

“Turing Machine” model (bit model) of computing

Programs = algorithms use 1-dimensional memory (“tape”) where 0s and 1s are stored

- In one step, head reads/writes and/or moves one unit
• Yes/no problems (like integer partition)

• A problem is in the class $\text{NP}$ if for any “YES” instance there exists a verification in time polynomial in the number of bits needed to describe the instance
Digression: NP-hardness

“Turing Machine” model (bit model) of computing

Programs = algorithms use 1-dimensional memory (“tape”) where 0s and 1s are stored

• In one step, head reads/writes and/or moves one unit
• Yes/no problems (like integer partition)

• A problem is in the class ${\mathbf{NP}}$ if for any “YES” instance there exists a verification in time polynomial in the number of bits needed to describe the instance

**Example:** given integers $w_1, w_2, \ldots, w_n$, does there exist a subset $J$ with

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

**Note:** We are simply verifying a certificate that somebody gave us
Digression: NP-hardness

“Turing Machine” model (bit model) of computing

Programs = algorithms use 1-dimensional memory (“tape”) where 0s and 1s are stored

- In one step, head reads/writes and/or moves one unit
• Yes/no problems (like integer partition)

• A problem is in the class NP if for any “YES” instance there exists a verification in time polynomial in the number of bits needed to describe the instance

**Example:** given integers \(w_1, w_2, \ldots, w_n\), does there exist a subset \(J\) with

\[
\sum_{j \in J} w_j = \sum_{j \notin J} w_j
\]

**Note:** We are simply verifying a certificate that somebody gave us

**Proxy concept:** problems in NP are “well-defined”
“Turing Machine” model (bit model) of computing

Programs = algorithms use 1-dimensional memory (“tape”) where 0s and 1s are stored

- In one step, head reads/writes and/or moves one unit
• Yes/no problems (like integer partition)

• A problem is in the class \( \text{NP} \) if for any “YES” instance there exists a verification in time polynomial in the number of bits needed to describe the instance

**Example:** given integers \( w_1, w_2, \ldots, w_n \), does there exist a subset \( J \) with

\[
\sum_{j \in J} w_j = \sum_{j \notin J} w_j
\]

**Note:** We are simply verifying a certificate that somebody gave us

**Proxy concept:** problems in \( \text{NP} \) are “well-defined”

• A problem class \( \mathcal{P} \) is **NP-complete** if any problem in the class \( \text{NP} \) can be reduced to a problem \( \mathcal{P} \) in polynomial time
Digression: NP-hardness

“Turing Machine” model (bit model) of computing

Programs = algorithms use 1-dimensional memory (“tape”) where 0s and 1s are stored

• In one step, head reads/writes and/or moves one unit
• Yes/no problems (like integer partition)

• A problem is in the class \textbf{NP} if for any “YES” instance there exists a verification in time polynomial in the number of bits needed to describe the instance

\textbf{Example}: given integers \(w_1, w_2, \ldots, w_n\), does there exist a subset \(J\) with

\[
\sum_{j \in J} w_j = \sum_{j \notin J} w_j
\]

\textbf{Note}: We are simply \textbf{verifying} a certificate that somebody gave us

\textbf{Proxy concept}: problems in \textbf{NP} are “well-defined”

• A problem class \(\mathcal{P}\) is \textbf{NP-complete} if any problem in the class \textbf{NP} can be reduced to a problem \(\mathcal{P}\) in polynomial time

• As a consequence, if somebody smart figured a way to solve \(\mathcal{P}\) in poly-
nomial time, we can then solve every problem in \( \text{NP} \) in polynomial time
Digression: NP-hardness

“Turing Machine” model (bit model) of computing

Programs = algorithms use 1-dimensional memory (“tape”) where 0s and 1s are stored

- In one step, head reads/writes and/or moves one unit
• Yes/no problems (like integer partition)

• A problem is in the class \( \textbf{NP} \) if for any “YES” instance there exists a verification in time polynomial in the number of bits needed to describe the instance.

**Example:** given integers \( w_1, w_2, \ldots, w_n \), does there exist a subset \( J \) with

\[
\sum_{j \in J} w_j = \sum_{j \notin J} w_j
\]

**Note:** We are simply verifying a certificate that somebody gave us

**Proxy concept:** problems in \( \textbf{NP} \) are “well-defined”

• A problem class \( \mathcal{P} \) is \textbf{NP-complete} if any problem in the class \( \textbf{NP} \)
  can be reduced to a problem \( \mathcal{P} \) in polynomial time

• As a consequence, if somebody smart figured a way to solve \( \mathcal{P} \) in poly-
nomial time, we can then solve every problem in NP in polynomial time.

- Examples: traveling salesman problem, 3-SAT, graph coloring, the problem above
But ... not all NP-hard problems are equally hard

Again: given integers $w_1, w_2, \ldots, w_n$, does there exist a subset $J$ with

$$
\sum_{j \in J} w_j = \sum_{j \notin J} w_j = \frac{1}{2} \sum_{j=1}^{n} w_j
$$

It is **hard** (NP-hard) to answer YES, or NO, exactly
But ... not all NP-hard problems are equally hard

**Again:** given integers $w_1, w_2, \ldots, w_n$, does there exist a subset $J$ with

$$
\sum_{j \in J} w_j = \sum_{j \notin J} w_j = \frac{1}{2} \sum_{j=1}^{n} w_j
$$

It is **hard** (NP-hard) to answer YES, or NO, exactly

It is **not** hard to answer YES or NO, approximately:
But ... not all NP-hard problems are equally hard

Again: given integers $w_1, w_2, \ldots, w_n$, does there exist a subset $J$ with

\[
\sum_{j \in J} w_j = \sum_{j \notin J} w_j = \frac{1}{2} \sum_{j=1}^{n} w_j
\]

It is hard (NP-hard) to answer YES, or NO, exactly

It is not hard to answer YES or NO, approximately:
Fix $0 < \epsilon < 1$. Then we can compute a set $J$ such that

$$\frac{1 - \epsilon}{2} \sum_{j=1}^{n} w_j \leq \sum_{j \in J} w_j \leq \frac{1 + \epsilon}{2} \sum_{j=1}^{n} w_j$$

• In time polynomial in $n$ and $\epsilon^{-1}$

(So approximate feasibility, in “practicable” time)

Problem is \textit{weakly} NP-hard
Folklore result: QCQP is NP-hard

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

$$W^* = \min - \sum_i x_i^2$$

s.t. $\sum_i w_i x_i = 0$,

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

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

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

- Lavaei & Low (2011), van Hentenryck & Coffrin (2014): AC-OPF is NP-
hard on trees
Take any \( \{-1, 1\}\)-linear program
Take any \([-1, 1]\)-linear program

$$\min \ c^T x$$

s.t. \( Ax = b \)

\( x \in \{-1, 1\}^n. \)
Take any \([-1, 1]\)-linear program

\[
\begin{align*}
\text{min } \quad & c^T x \\
\text{s.t. } \quad & Ax = b \\
\text{with } \quad & x \in \{-1, 1\}^n.
\end{align*}
\]

\[
\begin{align*}
\text{min } \quad & c^T x - M \sum_j x_j^2 \\
\text{s.t. } \quad & Ax = b \\
\quad & -1 \leq x_j \leq 1, \quad 1 \leq j \leq n.
\end{align*}
\]

(and many other similar transformations)
Take any $\{-1, 1\}$-linear program

$$\min \ c^T x$$

s.t. \ $Ax = b$

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

$$\rightarrow$$

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

s.t. \ $Ax = b$

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

(and many other similar transformations)
\(\rightarrow\) linearly constrained QCQP is as hard as any integer programming problem.

**Example:** TSP, graph coloring, set covering, etc.

**NO** nice approximation algorithms exist for these.

They are called **strongly** NP-hard.
Take any \{-1, 1\}-linear program

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \in \{-1, 1\}^n.
\end{align*}
\]

\[
\begin{align*}
\min & \quad c^T x - M \sum_j x_j^2 \\
\text{s.t.} & \quad Ax = b \\
& \quad -1 \leq x_j \leq 1, \ 1 \leq j \leq n.
\end{align*}
\]

(and many other similar transformations)
→ linearly constrained QCQP is as hard as any integer programming problem

**Example:** TSP, graph coloring, set covering, etc.

**NO** nice approximation algorithms exist for these

They are called *strongly* NP-hard

Even more general than QCQP:

Solving systems of polynomial equations.

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. $p_i(x) = 0$, $\forall i$
Even more general than QCQP:

Solving systems of polynomial equations.

**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 $3v^6w - v^4 + 7 = 0$. 
Even more general than QCQP:

Solving systems of polynomial equations.

**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 \( c \):

\[
\begin{align*}
c^2 &= 1 \\
v^2 - cv_2 &= 0 \\
v_2^2 - cv_4 &= 0 \\
v_2v_4 - cv_6 &= 0 \\
v_6w - cy &= 0 \\
3cy - cv_4 &= -7
\end{align*}
\]

This is a polynomial-time reduction
Smale’s 17th problem

Can a zero of $n$ polynomial equations on $n$ unknowns be found **approximately**, **on the average** in polynomial time?

(abridged)
Smale’s 17th problem

Can a zero of $n$ polynomial equations on $n$ unknowns be found approximately, on the average in polynomial time?

(abridged)

What is meant by approximately?

And what do we mean by on the average?
Q: How do practitioners (e.g. power engineers) solve systems of nonlinear equations?
“Approximately”

Q: How do practitioners (e.g. power engineers) solve systems of nonlinear equations?

A: Newton-Raphson, of course!

→ If we start near a solution, quadratic convergence
“Approximately”

Q: How do practitioners (e.g. power engineers) solve systems of nonlinear equations?

A: Newton-Raphson, of course!

\[ F(x) = 0, \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^n \]

Iterate:
\[ x^{k+1} = - [J(x^k)]^{-1} F(x^k) + x^k, \quad k = 1, \ldots \]

\[ J(x^k)_{ij} = \frac{\partial J_i}{\partial x_j}(x^k) \quad 1 \leq i, j \leq n \]

→ If we start near a solution, quadratic convergence
Smale’s $17^{th}$ problem

Can a zero of $n$ polynomial equations on $n$ unknowns be found approximately, on the average in polynomial time?

(abridged)

What is meant by approximate convergence?
Smale’s 17\textsuperscript{th} problem

Can a zero of \(n\) polynomial equations on \(n\) unknowns be found approximately, on the average in polynomial time?

(abridged)

What is meant by approximate convergence?

\textbf{Answer:} convergence to the region of quadratic convergence for Newton-Raphson
Smale’s 17th problem

Can a zero of $n$ polynomial equations on $n$ unknowns be found approximately, on the average in polynomial time?

(abridged)

What is meant by approximate convergence?

**Answer:** convergence to the region of quadratic convergence for Newton-Raphson

What is meant by on the average?
Smale’s 17th problem

Can a zero of $n$ polynomial equations on $n$ unknowns be found approximately, on the average in polynomial time?

(abridged)

What is meant by approximate convergence?

**Answer:** convergence to the region of quadratic convergence for Newton-Raphson

What is meant by on the average?
“On the average” in polynomial time

A QCQP could be **quite** difficult!

e.g., a unique feasible solution, which additionally is an irrational vector

**Example** in $\mathbb{R}^2$:

\[
\begin{align*}
x_1^2 + x_2^2 & \geq 1 \\
x_1^2 + x_2^2 & \leq 1 \\
x_1 - x_2 & = 0 \\
x_1^2 + (x_2 - 1)^2 & \leq 1
\end{align*}
\]
“On the average” in polynomial time

A QCQP could be quite difficult!

e.g., a unique feasible solution, which additionally is an irrational vector

Example in $\mathbb{R}^2$:

\[
\begin{align*}
x_1^2 + x_2^2 &\geq 1 \\
x_1^2 + x_2^2 &\leq 1 \\
x_1 - x_2 &= 0 \\
x_1^2 + (x_2 - 1)^2 &\leq 1
\end{align*}
\]
“On the average” in polynomial time

A QCQP could be quite difficult!
e.g., a unique feasible solution, which additionally is an irrational vector

but a “nearby” problem instance could be much easier
“On the average” in polynomial time

A QCQP could be quite difficult!
e.g., a unique feasible solution, which additionally is an irrational vector

but a “nearby” problem instance could be much easier

• View a problem as a vector in an appropriate space
“On the average” in polynomial time

A QCQP could be quite difficult!
e.g., a unique feasible solution, which additionally is an irrational vector

but a “nearby” problem instance could be much easier

- View a problem as a vector in an appropriate space
- Endow that space with an appropriate metric
  (Bombieri-Weyl Hermitian product)
“On the average” in polynomial time

A QCQP could be quite difficult!
e.g., a unique feasible solution, which additionally is an irrational vector

but a “nearby” problem instance could be much easier

- View a problem as a vector in an appropriate space
- Endow that space with an appropriate metric
  (Bombieri-Weyl Hermitian product)
- In that space, uniformly sample a ball (of appropriate radius) around a given problem
“On the average” in polynomial time

A QCQP could be quite difficult!
e.g., a unique feasible solution, which additionally is an irrational vector

but a “nearby” problem instance could be much easier

- View a problem as a vector in an appropriate space
- Endow that space with an appropriate metric
  (Bombieri-Weyl Hermitian product)
- In that space, consider the set of problems given by a ball (of appropriate radius) around a given problem
- We want the algorithm to is fast, on average, in that ball
“On the average” in polynomial time

A QCQP could be quite difficult!
e.g., a unique feasible solution, which additionally is an irrational vector

but a “nearby” problem instance could be much easier

- View a problem as a vector in an appropriate space
- Endow that space with an appropriate metric  
  (Bombieri-Weyl Hermitian product)
- In that space, consider the set of problems given by a ball (of appropriate radius) around a given problem
- We want the algorithm to is fast, on average, in that ball

→ A Las Vegas algorithm:
“On the average” in polynomial time

A QCQP could be quite difficult!
e.g., a unique feasible solution, which additionally is an irrational vector

but a “nearby” problem instance could be much easier

- View a problem as a vector in an appropriate space
- Endow that space with an appropriate metric
  (Bombieri-Weyl Hermitian product)
- In that space, consider the set of problems given by a ball (of appropriate radius) around a given problem
- We want the algorithm to is fast, on average, in that ball

→ A Las Vegas algorithm: it may fail to converge, but with probability zero
Smale’s 17th problem

Can a zero of $n$ polynomial equations on $n$ unknowns be found approximately, on the average in polynomial time, with a uniform algorithm?

(abridged; but we are cheating)

- Beltrán and Pardo (2009) – a randomized (Las Vegas) uniform algorithm that computes an approximate zero in expected polynomial time

- Bürgisser, Cucker (2012) – a deterministic $O(n^{\log \log n})$ (uniform) algorithm for computing approximate zeros

**Techniques:** Homotopy (path-following method solving a sequence of problems), Newton’s method
Smale’s 17th problem

Can a zero of \( n \) polynomial equations on \( n \) unknowns be found approximately, on the average in polynomial time?

(abridged; and we are cheating)

• Beltrán and Pardo (2009) – a randomized (Las Vegas) uniform algorithm that computes an approximate zero in expected polynomial time

• Bürgisser, Cucker (2012) – a deterministic \( O(n^{\log \log n}) \) (uniform) algorithm for computing approximate zeros

• Techniques: Homotopy (path-following method solving a sequence of problems), Newton’s method

But we are cheating: All of this is over \( \mathbb{C}^n \), not \( \mathbb{R}^n \)
Smale’s 17th problem

Can a zero of $n$ polynomial equations on $n$ unknowns be found approximately, on the average in polynomial time?

(abridged; and we are cheating)

- Beltrán and Pardo (2009) – a randomized (Las Vegas) uniform algorithm that computes an approximate zero in expected polynomial time

- Bürgisser, Cucker (2012) – a deterministic $O(n^{\log \log n})$ (uniform) algorithm for computing approximate zeros

- **Techniques**: Homotopy (path-following method solving a sequence of problems), Newton’s method

**But we are cheating**: All of this is over $\mathbb{C}^n$, not $\mathbb{R}^n

**So what can be done over the reals?**
Smale’s 17th problem

Can a zero of $n$ polynomial equations on $n$ unknowns be found approximately, on the average in polynomial time?

(abridged; and we are cheating)

- Beltrán and Pardo (2009) – a randomized (Las Vegas) uniform algorithm that computes an approximate zero in expected polynomial time

- Bürgisser, Cucker (2012) – a deterministic $O(n^\log \log n)$ (uniform) algorithm for computing approximate zeros

- **Techniques:** Homotopy (path-following method solving a sequence of problems), Newton’s method

**But we are cheating:** All of this is over $\mathbb{C}^n$, not $\mathbb{R}^n

**So what can be done over the reals?** Let’s start with “simple” results.
Simplest example: S-Lemma (abridged)

Let \( f, g : \mathbb{R}^n \rightarrow \mathbb{R} \) be quadratic functions (degree \( \leq 2 \) polynomials).

Suppose there exists \( \bar{x} \in \mathbb{R}^n \) such that \( g(\bar{x}) > 0 \). Then

\[
f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0
\]

if and only

there exists \( \gamma \geq 0 \) such that

\[
f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.
\]

Yakubovich (1971), also much earlier, related work

\( \gamma \) acts as a Lagrange multiplier.
Quick aside:

Suppose we want to solve: $F^* \triangleq \min \{ f(x) : g(x) \geq 0 \}$; here $f, g$ quadratics
Quick aside:

Suppose we want to solve: \( \min \{ f(x) : g(x) \geq 0 \} \); here \( f, g \) quadratics

**Algorithm.** (Binary search)

1. Guess a real \( \theta \).

2. Check if \( f(x) - \theta \geq 0, \ \forall \ x \text{ s.t. } g(x) \geq 0 \).

3. If “yes”, we know \( F^* \geq \theta \); if not, \( F^* < \theta \).

4. Either way we can update \( \theta \), and repeat. Works under compactness of \( \{ x : g(x) \geq 0 \} \).
Simplest example: S-Lemma (abridged)

Let \( f, g : \mathbb{R}^n \to \mathbb{R} \) be quadratic functions (degree \( \leq 2 \) polynomials).

Suppose there exists \( \bar{x} \in \mathbb{R}^n \) such that \( g(\bar{x}) > 0 \). Then

\[
f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0
\]

if and only

\[
\text{there exists} \quad \gamma \geq 0 \quad \text{such that} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.
\]

Yakubovich (1971), also much earlier, related work

\( \gamma \) acts as a Lagrange multiplier.

Corollary: Can solve

\[
\min \{ f(x) : g(x) \geq 0 \}
\]

in polynomial time (using semidefinite programming)
Simplest example: S-Lemma (abridged)

Let \( f, g : \mathbb{R}^n \to \mathbb{R} \) be quadratic functions (degree \( \leq 2 \) polynomials).

Suppose there exists \( \bar{x} \in \mathbb{R}^n \) such that \( g(\bar{x}) > 0 \). Then

\[
f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0
\]

if and only

\[
\text{there exists} \quad \gamma \geq 0 \quad \text{such that} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.
\]

Yakubovich (1971), also much earlier, related work

\( \gamma \) acts as a Lagrange multiplier.

**Corollary:** Can solve

\[
\min \{ f(x) : g(x) \geq 0 \}
\]

in polynomial time (using semidefinite programming)

\( \rightarrow \) Time for some math
Want to solve: \( \min \{ f(x) : g(x) \geq 0 \} \)
Given a real $\theta$, is it the case that $f(x) - \theta \geq 0$ whenever $g(x) \geq 0$?
Given a real $\theta$, is it the case that $f(x) - \theta \geq 0$ whenever $g(x) \geq 0$?

**S-Lemma:** iff there exists real $\gamma \geq 0$ s.t. $f(x) - \theta - \gamma g(x) \geq 0 \ \forall \ x \in \mathbb{R}^n$
Given a real \( \theta \), is it the case that \( f(x) - \theta \geq 0 \) whenever \( g(x) \geq 0 \) ?

**S-Lemma:** iff there exists real \( \gamma \geq 0 \) s.t. \( f(x) - \theta - \gamma g(x) \geq 0 \) \( \forall x \in \mathbb{R}^n \)

**Notation:** \( f(x) = x^T A x + 2a^T x + a_0 \), \( g(x) = x^T B x + 2b^T x + b_0 \),
Given a real $\theta$, is it the case that $f(x) - \theta \geq 0$ whenever $g(x) \geq 0$?

S-Lemma: iff there exists real $\gamma \geq 0$ s.t. $f(x) - \theta - \gamma g(x) \geq 0 \quad \forall \ x \in \mathbb{R}^n$

Notation: $f(x) = x^T A x + 2a^T x + a_0$, $g(x) = x^T B x + 2b^T x + b_0$,

So S-Lemma statement is:

$$(x^T, 1) \begin{pmatrix} A - \gamma B & a - \gamma b \\ (a - \gamma b)^T & a_0 - \gamma b_0 - \theta \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} \geq 0 \quad \forall \ x \in \mathbb{R}^n$$
Given a real $\theta$, is it the case that $f(x) - \theta \geq 0$ whenever $g(x) \geq 0$?

**S-Lemma:** iff there exists real $\gamma \geq 0$ s.t. $f(x) - \theta - \gamma g(x) \geq 0 \ \forall \ x \in \mathbb{R}^n$

**Notation:** $f(x) = x^T A x + 2a^T x + a_0, \ g(x) = x^T B x + 2b^T x + b_0,$

So S-Lemma statement is:

$$(x^T, 1) \begin{pmatrix} A - \gamma B & a - \gamma b \\ (a - \gamma b)^T & a_0 - \gamma b_0 - \theta \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} \geq 0 \ \forall \ x \in \mathbb{R}^n$$

**Can be proved that** this is equivalent to saying:

$$\begin{pmatrix} A - \gamma B & a - \gamma b \\ (a - \gamma b)^T & a_0 - \gamma b_0 - \theta \end{pmatrix} \succeq 0$$
Given a real $\theta$, is it the case that $f(x) - \theta \geq 0$ whenever $g(x) \geq 0$?

**S-Lemma:** If there exists real $\gamma \geq 0$ s.t. $f(x) - \theta - \gamma g(x) \geq 0 \quad \forall \ x \in \mathbb{R}^n$

**Notation:** $f(x) = x^T Ax + 2a^T x + a_0, \ g(x) = x^T Bx + 2b^T x + b_0,$

So S-Lemma statement is:

$$(x^T, 1) \begin{pmatrix} A - \gamma B & a - \gamma b \\ (a - \gamma b)^T & a_0 - \gamma b_0 - \theta \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} \geq 0 \quad \forall \ x \in \mathbb{R}^n$$

**Can be proved that** this is equivalent to saying:

$$\begin{pmatrix} A - \gamma B & a - \gamma b \\ (a - \gamma b)^T & a_0 - \gamma b_0 - \theta \end{pmatrix} \succeq 0$$

So in short, $\min \{f(x) : g(x) \geq 0\}$ is equivalent to

$$\max \ \theta$$

subject to

$$\begin{pmatrix} A - \gamma B & a - \gamma b \\ (a - \gamma b)^T & a_0 - \gamma b_0 - \theta \end{pmatrix} \succeq 0$$

$$\gamma \geq 0$$

which is an SDP (semidefinite program) on variables $\gamma, \theta$.  

Many applications for the S-Lemma

- Control Theory
- Dynamical Systems
- Robust error estimation
- Robust optimization
- ...
An application: the trust-region subproblem

\[
\min \{ f(x) \mid g(x) \leq 0 \}
\]

can be solved in polynomial time, where \( f, g \) quadratics, \( g \) convex

Scale, rotate, translate:

\[
\min \{ f(x) \mid \|x\| \leq 1 \}
\]
Digression: application of trust-region subproblem in engineering

→ Unconstrained optimization $\min\{f(x) : x \in \mathbb{R}^n\}$

• $f(x)$ can be anything

• constraints mapped into $f(x)$ by using penalties
Digression: application of trust-region subproblem in engineering

→ Unconstrained optimization  \( \min \{ f(x) : x \in \mathbb{R}^n \} \)

• \( f(x) \) can be anything

• constraints mapped into \( f(x) \) by using penalties

Example: \( \min \{ g(x_1, x_2) : 0 \leq x_1 \leq 1 \} \)
Digression: application of trust-region subproblem in engineering

→ Unconstrained optimization \( \min \{ f(x) : x \in \mathbb{R}^n \} \)

- \( f(x) \) can be anything
- constraints mapped into \( f(x) \) by using penalties

Example: \( \min \{ g(x_1, x_2) : 1/2 \leq x_1 \text{ and } x_2 \leq 1 \} \)

becomes:

\[
\min g(x_1, x_2) + \alpha \log(x_1 - 1/2) + \alpha \log(1 - x_2)
\]

subject to: \( x_1, x_2 \) unconstrained

\( \alpha > 0 \) a “barrier” parameter
Digression: application of trust-region subproblem in engineering

→ Unconstrained optimization $\min\{f(x) : x \in \mathbb{R}^n\}$
Digression: application of trust-region subproblem in engineering

→ Unconstrained optimization \( \min \{ f(x) : x \in \mathbb{R}^n \} \)

**Algorithm**

- Given an iterate \( w^t \), sample \( f(x) \) in a neighborhood \( \|x - x^t\| \leq \Delta \).
- Get pairs \((y^1, f(y^1)), (y^2, f(y^2)), \ldots, (y^m, f(y^m))\)
- Using these samples, construct a **quadratic** “model” of \( f(x) \)
  (model = spline, least squares estimate, etc).
Digression: application of trust-region subproblem in engineering

→ Unconstrained optimization \( \min\{f(x) : x \in \mathbb{R}^n\} \)

**Algorithm**

- Given an iterate \( w^t \), sample \( f(x) \) in a neighborhood \( \|x - x^t\| \leq \Delta \).

- Get pairs \( (y^1, f(y^1)), (y^2, f(y^2)), \ldots, (y^m, f(y^m)) \)

- Using these samples, construct a **quadratic** “model” of \( f(x) \) (model = spline, least squares estimate, etc).

- Call this model: \( Q(x) \)

- **Solve:** \( \min\{ Q(x) : \|x - w^t\| \leq \Delta \} \). This is the trust-region subproblem.

- The solution becomes \( w^{t+1} \).
  Or: conduct a line-search from \( w^t \) to the solution so as to compute \( w^{t+1} \).
Digression: application of trust-region subproblem in engineering

→ Unconstrained optimization $\min\{f(x) : x \in \mathbb{R}^n\}$

Algorithm

• Given an iterate $w^t$, sample $f(x)$ in a neighborhood $\|x - x^t\| \leq \Delta$.

• Get pairs $(y^1, f(y^1)), (y^2, f(y^2)), \ldots, (y^m, f(y^m))$

• Using these samples, construct a quadratic “model” of $f(x)$
  (model = spline, least squares estimate, etc).

• Call this model: $Q(x)$

• Solve: $\min\{Q(x) : \|x - w^t\| \leq \Delta\}$. This is the trust-region subproblem.

• The solution becomes $w^{t+1}$.
  Or: conduct a line-search from $w^t$ to the solution so as to compute $w^{t+1}$.

• General purpose codes: KNITRO, LOQO have been used on OPF.
An application: the trust-region subproblem

\[
\min\{f(x) : g(x) \leq 0\}
\]

can be solved in polynomial time, where \( f, g \) quadratics, \( g \) convex.

Scale, rotate, translate:

\[
\min\{f(x) : \|x\| \leq 1\}
\]

can be solved in poly time \( \rightarrow \log \epsilon^{-1} \)

Y. Ye (1992) \( \rightarrow \log \log \epsilon^{-1} \)

How about extensions of the trust-region subproblem?
**Sturm-Zhang (2003)**

Where $f(x)$ is a quadratic,

$$
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad \|x\| \leq 1 \\
& \quad a^T x \leq b \quad (\text{one linear side constraint})
\end{align*}
$$

can be solved in polynomial time, as can

$$
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad \|x\| \leq 1 \\
& \quad \|x - x^0\| \leq r_0 \quad (\text{one additional convex ball constraint})
\end{align*}
$$

**Ye-Zhang (2003)**

$$
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad \|x\| \leq 1 \\
& \quad a_i^T x \leq b_i \quad i = 1, 2 \\
& \quad (a_1^T x - b_1)(a_2^T x - b_2) = 0
\end{align*}
$$

(two linear side constraints, but at least one binding)
Anstreicher-Burer (2012)

In polynomial time, one can solve a problem of the form

$$\min \ x^T Q x + c^T x$$

s.t.  $$\|x\| \leq 1$$

$$a_i^T x \leq b_i \quad i = 1, 2$$

provided the two linear constraints are parallel:

![Diagram showing parallel two linear constraints and a ball constraint]
Anstreicher-Burer (2012)

In polynomial time, one can solve a problem of the form

\[
\begin{align*}
\min & \quad x^T Q x + c^T x \\
\text{s.t.} & \quad \|x\| \leq 1 \\
& \quad a_i^T x \leq b_i \quad i = 1, 2
\end{align*}
\]

provided the two linear constraints are parallel:

\[
\rightarrow \min \left\{ x^T Q x + c^T x : l \leq x_1 \leq u, \|x\| \leq 1 \right\}
\]
In polynomial time, one can solve a problem of the form
\[
\min x^T Q x + c^T x \\
\text{s.t. } \|x\| \leq 1 \\
a_i^T x \leq b_i \quad i = 1, 2
\]
provided the two linear constraints are parallel:

\[
\rightarrow \min \{ x^T Q x + c^T x : l \leq x_1 \leq u, \|x\| \leq 1 \} \quad (*)
\]
restate as: \[
\min \sum_{i,j} q_{ij} X_{ij} + c^T x \\
\text{s.t. } X_{11} + lu \leq (l + u)x_1 \\
\|X_{1} - lx\| \leq x_1 - l \\
\|ux - X_{1}\| \leq u - x_1 \\
\sum_{j} X_{jj} \leq 1 \\
X \succeq xx^T
\]
Equivalent to problem (*)? Yes, if \( X = xx^T \), i.e. a rank-1 solution
Anstreicher-Burer (2012)

In polynomial time, one can solve a problem of the form

$$\min \ x^T Q x + c^T x$$

s.t.  

$$\|x\| \leq 1$$

$$a_i^T x \leq b_i \ i = 1, 2$$

provided the two linear constraints are parallel:

$$\rightarrow \min \ \{ x^T Q x + c^T x : l \leq x_1 \leq u, \|x\| \leq 1 \} \quad (*)$$

restate as:  

$$\min \ \sum_{i,j} q_{ij} X_{ij} + c^T x \rightarrow \sum_{i,j} q_{ij} x_i x_j + c^T x$$

s.t.  

$$X_{11} + lu \leq (l + u)x_1$$

$$\|X_{1} - lx\| \leq x_1 - l$$

$$\|ux - X_{1}\| \leq u - x_1$$

$$\sum_j X_{jj} \leq 1 \rightarrow \sum_j x_{jj}^2 \leq 1$$

$$X \succeq xx^T$$

Equivalent to problem (*)? Yes, if \( X = xx^T \), i.e. a rank-1 solution
Anstreicher-Burer (2012)

In polynomial time, one can solve a problem of the form

$$\min \ x^T Q x + c^T x$$

s.t. $$\|x\| \leq 1$$

$$a_i^T x \leq b_i \ i = 1, 2$$

provided the two linear constraints are parallel:

\[ \rightarrow \min \left\{ x^T Q x + c^T x : l \leq x_1 \leq u, \|x\| \leq 1 \right\} \quad (*) \]

restate as: \( \min \sum_{i,j} q_{ij} x_{ij} + c^T x \)

s.t. \( X_{11} + lu \leq (l+u)x_1 \)

\( \|X_{1} - lx\| \leq x_1 - l \)

\( \|ux - X_1\| \leq u - x_1 \)

\( \sum_j X_{jj} \leq 1 \)

\( X \succeq xx^T \)

Lemma: This problem has an optimal solution with \( X = xx^T \), i.e. a rank-1 solution.
Burer-Yang (2012)

In polynomial time, one can solve a problem of the form

$$\min \quad x^T Q x + c^T x$$

s.t.  \[ \|x\| \leq 1 \]

\[ a_i^T x \leq b_i \quad 1 \leq i \leq m \]

if no two linear inequalities are simultaneously binding in the feasible region
Burer-Yang (2012)

In polynomial time, one can solve a problem of the form

$$\min \ x^T Q x + c^T x$$

s.t.  \( \|x\| \leq 1 \)

$$a_i^T x \leq b_i \quad 1 \leq i \leq m$$

if no two linear inequalities are simultaneously binding in the feasible region

Lemma: the following problem has an optimal solution with \( X = xx^T \).

$$\min \ \sum_{i,j} q_{ij} X_{ij} \ + \ c^T x$$

s.t.  \( X_{11} + lu \leq (l + u)x_1 \)

\( \|b_i x - X a_i\| \leq b_i - a_i^T x \quad i \leq m \)

\( b_i b_j - b_j a_i^T x - b_i a_j^T x + a_i^T X a_j \leq 0 \quad i < j \leq m \)

\( \sum_j X_{jj} \leq 1 \quad X \succeq xx^T \)
Generalizations?

(B. and Alex Michalka, SODA 2014)

$$\min x^T Q x + c^T x$$

s.t. $$\|x - \mu_h\| \leq r_h, \; h \in S,$$
$$\|x - \mu_h\| \geq r_h, \; h \in K,$$
$$x \in P \doteq \{x \in \mathbb{R}^n : Ax \leq b\}$$

**Theorem.**
For each fixed $|S|$, $|K|$ can be solved in polynomial time if either

(1) $|S| \geq 1$ and polynomially large number of faces of $P$ intersect

$$\bigcap_{h \in S} \{x \in \mathbb{R}^n : \|x - \mu_h\| \leq r_h\},$$

or

(2) $|S| = 0$ and the number of rows of $A$ is bounded.
Generalizations?

(B. and Alex Michalka, SODA 2014)

\[
\begin{align*}
\min\ & x^T Q x + c^T x \\
\text{s.t.} \ & \| x - \mu_h \| \leq r_h, \quad h \in S, \\
\ & \| x - \mu_h \| \geq r_h, \quad h \in K, \\
\ & x \in P = \{ x \in \mathbb{R}^n : Ax \leq b \}
\end{align*}
\]

Theorem.
For each fixed $|S|$, $|K|$ can be solved in polynomial time if either

(1) $|S| \geq 1$ and polynomially large number of faces of $P$ intersect

\[
\bigcap_{h \in S} \{ x \in \mathbb{R}^n : \| x - \mu_h \| \leq r_h \},
\]

or

(2) $|S| = 0$ and the number of rows of $A$ is bounded.

- Does **not** use semidefinite programming
- **Note:** the curvature in all quadratics is the same
Why not general QCQP?
Why not general QCQP?

(QCQP): \[\begin{align*}
\text{min} & \quad x^T Q x + 2c^T x \\
\text{s.t.} & \quad x^T A_i x + 2b_i^T x + r_i \geq 0 \quad i = 1, \ldots, m \\
x & \in \mathbb{R}^n.
\end{align*}\]
Why not general QCQP?

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

\[\rightarrow \text{form the semidefinite relaxation}\]

\[(\text{SR}): \quad \min \ \left( \begin{array}{cc} 0 & c^T \\ c & Q \end{array} \right) \bullet X \]
\[\text{s.t.} \quad \left( \begin{array}{cc} r_i & b_i^T \\ b_i & A_i \end{array} \right) \bullet X \geq 0 \quad i = 1, \ldots, m \]
\[X \succeq 0, \quad X_{11} = 1.\]

Here, for symmetric matrices \( M, \ N, \)
\[M \bullet N = \sum_{h,k} M_{hk} N_{hk} \]
Why not general QCQP?

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

→ form the semidefinite relaxation

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

Here, for symmetric matrices \( M, \ N \),

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

Why do we call it a relaxation?
Why not general QCQP?

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

→ form the semidefinite relaxation

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

Here, for symmetric matrices \( M, N \),

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

Why do we call it a relaxation?

Given \( x \) feasible for \( QCQP \), the matrix \( (1, x^T) \begin{pmatrix} 1 \\ x \end{pmatrix} \) feasible for \( SR \) and with the same value
Why not general QCQP?

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

→ form the semidefinite relaxation

\[
\begin{align*}
\text{(SR):} & \quad \min \ \begin{pmatrix} 0 & c^T \\ c & Q \end{pmatrix} \cdot X \\
& \text{s.t.} \quad \begin{pmatrix} r_i & b_i^T \\ b_i & A_i \end{pmatrix} \cdot X \geq 0 \quad i = 1, \ldots, m \\
& \quad X \succeq 0, \quad X_{11} = 1.
\end{align*}
\]

Here, for symmetric matrices \( M, \ N, \)

\[
M \cdot N = \sum_{h,k} M_{hk} N_{hk}
\]

Why do we call it a relaxation?

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

So the value of problem SR is a lower bound for QCQP
Why not general QCQP?

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

→ form the semidefinite relaxation

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

Here, for symmetric matrices \( M, N \),
\[ M \bullet N = \sum_{h,k} M_{hk} N_{hk} \]

Why do we call it a relaxation?

Given \( x \) feasible for QCQP, the matrix \( (1, x^T) \begin{pmatrix} 1 \\ x \end{pmatrix} \) 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.
Why not general QCQP?

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

\(\rightarrow\) form the semidefinite relaxation

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

Here, for symmetric matrices \( M, \ N, \)

\[M \cdot N = \sum_{h,k} M_{hk}N_{hk}\]

Why do we call it a relaxation?

Given \( x \) feasible for QCQP, the matrix \( \begin{pmatrix} 1, x^T \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} \) 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

(SR): \[ \begin{align*} 
\text{min} & \quad M \cdot X \\
\text{s.t.} & \quad N^i \cdot X \geq b_i \quad i = 1, \ldots, m \\
& \quad X \succeq 0, \quad X \text{ an } n \times n \text{ matrix},
\end{align*} \]

always has a solution of rank \( O(m^{1/2}) \), and this result is best possible.
Generalizations?

(B. and Alex Michalka, SODA 2014)

\[
\begin{align*}
\min & \quad x^T Q x + c^T x \\
\text{s.t.} & \quad \|x - \mu_h\| \leq r_h, \quad h \in S, \\
& \quad \|x - \mu_h\| \geq r_h, \quad h \in K, \\
& \quad x \in P \doteq \{ x \in \mathbb{R}^n : Ax \leq b \}
\end{align*}
\]

**Theorem.**
For each fixed \(|S|, |K|\) can be solved in polynomial time if either

(1) \(|S| \geq 1 \) and polynomially large number of faces of \(P\) intersect
\[
\bigcap_{h \in S} \{ x \in \mathbb{R}^n : \|x - \mu_h\| \leq r_h \},
\]
or

(2) \(|S| = 0 \) and the number of rows of \(A\) is bounded.

• Does **not** use semidefinite programming

• **Note:** the curvature in all quadratics is the same
The trust-region subproblem:

\[
\begin{align*}
\min & \quad x^T Q x + c^T x \\
\text{s.t.} & \quad \| x - \mu \| \leq r
\end{align*}
\]
The trust-region subproblem:

\[
\begin{align*}
\min & \quad x^T Q x + c^T x \\
\text{s.t.} & \quad \|x - \mu\| \leq r
\end{align*}
\]

Generalization: CDT (Celis-Dennis-Tapia) problem

\[
\begin{align*}
\min & \quad x^T Q_0 x + c_0^T x \\
\text{s.t.} & \quad x^T Q_1 x + c_1^T x + d_1 \leq 0 \\
& \quad x^T Q_2 x + c_2^T x + d_2 \leq 0
\end{align*}
\]

where \( Q_1 \succ 0, \ Q_2 \succ 0 \)
Even more general than QCQPs

Barvinok (STOC 1992):

For each fixed $p \geq 1$, there is a polynomial-time algorithm for deciding feasibility of a system

\[
x^T M_i x = 0, \quad 1 \leq i \leq p, \\
\|x\| = 1, \quad x \in \mathbb{R}^n
\]

where the $M_i$ are general matrices.
Even more general than QCQPs

Barvinok (STOC 1992):

For each fixed $p \geq 1$, there is a polynomial-time algorithm for deciding feasibility of a system

\[ x^T M_i x = 0, \quad 1 \leq i \leq p, \]
\[ \|x\| = 1, \quad x \in \mathbb{R}^n \]

where the $M_i$ are general matrices.

- **Non-constructive.** Algorithm says “yes” or “no.”

- **Computational model?**

  Stated as: computation over the reals using infinite precision
Even more general than QCQPs

Barvinok (STOC 1992):

For each fixed $p \geq 1$, there is a polynomial-time algorithm for deciding feasibility of a system

$$x^T M_i x = 0, \quad 1 \leq i \leq p,$$

$$\|x\| = 1, \quad x \in \mathbb{R}^n$$

where the $M_i$ are general matrices.

- **Non-constructive.** Algorithm says “yes” or “no.”

- **Computational model?**

  Stated as: computation over the reals using infinite precision

- There is a separate community in mathematics dealing with these problems
- Methodology does **not** use semidefinite programming
- Instead, uses algebraic geometry
- Explicit emphasis in handling “cases”
A (better?) alternative: $\epsilon$-feasibility

For each fixed $p \geq 1$, given a system

$$x^T M_i x = 0, \quad 1 \leq i \leq p,$$
$$\|x\| = 1, \quad x \in \mathbb{R}^n$$

and given $0 < \epsilon < 1$, either

- **Prove** that the system is infeasible, or

- **Output** $\hat{x} \in \mathbb{R}^n$ with

$$-\epsilon \leq x^T M_i \leq \epsilon, \quad 1 \leq i \leq p,$$
$$1 - \epsilon \leq \|\hat{x}\| \leq 1 + \epsilon,$$

in time polynomial in the data and in $\log \epsilon^{-1}$. 
A (better?) alternative: $\epsilon$-feasibility

For each fixed $p \geq 1$, given a system

$$x^T M_i x = 0, \quad 1 \leq i \leq p,$$
$$\|x\| = 1, \quad x \in \mathbb{R}^n$$

and given $0 < \epsilon < 1$, either

- **Prove** that the system is infeasible, or

- **Output** $\hat{x} \in \mathbb{R}^n$ with

  $$-\epsilon \leq x^T M_i \leq \epsilon, \quad 1 \leq i \leq p,$$
  $$1 - \epsilon \leq \|\hat{x}\| \leq 1 + \epsilon,$$

  in time polynomial in the data and in $\log \epsilon^{-1}$.

**Two issues:** Constructiveness, and $\epsilon$-feasibility
Modification to Barvinok’s result

Assume that for each fixed $p \geq 1$, there is an algorithm that given a system

$\begin{align*}
    x^T M_i x &= 0, \quad 1 \leq i \leq p, \\
    \|x\| &= 1, \quad x \in \mathbb{R}^n
\end{align*}$

and given $0 < \epsilon < 1$, either

- **Proves** that the system is **infeasible**, or

- **Proves** that is $\epsilon$-feasible,

in time polynomial in the data and in $\log \epsilon^{-1}$.

(so still nonconstructive)
Modification to Barvinok’s result

Assume that for each fixed \( p \geq 1 \), there is an algorithm that given a system

\[
x^T M_i x = 0, \quad 1 \leq i \leq p,
\]
\[
\|x\| = 1, \quad x \in \mathbb{R}^n
\]

and given \( 0 < \epsilon < 1 \), either

- Proves that the system is infeasible, or

- Proves that is \( \epsilon \)-feasible,

in time polynomial in the data and in \( \log \epsilon^{-1} \).

(so still nonconstructive)

Assuming such an algorithm exists ...
Theorem.

For each fixed $m \geq 1$ there is a polynomial-time algorithm that, given an optimization problem

$$\min f_0(x) \triangleq x^TQ_0x + c_0^Tx$$

s.t. $x^TQ_ix + c_i^Tx + d_i \leq 0 \quad 1 \leq i \leq m,$

where $Q_1 \succ 0$, and $0 < \epsilon < 1$, either

(1) proves that the problem is infeasible,

or

(2) computes an $\epsilon$-feasible vector $\hat{x}$ such that there exists no feasible $x \in \mathbb{R}^n$ with $f_0(x) < f(\hat{x}) - \epsilon$.

The complexity of the algorithm is polynomial in the number of bits in the data and in $\log \epsilon^{-1}$.
Theorem.

For each fixed \( m \geq 1 \) there is a polynomial-time algorithm that, given an optimization problem

\[
\begin{align*}
\min & \quad f_0(x) = x^T Q_0 x + c_0^T x \\
\text{s.t.} & \quad x^T Q_i x + c_i^T x + d_i \leq 0 \quad 1 \leq i \leq m,
\end{align*}
\]

where \( Q_1 \succ 0 \), and \( 0 < \epsilon < 1 \), either

(1) proves that the problem is infeasible,

or

(2) computes an \( \epsilon \)-feasible vector \( \hat{x} \) such that there exists no feasible \( x \in \mathbb{R}^n \) with \( f_0(x) < f(\hat{x}) - \epsilon \).

The complexity of the algorithm is polynomial in the number of bits in the data and in \( \log \epsilon^{-1} \)

→ Related algebraic geometry work by Grigoriev, Pasechnik, other Russians
Let $f, g : \mathbb{R}^n \to \mathbb{R}$ be quadratic functions (degree $\leq 2$ polynomials).

Suppose there exists $\bar{x} \in \mathbb{R}^n$ such that $g(\bar{x}) > 0$. Then

$$f(x) \geq 0 \text{ whenever } g(x) \geq 0 \text{ iff exists } \gamma \geq 0 \text{ s.t. } f(x) \geq \gamma g(x) \text{ for all } x \in \mathbb{R}^n.$$
Back to S-Lemma, +

Let \( f, g : \mathbb{R}^n \to \mathbb{R} \) be quadratic functions (degree \( \leq 2 \) polynomials).

Suppose there exists \( \bar{x} \in \mathbb{R}^n \) such that \( g(\bar{x}) > 0 \). Then

\[
  f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0 \quad \text{iff exists} \quad \gamma \geq 0 \quad \text{s.t.} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.
\]
i.e., iff exists \( \gamma \geq 0 \) s.t. \( (f - \gamma g)(x) \geq 0 \) for all \( x \in \mathbb{R}^n \).
Let $f, g : \mathbb{R}^n \to \mathbb{R}$ be quadratic functions (degree $\leq 2$ polynomials).

Suppose there exists $\bar{x} \in \mathbb{R}^n$ such that $g(\bar{x}) > 0$. Then

$$f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0 \quad \text{iff} \quad \exists \gamma \geq 0 \quad \text{s.t.} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.$$  

i.e., iff exists $\gamma \geq 0$ s.t. $(f - \gamma g)(x) \geq 0$ for all $x \in \mathbb{R}^n$.

in other words, Hilbert (1888): iff exists $\gamma \geq 0, S_0(x)$ s.t. $f(x) = S_0(x) + \gamma g(x)$

where $S_0(x)$ is a sum of squares of polynomials
Back to S-Lemma, +

Let \( f, g : \mathbb{R}^n \to \mathbb{R} \) be quadratic functions (degree \( \leq 2 \) polynomials).

Suppose there exists \( \bar{x} \in \mathbb{R}^n \) such that \( g(\bar{x}) > 0 \). Then

\[
f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0 \quad \iff \quad \exists \gamma \geq 0 \quad \text{s.t.} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.
\]

i.e., iff exists \( \gamma \geq 0 \) s.t. \( (f - \gamma g)(x) \geq 0 \) for all \( x \in \mathbb{R}^n \).

in other words, **Hilbert (1888):** iff exists \( \gamma \geq 0, \ S_0(x) \quad \text{s.t.} \quad f(x) = S_0(x) + \gamma g(x) \)

where \( S_0(x) \) is a **sum of squares of polynomials**! This paper started the field of algebraic geometry.
Back to S-Lemma, +

Let $f, g : \mathbb{R}^n \to \mathbb{R}$ be quadratic functions (degree $\leq 2$ polynomials).

Suppose there exists $\bar{x} \in \mathbb{R}^n$ such that $g(\bar{x}) > 0$. Then

$$f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0 \quad \text{iff exists} \quad \gamma \geq 0 \quad \text{s.t.} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.$$ 

i.e., iff exists $\gamma \geq 0$ s.t. $(f - \gamma g)(x) \geq 0$ for all $x \in \mathbb{R}^n$.

in other words, Hilbert (1888): iff exists $\gamma \geq 0$, $S_0(x)$ s.t. $f(x) = S_0(x) + \gamma g(x)$

where $S_0(x)$ is a sum of squares of polynomials! This paper started the field of algebraic geometry.

And how about generalizations of the S-Lemma?

Given quadratics $Q_0(x), Q_1(x), \ldots, Q_m(x)$ with $m \geq 2$, is it true that

$$Q_0(x) \geq 0 \quad \text{whenever} \quad Q_i(x) \geq 0, \quad 1 \leq i \leq m,$$

iff exist $\gamma_i \geq 0$ s.t. $Q_0(x) \geq \sum_{i=1}^m \gamma_i Q_i(x)$ for all $x \in \mathbb{R}^n$?
Back to S-Lemma, +

Let \( f, g : \mathbb{R}^n \to \mathbb{R} \) be quadratic functions (degree \( \leq 2 \) polynomials).

Suppose there exists \( \bar{x} \in \mathbb{R}^n \) such that \( g(\bar{x}) > 0 \). Then

\[
f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0 \quad \text{iff exists} \quad \gamma \geq 0 \quad \text{s.t.} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.
\]

i.e., iff exists \( \gamma \geq 0 \) s.t. \((f - \gamma g)(x) \geq 0\) for all \( x \in \mathbb{R}^n \).

in other words, Hilbert (1888): iff exists \( \gamma \geq 0, \ S_0(x) \) s.t. \( f(x) = S_0(x) + \gamma g(x) \)

where \( S_0(x) \) is a sum of squares of polynomials! This paper started the field of algebraic geometry

And how about generalizations of the S-Lemma?

Given quadratics \( Q_0(x), Q_1(x), \ldots, Q_m(x) \) with \( m \geq 2 \), is it true that

\[
Q_0(x) \geq 0 \quad \text{whenever} \quad Q_i(x) \geq 0, \ 1 \leq i \leq m,
\]

iff exist \( \gamma_i \geq 0, \ S_0(x), \) s.t. \( Q_0(x) = S_0(x) + \sum_{i=1}^{m} \gamma_i Q_i(x), \ S_0(x) = \) a sum of squares?
Back to S-Lemma, +

Let \( f, g : \mathbb{R}^n \rightarrow \mathbb{R} \) be quadratic functions (degree \( \leq 2 \) polynomials).

Suppose there exists \( \bar{x} \in \mathbb{R}^n \) such that \( g(\bar{x}) > 0 \). Then

\[
f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0 \quad \text{iff exists} \quad \gamma \geq 0 \quad \text{s.t.} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.
\]

i.e., iff exists \( \gamma \geq 0 \) s.t. \( (f - \gamma g)(x) \geq 0 \) for all \( x \in \mathbb{R}^n \).

in other words, Hilbert (1888): iff exists \( \gamma \geq 0, \ S_0(x) \) s.t. \( f(x) = S_0(x) + \gamma g(x) \)

where \( S_0(x) \) is a sum of squares of polynomials! This paper started the field of algebraic geometry

And how about generalizations of the S-Lemma?

Given quadratics \( Q_0(x), Q_1(x), \ldots, Q_m(x) \) with \( m \geq 2 \), is it true that

\[
Q_0(x) \geq 0 \quad \text{whenever} \quad Q_i(x) \geq 0, \quad 1 \leq i \leq m,
\]

iff exist \( \gamma_i \geq 0, \ S_0(x), \) s.t. \( Q_0(x) = S_0(x) + \sum_{i=1}^{m} \gamma_i Q_i(x), \ S_0(x) = \text{a sum of squares?} \)

No.
Back to S-Lemma, +

Let \( f, g : \mathbb{R}^n \to \mathbb{R} \) be quadratic functions (degree \( \leq 2 \) polynomials).

Suppose there exists \( \bar{x} \in \mathbb{R}^n \) such that \( g(\bar{x}) > 0 \). Then

\[
f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0 \quad \text{iff} \quad \exists \gamma \geq 0 \quad \text{s.t.} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.
\]

i.e., iff exists \( \gamma \geq 0 \) s.t. \( (f - \gamma g)(x) \geq 0 \) for all \( x \in \mathbb{R}^n \).

In other words, Hilbert (1888): iff exists \( \gamma \geq 0 \), \( S_0(x) \) s.t. \( f(x) = S_0(x) + \gamma g(x) \)

where \( S_0(x) \) is a sum of squares of polynomials! This paper started the field of algebraic geometry

And how about generalizations of the S-Lemma?

Given quadratics \( Q_0(x), Q_1(x), \ldots, Q_m(x) \) with \( m \geq 2 \), is it true that

\[
Q_0(x) \geq 0 \quad \text{whenever} \quad Q_i(x) \geq 0, \quad 1 \leq i \leq m,
\]

\[
\text{iff exist} \quad \gamma_i \geq 0, \quad S_0(x), \quad \text{s.t.} \quad Q_0(x) = S_0(x) + \sum_{i=1}^{m} \gamma_i Q_i(x), \quad S_0(x) = \text{a sum of squares?}
\]

No.

However*: \( Q_0(x) > 0 \quad \text{whenever} \quad Q_i(x) \geq 0, \quad 1 \leq i \leq m, \)

\[
\text{iff exist} \quad S_0(x), S_1(x), \ldots S_m(x) \quad \text{s.t.} \quad Q_0(x) = S_0(x) + \sum_{i=1}^{m} S_i(x) Q_i(x)
\]

where each \( S_i(x) \) is a sum of squares of polynomials. Putinar (1993).
Back to S-Lemma, +

Let \( f, g : \mathbb{R}^n \to \mathbb{R} \) be quadratic functions (degree \( \leq 2 \) polynomials).

Suppose there exists \( \bar{x} \in \mathbb{R}^n \) such that \( g(\bar{x}) > 0 \). Then

\[
f(x) \geq 0 \quad \text{whenever} \quad g(x) \geq 0 \quad \text{iff exists} \quad \gamma \geq 0 \quad \text{s.t.} \quad f(x) \geq \gamma g(x) \quad \text{for all} \quad x \in \mathbb{R}^n.
\]
i.e., iff exists \( \gamma \geq 0 \) s.t. \( (f - \gamma g)(x) \geq 0 \) for all \( x \in \mathbb{R}^n \).

in other words, Hilbert (1888): iff exists \( \gamma \geq 0 \), \( S_0(x) \) s.t. \( f(x) = S_0(x) + \gamma g(x) \)

where \( S_0(x) \) is a sum of squares of polynomials! This paper started the field of algebraic geometry

And how about generalizations of the S-Lemma?

Given quadratics \( Q_0(x), Q_1(x), \ldots, Q_m(x) \) with \( m \geq 2 \), is it true that

\[
Q_0(x) \geq 0 \quad \text{whenever} \quad Q_i(x) \geq 0, \quad 1 \leq i \leq m,
\]

iff exist \( \gamma_i \geq 0 \), \( S_0(x), S_0(x), \ldots S_m(x) \) s.t. \( Q_0(x) = S_0(x) + \sum_{i=1}^{m} \gamma_i Q_i(x), \quad S_0(x) = \text{a sum of squares?} \)

No.

However*: \( Q_0(x) > 0 \quad \text{whenever} \quad Q_i(x) \geq 0, \quad 1 \leq i \leq m, \)

iff exist \( S_0(x), S_1(x), \ldots S_m(x) \) s.t. \( Q_0(x) = S_0(x) + \sum_{i=1}^{m} S_i(x)Q_i(x) \)

where each \( S_i(x) \) is a sum of squares of polynomials. Putinar (1993).

* \( \{x \in \mathbb{R}^n : Q_i(x) \geq 0, \ 1 \leq i \leq m\} \) is bounded (and represented as such)
More complete statement of Putinar’s theorem – still abridged

- Given **polynomials** $P_0(x), G_1(x), \ldots, G_m(x), \ x \in \mathbb{R}^n$,

- One of the $G_i(x)$ being $\|x\|^2 \leq R^2$
More complete statement of Putinar’s theorem – still abridged

- Given polynomials $P_0(x), G_1(x), \ldots, G_m(x), \ x \in \mathbb{R}^n,$

- One of the $G_i(x)$ being $\|x\|^2 \leq R^2$

- Then: $P_0(x) > 0$ in $\{x : G_i(x) \geq 0, \ 1 \leq i \leq m\}$ implies:

$$P_0(x) = S_0(x) + \sum_{i=1}^{m} S_i(x)G_i(x)$$

where each $S_i(x)$ is a sum of squares of polynomials
More complete statement of Putinar’s theorem – still abridged

- Given polynomials $P_0(x), G_1(x), \ldots, G_m(x)$, $x \in \mathbb{R}^n$,

- One of the $G_i(x)$ being $\|x\|^2 \leq R^2$

- Then: $P_0(x) > 0$ in $\{ x : G_i(x) \geq 0, \ 1 \leq i \leq m \}$ implies:

$$P_0(x) = S_0(x) + \sum_{i=1}^{m} S_i(x) G_i(x)$$

where each $S_i(x)$ is a sum of squares of polynomials

Questions
- What are the $S_i(x)$? Can we compute them efficiently?
More complete statement of Putinar’s theorem – still abridged

• Given polynomials $P_0(x), G_1(x), \ldots, G_m(x)$, $x \in \mathbb{R}^n$,

• One of the $G_i(x)$ being $\|x\|^2 \leq R^2$

• Then: $P_0(x) > 0$ in $\{ x : G_i(x) \geq 0, 1 \leq i \leq m \}$ implies:

$$P_0(x) = S_0(x) + \sum_{i=1}^{m} S_i(x)G_i(x)$$

where each $S_i(x)$ is a sum of squares of polynomials

Questions

• What are the $S_i(x)$? Can we compute them efficiently?

• Can we at least estimate them? Can we say anything about their degree?
More complete statement of Putinar’s theorem – still abridged

- Given polynomials $P_0(x), G_1(x), \ldots, G_m(x)$, $x \in \mathbb{R}^n$,

- One of the $G_i(x)$ being $\|x\|^2 \leq R^2$

- Then: $P_0(x) > 0$ in $\{ x : G_i(x) \geq 0, \ 1 \leq i \leq m \}$ implies:

$$P_0(x) = S_0(x) + \sum_{i=1}^m S_i(x)G_i(x)$$

where each $S_i(x)$ is a sum of squares of polynomials

Questions

- What are the $S_i(x)$? Can we compute them efficiently?

- Can we at least estimate them? Can we say anything about their degree?

Nie and Schweighofer (2005): upper bound on the max degree, as a function of the $P_0, G_1, \ldots, G_m$. 
More complete statement of Putinar’s theorem – still abridged

• Given polynomials $P_0(x), G_1(x), \ldots, G_m(x)$, $x \in \mathbb{R}^n$,

• One of the $G_i(x)$ being $\|x\|^2 \leq R^2$

• Then: $P_0(x) > 0$ in $\{ x : G_i(x) \geq 0, 1 \leq i \leq m \}$ implies:

$$P_0(x) = S_0(x) + \sum_{i=1}^m S_i(x) G_i(x)$$

where each $S_i(x)$ is a sum of squares of polynomials

Questions

• What are the $S_i(x)$? Can we compute them efficiently?

• Can we at least estimate them? Can we say anything about their degree?

Nie and Schweighofer (2005): upper bound on the max degree, as a function of the $P_0, G_1, \ldots, G_m$.

• How can Putinar’s result help us solve

$$\min \ P_0(x)$$

s.t. $G_i(x) \geq 0$, $1 \leq i \leq m$?
\[ P^* \triangleq \min P_0(x) \]
\[ \text{s.t. } G_i(x) \geq 0, \quad 1 \leq i \leq m? \]
\[ P^* = \min \quad P_0(x) \]
\[ \text{s.t.} \quad G_i(x) \geq 0, \quad 1 \leq i \leq m? \]

**Idea:** constrain the degrees of the sum-of-square “certificate” polynomials \( S_i(x) \)
$$P^* \overset{\text{def}}{=} \min \quad P_0(x)$$
\text{s.t.} \quad G_i(x) \geq 0, \quad 1 \leq i \leq m.

\text{Idea:} \text{ constrain the degrees of the sum-of-square “certificate” polynomials} \ S_i(x)

\text{Pick an integer} \ t > 0, \text{ and define}

$$P^{(t)} \overset{\text{def}}{=} \sup \ \rho$$
\text{s.t.} \quad P_0(x) - \rho = S_0(x) + \sum_{i=1}^{m} S_i(x)G_i(x)

\text{each} \ S_i(x) \text{ SOS}

\text{deg}(S_0(x)) \leq 2t, \quad \text{deg}(S_i(x)g_i(x)) \leq 2t.$$
\[ P^* \overset{\text{def}}{=} \min_{P_0(x)} \quad \text{s.t.} \quad G_i(x) \geq 0, \quad 1 \leq i \leq m? \]

**Idea:** constrain the degrees of the sum-of-square “certificate” polynomials \( S_i(x) \)

**Pick** an integer \( t > 0 \), and define

\[ P(t) \overset{\text{def}}{=} \sup \rho \quad \text{s.t.} \quad P_0(x) - \rho = S_0(x) + \sum_{i=1}^{m} S_i(x)G_i(x) \]

each \( S_i(x) \) **SOS**

\[ \deg(S_0(x)) \leq 2t, \quad \deg(S_i(x)g_i(x)) \leq 2t. \]

- \( P(t) \leq P^* \)
- \( P(t) \to P^* \) as \( t \to +\infty \) (finite convergence)

- Does this help?
\[ P^{(t)} = \sup \tau \]

s.t. \[ P_0(x) - \tau = S_0(x) + \sum_{i=1}^{m} S_i(x)G_i(x) \]

each \( S_i(x) \) \text{ SOS} \quad \begin{align*}
\deg(S_0(x)) &\leq 2t, \\
\deg(S_i(x)g_i(x)) &\leq 2t.
\end{align*}
\[ P^{(t)} \triangleq \sup \rho \]

s.t. \[ P_0(x) - \rho = S_0(x) + \sum_{i=1}^{m} S_i(x) G_i(x) \]

Here, \textcolor{blue}{blue} polynomials are known, \textcolor{black}{black} polynomials are unknown.
Example:

\[(\alpha x_1^2 + \beta x_1 x_2 + \gamma x_1) (x_1 + 2x_2 + 1)\]

\[= \alpha x_1^3 + (2\alpha + \beta)x_1^2 x_2 + 2\beta x_1 x_2^2 + (\alpha + \gamma)x_1^2 + (\beta + 2\gamma)x_1 x_2 + \gamma x_1\]
\[ P^{(t)} := \sup \rho \]

s.t. \[ P_0(x) - \rho = S_0(x) + \sum_{i=1}^m S_i(x) G_i(x) \] 

(2)

each \( S_i(x) \) is SOS

\[ \deg(S_0(x)) \leq 2t, \quad \deg(S_i(x)g_i(x)) \leq 2t. \]

**FACT:** \( P^{(t)} \) can be computed as a semidefinite program of dimension \( O(n^t) \)

**FACT:** Checking whether a given polynomial \( F(x) \) is SOS can be stated as an SDP
\[ P^{(t)} = \sup \rho \]

s.t. \[ P_0(x) - \rho = S_0(x) + \sum_{i=1}^{m} S_i(x)G_i(x) \]

each \( S_i(x) \) is SOS

\[ \deg(S_0(x)) \leq 2t, \quad \deg(S_i(x)g_i(x)) \leq 2t. \]

**FACT:** \( P^{(t)} \) can be computed as a semidefinite program of dimension \( O(n^t) \)

**FACT:** Checking whether a given polynomial \( F(x) \) is SOS can be stated as an SDP

**Example:**

\[ (x_1^2 + 2x_1 + x_2)^2 = (x_1^2 + 2x_1 + x_2)(x_1^2 + 2x_1 + x_2) \]

\[ (x_1^2 + 2x_1 + x_2) = (x_1^2 + 0x_2^2 + 0x_1x_2 + 2x_1 + x_2 + 0) = \]

\[
\begin{pmatrix}
1 \\
0 \\
0 \\
2 \\
1 \\
0
\end{pmatrix}
\]

\[ (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \]
\[
(x_1^2 + 2x_1 + x_2) = (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix}
\]
\[(x_1^2 + 2x_1 + x_2) = \begin{pmatrix} x_1^2, x_2^2, x_1x_2, x_1, x_2, 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix} \]

So \[(x_1^2 + 2x_1 + x_2)^2 = (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix} (1, 0, 0, 2, 1, 0) \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix} = \]
\[
(x_1^2 + 2x_1 + x_2) = (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix}
\]

So \((x_1^2 + 2x_1 + x_2)^2 = (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix} (1, 0, 0, 2, 1, 0) \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 4 & 2 & 0 \\ 1 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix}
\[(x_1^2 + 2x_1 + x_2) \, \mathbf{=} \, (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \left( \begin{array}{c} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{array} \right) \] 

So \((x_1^2 + 2x_1 + x_2)^2 = (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \left( \begin{array}{c} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{array} \right) \begin{pmatrix} 1, 0, 0, 2, 1, 0 \end{pmatrix} \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix} = \]

\[(x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \begin{pmatrix} 1 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 4 & 2 & 0 \\ 1 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix} = \]

So: if a given polynomial \( F(x_1, x_2) \) is a sum of squares of quadratic polynomials in \( x_1, x_2 \), then:
\[(x_1^2 + 2x_1 + x_2) = (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix}\]

So \((x_1^2 + 2x_1 + x_2)^2 = (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix} (1, 0, 0, 2, 1, 0) \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix} =\]

\[
\begin{pmatrix}
1 & 0 & 0 & 2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 4 & 2 & 0 \\
1 & 0 & 0 & 2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
x_1^2 \\ x_2^2 \\ x_1x_2 \\ x_1 \\ x_2 \\ 1
\end{pmatrix}
\]

So: if a given polynomial \(F(x_1, x_2)\) is a sum of squares of quadratic polynomials in \(x_1, x_2\), then:

\[F(x_1, x_2) = (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1),\] times a PSD matrix, times \[
\begin{pmatrix}
x_1^2 \\ x_2^2 \\ x_1x_2 \\ x_1 \\ x_2 \\ 1
\end{pmatrix}
\]
Consider the optimization problem

\[ f^* \triangleq \min f(x) : x \in K \]

where \( f(x) \) continuous, \( K \subseteq \mathbb{R}^n \) compact.
Something different

Consider the optimization problem

\[ f^* \doteq \min f(x) : x \in K \]

where \( f(x) \) continuous, \( K \subseteq \mathbb{R}^n \) compact

\[ f^* \leq f(x), \quad \forall x \in K \]
Something different

Consider the optimization problem

\[ f^* \doteq \min f(x) : x \in K \]

where \( f(x) \) continuous, \( K \subseteq \mathbb{R}^n \) compact

\[ f^* \leq f(x), \ \forall x \in K \]

so if \( \mu \) is a measure over \( K \), i.e. \( \int_K d\mu = 1 \),
Consider the optimization problem

\[ f^* \doteq \min_{x \in K} f(x) \]

where \( f(x) \) continuous, \( K \subseteq \mathbb{R}^n \) compact

\[ f^* \leq f(x), \quad \forall x \in K \]

so if \( \mu \) is a measure over \( K \), i.e. \( \int_K d\mu = 1 \), then

\[ f^* \leq \mathbb{E}_\mu f(x) \]
Consider the optimization problem

\[ f^* \triangleq \min_{x} f(x) : x \in K \]

where \( f(x) \) continuous, \( K \subseteq \mathbb{R}^n \) compact

\[ f^* \leq f(x), \quad \forall x \in K \]

so if \( \mu \) is a measure over \( K \), i.e. \( \int_K d\mu = 1 \), then

\[ f^* \leq \mathbb{E}_\mu f(x) \]

and so

\[ f^* \leq \inf_{\mu} \mathbb{E}_\mu f(x) \]
Consider the optimization problem

\[ f^* \triangleq \min_{x} f(x) : x \in K \]

where \( f(x) \) continuous, \( K \subseteq \mathbb{R}^n \) compact

\[ f^* \leq f(x), \quad \forall x \in K \]

so if \( \mu \) is a measure over \( K \), i.e. \( \int_{K} d\mu = 1 \), then

\[ f^* \leq \mathbb{E}_{\mu} f(x) \]

and so

\[ f^* \leq \inf_{\mu} \mathbb{E}_{\mu} f(x) \]

Suppose \( y \in K \) has \( f(y) = f^* \),
Consider the optimization problem

$$f^* \triangleq \min f(x) : x \in K$$

where $f(x)$ continuous, $K \subseteq \mathbb{R}^n$ compact.

$$f^* \leq f(x), \ \forall x \in K$$

so if $\mu$ is a measure over $K$, i.e. $\int_K d\mu = 1$, then $f^* \leq E_{\mu} f(x)$

and so

$$f^* \leq \inf_{\mu} E_{\mu} f(x)$$

Suppose $y \in K$ has $f(y) = f^*$, and let $\delta_y$ be the measure with weight 1 at $y$
Consider the optimization problem

$$f^* \triangleq \min f(x) : x \in K$$

where $f(x)$ continuous, $K \subseteq \mathbb{R}^n$ compact

$$f^* \leq f(x), \ \forall x \in K$$

so if $\mu$ is a measure over $K$, i.e. $\int_K d\mu = 1$, then $f^* \leq \mathbb{E}_\mu f(x)$

and so

$$f^* \leq \inf_\mu \mathbb{E}_\mu f(x)$$

Suppose $y \in K$ has $f(y) = f^*$, and let $\delta_y$ be the measure with weight $1$ at $y$

Then $f^* = f(y) = \mathbb{E}_{\delta_y} f(x)$
Consider the optimization problem

$$f^* = \min f(x) : x \in K$$

where $f(x)$ continuous, $K \subseteq \mathbb{R}^n$ compact

$$f^* \leq f(x), \quad \forall x \in K$$

so if $\mu$ is a measure over $K$, i.e. $\int_K d\mu = 1$, then $f^* \leq \mathbb{E}_\mu f(x)$

and so

$$f^* \leq \inf_{\mu} \mathbb{E}_\mu f(x)$$

Suppose $y \in K$ has $f(y) = f^*$, and let $\delta_y$ be the measure with weight 1 at $y$

Then $f^* = f(y) = \mathbb{E}_{\delta_y} f(x)$

And so

$$f^* = \inf_{\mu} \mathbb{E}_\mu f(x)$$
Consider the optimization problem

\[ f^* \doteq \min_{x \in K} f(x) \]

where \( f(x) \) continuous, \( K \subseteq \mathbb{R}^n \) compact

\[ f^* \leq f(x), \quad \forall x \in K \]

so if \( \mu \) is a measure over \( K \), i.e. \( \int_K d\mu = 1 \), then

\[ f^* \leq \mathbb{E}_\mu f(x) \]

and so

\[ f^* \leq \inf_\mu \mathbb{E}_\mu f(x) \]

Suppose \( y \in K \) has \( f(y) = f^* \), and let \( \delta_y \) be the measure with weight \( 1 \) at \( y \)

Then \( f^* = f(y) = \mathbb{E}_{\delta_y} f(x) \)

And so

\[ f^* = \inf_\mu \mathbb{E}_\mu f(x) \]

How do we use this fact?
Polynomial optimization

Consider the polynomial optimization problem

$$f^*_0 \doteq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \},$$

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

- Each $\pi$ 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 \textit{tuples}, and the $a_{i,\pi}$ are reals.
Polynomial optimization

Consider the polynomial optimization problem

$$f^*_0 \doteq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \},$$

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

- Each $\pi$ 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.

We know $f^*_0 = \inf_{\mu} \mathbb{E}_\mu f_0(x)$, over all measures $\mu$ over $K \doteq \{ x \in \mathbb{R}^n : f_i(x) \geq 0, \ 1 \leq i \leq m \}$. 
Polynomial optimization

Consider the polynomial optimization problem

\[ f^*_0 \doteq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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

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

We know \( f^*_0 = \inf_{\mu} \mathbb{E}_\mu f_0(x) \), over all measures \( \mu \) over \( K = \{ x \in \mathbb{R}^n : f_i(x) \geq 0, \ 1 \leq i \leq m \} \).

i.e. \( f^*_0 = \inf \left\{ \sum_{\pi \in S(0)} a_{0,\pi} y_\pi : y \text{ is a } K\text{-moment} \right\} \)

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

Consider the polynomial optimization problem

\[ f^*_0 \doteq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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

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

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

We know \( f^*_0 = \inf_\mu \mathbb{E}_\mu f_0(x) \), over all measures \( \mu \) 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 \( \mu \) over \( K \) with \( y_\pi = \mathbb{E}_\mu x^\pi \) for each tuple \( \pi \)

(Cough! Here, \( y \) is an infinite-dimensional vector).
Polynomial optimization

Consider the polynomial optimization problem

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

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

- Each \( \pi \) is a tuple \( \pi_1, \pi_2, \ldots, \pi_n \) of \textbf{nonnegative integers}, and \( x^\pi \doteq x_1^{\pi_1} x_2^{\pi_2} \cdots x_n^{\pi_n} \)

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

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

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

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

\textbf{(Cough! Here, } y \text{ is an infinite-dimensional vector). Can we make an easier statement?}
Polynomial optimization

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

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

\[ f^*_0 = \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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 \( \mu \) over \( K \) with \( y_\pi = \mathbb{E}_\mu x^\pi \) for each tuple \( \pi \))

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

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

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 \( \mu \) over \( K \) with \( y^\pi = \mathbb{E}_\mu x^\pi \) for each tuple \( \pi \))

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

So: \( y_0 = 1. \)
Polynomial optimization

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

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 \( \mu \) over \( K \) with \( y_\pi = \mathbb{E}_\mu x^{\pi} \) for each tuple \( \pi \))

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

So: \( y_0 = 1 \). Can we say more?
Polynomial optimization

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

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 \( \mu \) over \( K \) with \( y_\pi = \mathbb{E}_\mu x^\pi \) for each tuple \( \pi \))

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

So: \( y_0 = 1 \). Can we say more? Define \( v = (x^\pi) \) (all monomials).
Polynomial optimization

\[ f_0^* = \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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 \( \mu \) over \( K \) with \( y_\pi = \mathbb{E}_\mu x^\pi \) for each tuple \( \pi \) )

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

So: \( y_0 = 1. \) Can we say more? Define \( v = (x^\pi) \) (all monomials). Also define \( M[y] \triangleq \mathbb{E}_\mu vv^T. \)
Polynomial optimization

\[ f_0^* \triangleq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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 \( \mu \) over \( K \) with \( y_{\pi} = \mathbb{E}_\mu x^\pi \) for each tuple \( \pi \))

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

So: \( y_0 = 1. \) Can we say more? Define \( v = (x^\pi) \) (all monomials). Also define \( M[y] \triangleq \mathbb{E}_\mu vv^T. \)

So for any tuples \( \pi, \rho, \) \( M[y]_{\pi,\rho} = \mathbb{E}_\nu x^\pi x^\rho = \mathbb{E}_\nu x^{\pi+\rho} = y_{\pi+\rho} \).
Polynomial optimization

\[ f_0^* \triangleq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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 \( \mu \) over \( K \) with \( y_{\pi} = \mathbb{E}_\mu x^\pi \) for each tuple \( \pi \))

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

So: \( y_0 = 1. \) Can we say more? Define \( v = (x^\pi) \) (all monomials). Also define \( M[y] \triangleq \mathbb{E}_\mu vv^T. \)

So for any tuples \( \pi, \rho, \) \( M[y]_{\pi,\rho} = \mathbb{E}_\nu x^\pi x^\rho = \mathbb{E}_\nu x^{\pi+\rho} = y_{\pi+\rho} \)

So for any (\( \infty \)-dimensional) vector \( z, \) indexed by tuples, i.e. with entries \( z_\pi \) for each tuple \( \pi, \)
Polynomial optimization

\[ f^*_0 \overset{\text{def}}{=} \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \} , \]

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 \( \mu \) over \( K \) with \( y_{\pi} = \mathbb{E}_\mu x^\pi \) for each tuple \( \pi \) )

\((K \overset{\text{def}}{=} \{ x \in \mathbb{R}^n : f_i(x) \geq 0, \ 1 \leq i \leq m \})\).

So: \( y_0 = 1 \). Can we say more? Define \( v = (x^\pi) \) (all monomials). Also define \( M[y] \overset{\text{def}}{=} E_\mu vv^T \).

So for any tuples \( \pi, \rho \), \( M[y]_{\pi,\rho} = \mathbb{E}_\nu x^{\pi}x^{\rho} = \mathbb{E}_\nu x^{\pi+\rho} = y_{\pi+\rho} \)

So for any (\( \infty \)-dimensional) vector \( z \), indexed by tuples, i.e. with entries \( z_{\pi} \) for each tuple \( \pi \),

\[ z^T M[y] z = \sum_{\pi,\rho} E_\mu z_{\pi} x^{\pi} x^{\rho} z_{\rho} = E_\mu (\sum_{\pi} z_{\pi} x^{\pi})^2 \geq 0 \]
Polynomial optimization

\[ f_0^* \triangleq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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 \( \mu \) over \( K \) with \( y_\pi = \mathbb{E}_{\mu} x^\pi \) for each tuple \( \pi \))

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

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

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

So for any (\( \infty\)-dimensional) vector \( z, \) indexed by tuples, i.e. with entries \( z_\pi \) for each tuple \( \pi, \)

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

so \( M[y] \succeq 0 !! \)
Polynomial optimization

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

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 \( y; \)

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

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

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

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

So for any (\( \infty \)-dimensional) vector \( z, \) \text{ indexed by tuples}, i.e. with entries \( z_\pi \) for each tuple \( \pi, \)

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

so \( M[y] \succeq 0 \) !!

so

\[ f_0^* \geq \min \sum_{\pi} a_{0,\pi} y_\pi \]

s.t. \( y_0 = 1, \)

\( M \succeq 0, \)

\( M_{\pi,\rho} = y_{\pi+\rho}, \) for all tuples \( \pi, \rho \)

the zeroth row and column of \( M \) both equal \( y. \) (redundant)
Polynomial optimization

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

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 \( \mu \) over \( K \) with \( y_\pi = \mathbb{E}_\mu x^\pi \) for each tuple \( \pi \))

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

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

So for any tuples \( \pi, \rho \), \( M[y]_{\pi,\rho} = E_\nu x^\pi x^\rho = E_\nu x^{\pi+\rho} = y_{\pi+\rho} \)

So for any (\( \infty \)-dimensional) vector \( z \), indexed by tuples, i.e. with entries \( z_\pi \) for each tuple \( \pi \),

\[ z^T M[y] z = \sum_{\pi,\rho} E_\mu z_\pi x^\pi x^\rho z_\rho = E_\mu (\sum_\pi z_\pi x^\pi)^2 \geq 0 \]

so \( M[y] \succeq 0 \) !!

so

\[ f_0^* \geq \min \sum_{\pi} a_{0,\pi} y_\pi \]

s.t. \( y_0 = 1, \)
\( M \succeq 0, \)
\( M_{\pi,\rho} = y_{\pi+\rho}, \) for all tuples \( \pi, \rho \)

the zeroth row and column of \( M \) both equal \( y \).

An infinite-dimensional semidefinite program!!
\[ f_0^* = \min \{ f_0(x) : f_i(x) \geq 0, \quad 1 \leq i \leq m, \quad x \in \mathbb{R}^n \}, \]

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

\[ f_0^* \geq \min \sum_{\pi} a_{0,\pi} y_{\pi} \]

s.t. \( y_0 = 1 \),
\( M \succeq 0 \),
\( M_{\pi,\rho} = y_{\pi+\rho} \), for all tuples \( \pi, \rho \)
the zeroth row and column of \( M \) both equal \( y \).
\[ f_0^* = \min \{ f_0(x) : f_i(x) \geq 0, \quad 1 \leq i \leq m, \quad x \in \mathbb{R}^n \}, \]

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

\[ f_0^* \geq \min \sum_{\pi} a_{0,\pi} y_{\pi} \]

s.t. \quad \begin{align*}
y_0 &= 1, \\
M &\succeq 0, \\
M_{\pi,\rho} &= y_{\pi+\rho}, \quad \text{for all tuples } \pi, \rho \end{align*}

the zeroth row and column of \( M \) both equal \( y \).

**Restrict:** pick an integer \( d \geq 1 \). Restrict the SDP to all tuples \( \pi \) with \( |\pi| \leq d \).
\[
f_0^* = \min \{ f_0(x) : f_i(x) \geq 0, \quad 1 \leq i \leq m, \quad x \in \mathbb{R}^n \},
\]

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

\[
f_0^* \geq \min \sum_{\pi} a_{0,\pi} y_{\pi}
\]
s.t. \( y_0 = 1 \),

\( M \succeq 0 \),

\( M_{\pi,\rho} = y_{\pi+\rho} \), for all tuples \( \pi, \rho \)

the zeroth row and column of \( M \) both equal \( y \).

**Restrict:** pick an integer \( d \geq 1 \). Restrict the SDP to all tuples \( \pi \) 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 \( 1 + 7 + 1 > 8 \).
\[ f^*_0 \doteq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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

\[ f^*_0 \geq \min \sum_{\pi} a_{0,\pi} y_{\pi} \]

s.t. \( y_0 = 1, \)
\( M \succeq 0, \)
\( M_{\pi,\rho} = y_{\pi+\rho}, \) for all tuples \( \pi, \rho \)
the zeroth row and column of \( M \) both equal \( y. \)

**Restrict:** pick an integer \( d \geq 1. \) Restrict the SDP to all tuples \( \pi \) with \(|\pi| \leq d. \)

\[ f^*_0 \geq \min \sum_{\pi} a_{0,\pi} y_{\pi} \]

s.t. \( y_0 = 1, \)
the rows and columns of \( M, \) and the entries in \( y, \) indexed by tuples of size \( \leq d \)
\( M \succeq 0, \)
\( M_{\pi,\rho} = y_{\pi+\rho}, \) for all appropriate tuples \( \pi, \rho \)
the zeroth row and column of \( M \) both equal \( y, \)
\[ f^*_0 = \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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

\[
\begin{align*}
f^*_0 &\geq \min \sum_{\pi} a_{0,\pi} y_{\pi} \\
s.t. & \quad y_0 = 1, \\
& \quad M \succeq 0, \\
& \quad M_{\pi,\rho} = y_{\pi+\rho}, \text{ for all tuples } \pi, \rho \\
& \quad \text{the zeroth row and column of } M \text{ both equal } y.
\end{align*}
\]

**Restrict:** pick an integer \( d \geq 1 \). Restrict the SDP to all tuples \( \pi \) with \( |\pi| \leq d \).

\[
\begin{align*}
f^*_0 &\geq \min \sum_{\pi} a_{0,\pi} y_{\pi} \\
s.t. & \quad y_0 = 1, \\
& \quad \text{the rows and columns of } M, \text{ and the entries in } y, \text{ indexed by tuples of size } \leq d \\
& \quad M \succeq 0, \\
& \quad M_{\pi,\rho} = y_{\pi+\rho}, \text{ for all appropriate tuples } \pi, \rho \\
& \quad \text{the zeroth row and column of } M \text{ both equal } y.
\end{align*}
\]

A **finite-dimensional** semidefinite program!!
\[ f^*_0 = \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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

\[ f^*_0 \geq \min \sum_{\pi} a_{0,\pi} y_\pi \]

s.t. \( y_0 = 1 \),

\[ M \succeq 0, \]

\[ M_{\pi,\rho} = y_{\pi+\rho}, \ \text{for all tuples } \pi, \rho \]

the zeroth row and column of \( M \) both equal \( y \).

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

\[ f^*_0 \geq \min \sum_{\pi} a_{0,\pi} y_\pi \]

s.t. \( y_0 = 1 \),

the rows and columns of \( M \), and the entries in \( y \), indexed by tuples of size \( \leq d \)

\[ M \succeq 0, \]

\[ M_{\pi,\rho} = y_{\pi+\rho}, \ \text{for all appropriate tuples } \pi, \rho \]

the zeroth row and column of \( M \) both equal \( y \)

A finite-dimensional semidefinite program!! But could be very large!!
\[ f_0^* \triangleq \min \left\{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \right\}, \]

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

\[
\begin{align*}
    f_0^* &\geq \min \sum_{\pi} a_{0,\pi} y_{\pi} \\
    \text{s.t.} \quad &y_0 = 1, \\
    &M \succeq 0, \\
    &M_{\pi,\rho} = y_{\pi+\rho}, \text{ for all tuples } \pi, \rho \\
    &\text{the zeroth row and column of } M \text{ both equal } y.
\end{align*}
\]

**Restrict:** pick an integer \( d \geq 1 \). Restrict the SDP to all tuples \( \pi \) with \( |\pi| \leq d \).

\[
\begin{align*}
    f_0^* &\geq \min \sum_{\pi} a_{0,\pi} y_{\pi} \\
    \text{s.t.} \quad &y_0 = 1, \\
    &\text{the rows and columns of } M, \text{ and the entries in } y, \text{ indexed by tuples of size } \leq d \\
    &M \succeq 0, \\
    &M_{\pi,\rho} = y_{\pi+\rho}, \text{ for all appropriate tuples } \pi, \rho \\
    &\text{the zeroth row and column of } M \text{ both equal } y
\end{align*}
\]

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

• Can be strengthened to account for the constraints \( f_i(x) \geq 0 \).
\[ f_0^* \triangleq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \}, \]

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

\[ f_0^* \geq \min \sum_{\pi} a_{0,\pi} y_\pi \]

s.t. \( y_0 = 1 \),
\( M \succeq 0 \),
\( M_{\pi,\rho} = y_{\pi+\rho} \),
the zeroth row and column of \( M \) both equal \( y \).

**Restrict:** pick an integer \( d \geq 1 \). Restrict the SDP to all tuples \( \pi \) with \( |\pi| \leq d \).

\[ f_0^* \geq \min \sum_{\pi} a_{0,\pi} y_\pi \]

s.t. \( y_0 = 1 \),
the rows and columns of \( M \), and the entries in \( y \), indexed by tuples of size \( \leq d \)
\( M \succeq 0 \),
\( M_{\pi,\rho} = y_{\pi+\rho} \), for all appropriate tuples \( \pi, \rho \)
the zeroth row and column of \( M \) both equal \( y \)

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

- Can be strengthened to account for the constraints \( f_i(x) \geq 0 \).
- This is the level- \( d \) Lasserre relaxation (abridged).
\[ f^*_0 \triangleq \min \{ f_0(x) : f_i(x) \geq 0, \ 1 \leq i \leq m, \ x \in \mathbb{R}^n \} , \]

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

\[ f^*_0 \geq \min \sum_{\pi} a_{0,\pi} y_\pi \]

s.t. \( y_0 = 1, \)
\( M \succeq 0, \)
\( M_{\pi,\rho} = y_{\pi+\rho}, \)

the zeroth row and column of \( M \) both equal \( y. \)

**Restrict:** pick an integer \( d \geq 1. \) Restrict the SDP to all tuples \( \pi \) with \( |\pi| \leq d. \)

\[ f^*_0 \geq \min \sum_{\pi} a_{0,\pi} y_\pi \]

s.t. \( y_0 = 1, \)

the rows and columns of \( M \), and the entries in \( y \), indexed by tuples of size \( \leq d \)
\( M \succeq 0, \)
\( M_{\pi,\rho} = y_{\pi+\rho}, \) for all appropriate tuples \( \pi, \rho \)

the zeroth row and column of \( M \) both equal \( y. \)

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

- Can be strengthened to account for the constraints \( f_i(x) \geq 0. \)
- This is the level- \( d \) Lasserre relaxation (abridged).
- Dominates the SOS relaxations. Up to a point.