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 $\boldsymbol{km}$, 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:

- 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 problem

Find a solution to:

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

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

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

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

Many possible variations

- Line limits
- Various optimization versions

Quadratically constrained, quadratic programming problems (QCQPs):

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

Here,

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

is a general quadratic

Each M_i is $n \times n$, wlog symmetric

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

$$\boldsymbol{x^T M x} \quad = \quad \boldsymbol{x^T M^T x}, \text{ so } \boldsymbol{x^T M x} \quad = \quad \tfrac{1}{2}(\boldsymbol{x^T M x} + \boldsymbol{x^T M^T x}) \quad = \quad \boldsymbol{x^T} ($$

Quadratically constrained, quadratic programming problems (QCQPs):

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

Here,

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

is a general quadratic

Each M_i is $n \times n$, wlog symmetric

Special case: Linear Programming

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b, \\ & x \in \mathbb{R}^n \end{aligned}$$

Quadratically constrained, quadratic programming problems (QCQPs):

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

Here,

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

is a general quadratic

Each M_i is $n \times n$, wlog symmetric

Special case: Convex Quadratic Programming:

$$M_0 \succeq 0, \quad M_i \preceq 0, \quad 1 \le i \le m$$

Folklore result: QCQP is NP-hard

Folklore result: QCQP is NP-hard

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

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

Folklore result: QCQP is NP-hard

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

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

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

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

Folklore result: QCQP is NP-hard

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

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

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

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

- 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:

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

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

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

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:

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

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

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

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:

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

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

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

Actually, exactly what are "NP-hard" problems?

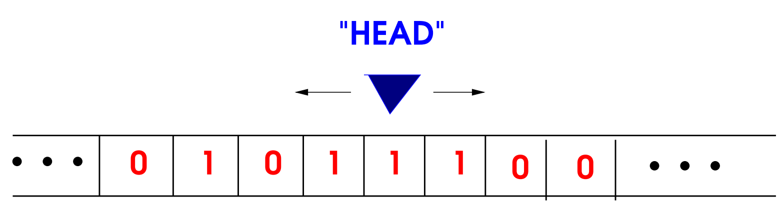
- Really, really hard problems?

- 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

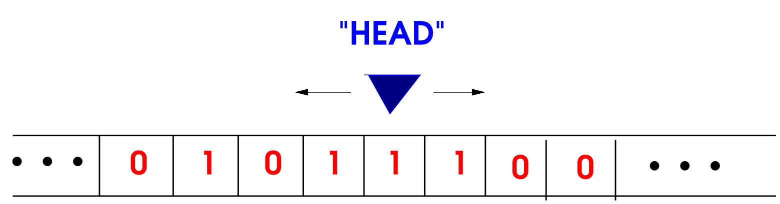


- In one step, head reads/writes and/or moves one unit

Digression: NP-hardness

"Turing Machine" model (bit model) of computing

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



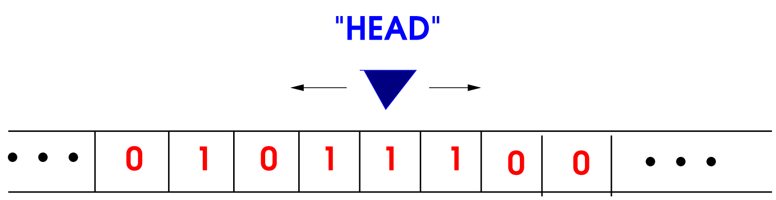
- In one step, head reads/writes and/or moves one unit

- 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

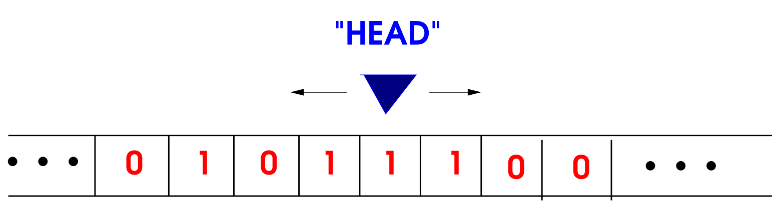
- Yes/no problems (like integer partition)

- A problem is in the class **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 **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

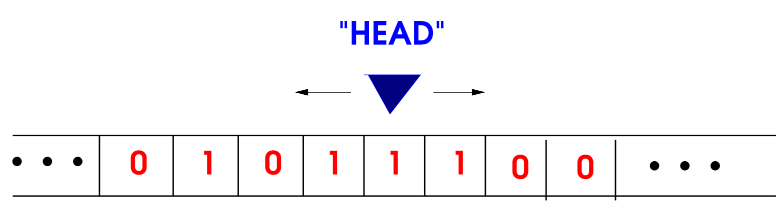
$$\sum_{j \in J} w_j = \sum_{j \notin J} w_j$$

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

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

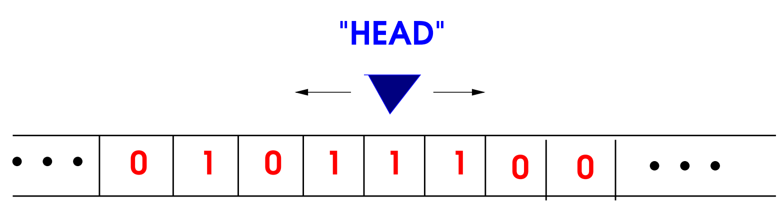
Note: We are simply **verifying** a certificate that somebody gave us

Proxy concept: problems in **NP** are “well-defined”

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

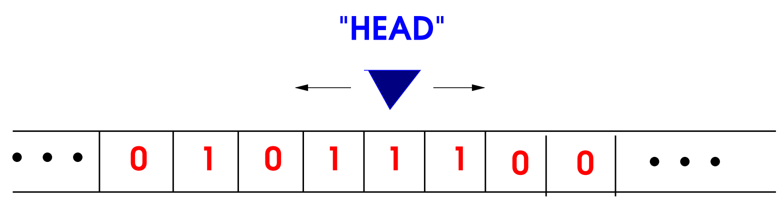
Proxy concept: problems in **NP** are "well-defined"

- A problem class $\mathcal{P}$ is **NP-complete** if any problem in the class **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 **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"

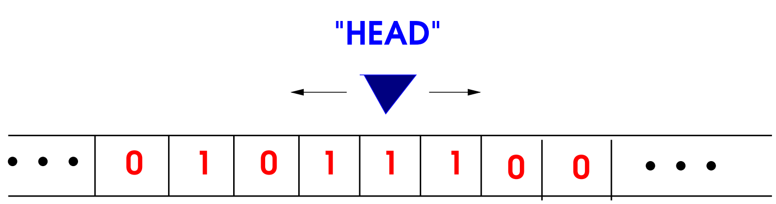
- A problem class $\mathcal{P}$ is **NP-complete** if any problem in the class **NP** can be reduced to a problem $\mathcal{P}$ in polynomial time

- As a consequence, if somebody smart figured a way to solve $\mathcal{P}$ in polynomial time, we can then solve **every** problem in **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 **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"

- A problem class $\mathcal{P}$ is **NP-complete** if any problem in the class **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 $\mathbf{0} < \boldsymbol{\epsilon} < \mathbf{1}$. Then we can compute a set J

- Such that

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

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

(So approximate feasibility, in "practicable" time)

Problem is **weakly** NP-hard

Folklore result: QCQP is NP-hard

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

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

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

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

- 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

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

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

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

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

$$\min \ c^T x$$

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

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

$\rightarrow$

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

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

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

(and many other similar transformations)

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

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

$\longrightarrow$

$$\begin{aligned} &\min \; c^T x \; - \; \boldsymbol{M} \sum_j x_j^2 \\ &\text{s.t. } \; Ax = b \\ &\quad -1 \le x_j \le 1, \;\; 1 \le j \le n. \end{aligned}$$

(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

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

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

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

$\rightarrow$

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

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

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

(and many other similar transformations)

$\longrightarrow$ **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

A. Verma (2009): AC-OPF is **strongly** NP-hard.

Even more general than QCQP:

Solving systems of polynomial equations.

Problem: given polynomials $p_i : \mathbb{R}^n \rightarrow \mathbb{R}$, 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 $p_i : \mathbb{R}^n \longrightarrow \mathbb{R}$, 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 a polynomial-time reduction

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)

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 **approximately?**

And what do we mean by **on the average?**

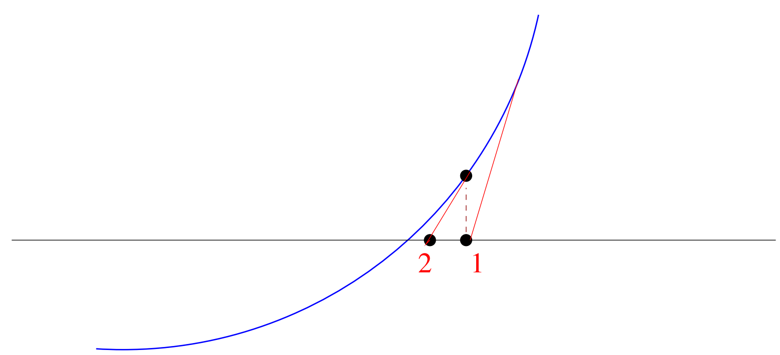
“Approximately”

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!

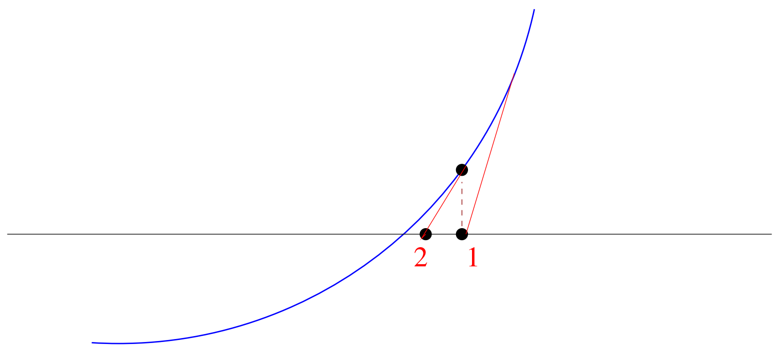


$\longrightarrow$ If we start near a solution, quadratic convergence

"Approximately"

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

A: Newton-Raphson, of course!



$\longrightarrow$ If we start near a solution, quadratic convergence

To solve $\boldsymbol{F(x) = 0}$, where $\boldsymbol{F} : \mathbb{R}^{\boldsymbol{n}} \longrightarrow \mathbb{R}^{\boldsymbol{n}}$:

Iterate: $\boldsymbol{x^{k+1} = -\left[J(x^k)\right]^{-1} F(x^k) + x^k}$, $\quad$ k = 1, ...

$$\boldsymbol{J(x^k)_{ij} = \frac{\partial J_i}{\partial x_j}(x^k) \qquad 1 \le i, j \le n}$$

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^{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?**

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

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?**

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

What is meant by **on the average?**

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?**

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 $\mathbb{R}^2$:

$$\begin{array}{rcrcl} x_1^2 & + & x_2^2 & \geq & 1 \\ x_1^2 & + & x_2^2 & \leq & 1 \\ x_1 & - & x_2 & = & 0 \\ x_1^2 & + & (x_2-1)^2 & \leq & 1 \end{array}$$

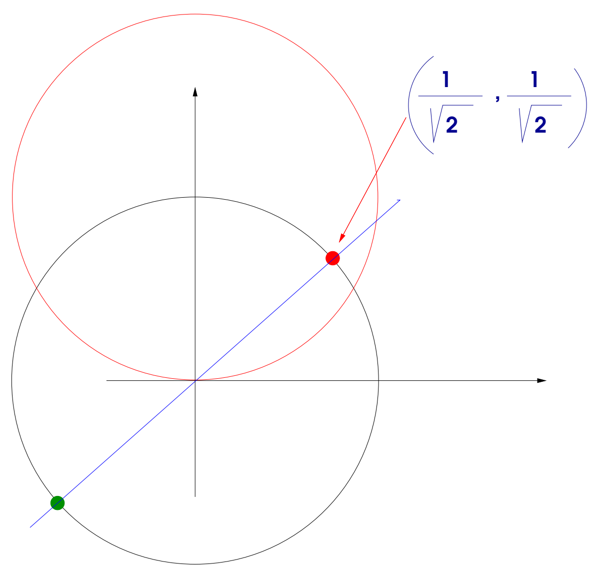
"On the average" in polynomial time

A QCQP could be **quite** difficult!

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

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

$$\begin{array}{rcrcl} x_1^2 & + & x_2^2 & \geq & 1 \\ x_1^2 & + & x_2^2 & \leq & 1 \\ x_1 & - & x_2 & = & 0 \\ x_1^2 & + & (x_2-1)^2 & \leq & 1 \end{array}$$



“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

$\longrightarrow$ 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

$\longrightarrow$ A **Las Vegas** algorithm: it may fail to converge, but with probability zero

Smale's 17^{th} 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 17^{th} 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 17^{th} 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 17^{th} 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 $f, g : \mathbb{R}^n \longrightarrow \mathbb{R}$ be quadratic functions (degree ≤ 2 polynomials).

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

$f(x) \geq 0$ whenever $g(x) \geq 0$

if and only

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

Yakubovich (1971), also much earlier, related work

γ acts as a Lagrange multiplier.

Quick aside:

Suppose we want to solve: $\boldsymbol{F^* \doteq \min\{ f(x) \ : \ g(x) \geq 0 \}}$; here $\boldsymbol{f, g}$ 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 θ.

2. Check if $f(x) - \theta \geq 0, \quad \forall\, x$ s.t. $g(x) \geq 0$.

3. If "yes", we know $F^* \geq \theta$; if not, $F^* < \theta$.

4. Either way we can update θ, and repeat. Works under compactness of $\{\, x \,:\, g(x) \geq 0 \,\}$.

Simplest example: S-Lemma (abridged)

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

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

$\boldsymbol{f}(\boldsymbol{x}) \geq \mathbf{0}$ whenever $\boldsymbol{g}(\boldsymbol{x}) \geq \mathbf{0}$

if and only

there exists $\boldsymbol{\gamma} \geq \mathbf{0}$ such that $\boldsymbol{f}(\boldsymbol{x}) \geq \boldsymbol{\gamma g}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$.

Yakubovich (1971), also much earlier, related work

$\boldsymbol{\gamma}$ acts as a Lagrange multiplier.

Corollary: Can solve

$$\min\{\boldsymbol{f}(\boldsymbol{x}) \,:\, \boldsymbol{g}(\boldsymbol{x}) \geq \mathbf{0}\}$$

in polynomial time (using semidefinite programming)

Simplest example: S-Lemma (abridged)

Let $f, g : \mathbb{R}^n \longrightarrow \mathbb{R}$ be quadratic functions (degree ≤ 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

γ acts as a Lagrange multiplier.

Corollary: Can solve

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

in polynomial time (using semidefinite programming)

$\longrightarrow$ Time for some math

Want to solve: $\mathbf{min}\{\boldsymbol{f(x)} \ : \ \boldsymbol{g(x)} \geq \mathbf{0}\}$

Given a real $\boldsymbol{\theta}$, is it the the case that $\boldsymbol{f(x) - \theta \geq 0}$ whenever $\boldsymbol{g(x) \geq 0}$?

Given a real $\boldsymbol{\theta}$, is it the the case that $\boldsymbol{f(x)} - \boldsymbol{\theta} \geq \mathbf{0}$ whenever $\boldsymbol{g(x)} \geq \mathbf{0}$?

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

Given a real $\boldsymbol{\theta}$, is it the the case that $\boldsymbol{f(x)} - \boldsymbol{\theta} \geq \mathbf{0}$ whenever $\boldsymbol{g(x)} \geq \mathbf{0}$?

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

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

Given a real $\boldsymbol{\theta}$, is it the the case that $\boldsymbol{f(x) - \theta \geq 0}$ whenever $\boldsymbol{g(x) \geq 0}$?

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

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

So S-Lemma statement is:

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

Given a real $\boldsymbol{\theta}$, is it the the case that $\boldsymbol{f(x) - \theta \geq 0}$ whenever $\boldsymbol{g(x) \geq 0}$?

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

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

So S-Lemma statement is:

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

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 $\boldsymbol{\theta}$, is it the the case that $\boldsymbol{f(x) - \theta \geq 0}$ whenever $\boldsymbol{g(x) \geq 0}$?

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

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

So S-Lemma statement is:

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

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

$$\max \quad \theta$$

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 $\boldsymbol{\gamma}$, $\boldsymbol{\theta}$.

Many applications for the S-Lemma

- Control Theory
- Dynamical Systems
- Robust error estimation
- Robust optimization
- ...

An application: the trust-region subproblem

$$\min\{f(x)\ :\ g(x) \le 0\}$$

can be solved in polynomial time, where $\boldsymbol{f}$, $\boldsymbol{g}$ quadratics, $\boldsymbol{g}$ convex

Scale, rotate, translate:

$$\min\{f(x)\ :\ \|x\| \le 1\}$$

Digression: application of trust-region subproblem in engineering

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

- $f(x)$ can be antyhing
- constraints mapped into $f(x)$ by using penalties

Digression: application of trust-region subproblem in engineering

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

- $f(x)$ can be antyhing
- 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

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

- $f(x)$ can be antyhing
- constraints mapped into $f(x)$ by using penalties

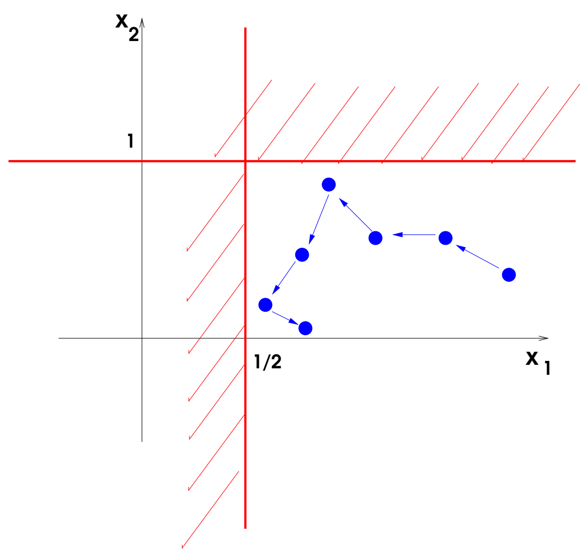
Example: $\min\{ g(x_1, x_2) : 1/2 \le x_1$ and $x_2 \le 1\}$

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

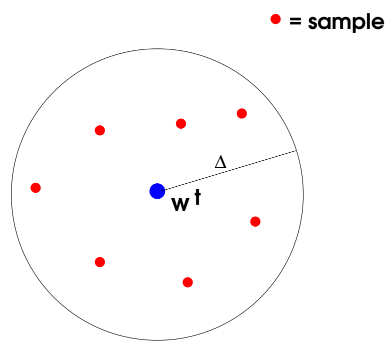
$\longrightarrow$ Unconstrained optimization $\min\{f(x) : x \in \mathbb{R}^n\}$

Digression: application of trust-region subproblem in engineering

$\longrightarrow$ 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 $(y^1, f(y^1)), (y^2, f(y^2)), \ldots, (y^m, f(y^m))$

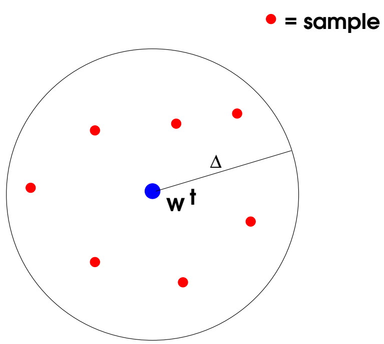
- Using these samples, construct a **quadratic** "model" of $f(x)$
(model = spline, least squares estimate, etc).

Digression: application of trust-region subproblem in engineering

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

Algorithm

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



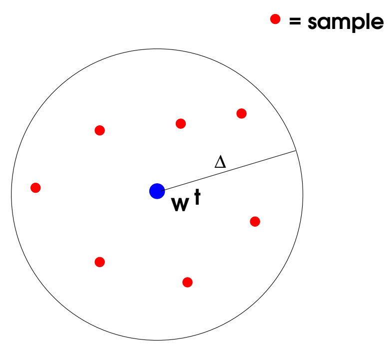
- Get pairs $(y^1, f(y^1)), (y^2, f(y^2)), \ldots, (y^m, f(y^m))$
- Using these samples, construct a **quadratic** "model" of $f(x)$ (model = spline, least squares estimate, etc).
- Call this model: $Q(x)$
- **Solve:** $\min\{\, Q(x) \,:\, \|x - w^t\| \leq \Delta\}$. This is the trust-region subproblem.
- 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

$\longrightarrow$ 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

$$\min\{f(x)\ :\ g(x) \le 0\}$$

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

Scale, rotate, translate:

$$\min\{f(x)\ :\ \|x\| \le 1\}$$

can be solved in poly time $\longrightarrow$ $\log \epsilon^{-1}$

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

How about *extensions* of the trust-region subproblem?

Sturm-Zhang (2003)

Where $f(x)$ is a quadratic,

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

can be solved in polynomial time, as can

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

Ye-Zhang (2003)

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

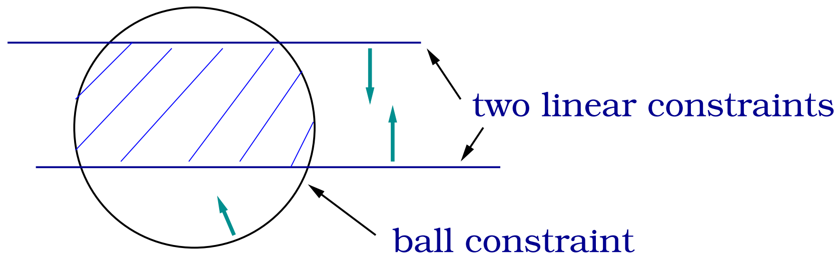
(two linear side constraints, but at least one binding)

Anstreicher-Burer (2012)

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

$$\begin{array}{lrcl} \min & x^T Q x + c^T x & & \\ & \text{s.t.} \quad \|x\| & \leq & 1 \\ & a_i^T x & \leq & b_i \quad i = 1, 2 \end{array}$$

provided the two linear constraints are parallel:

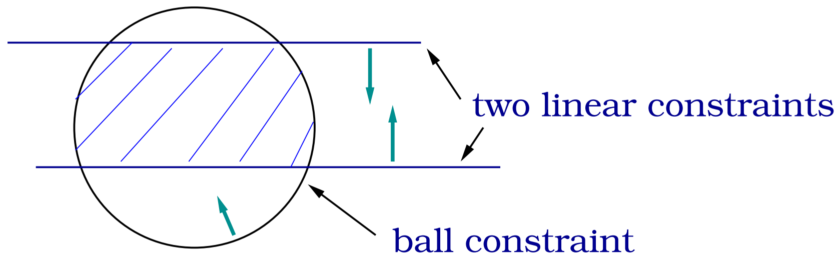


Anstreicher-Burer (2012)

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

$$\begin{array}{rrcl} \min & x^T Q x + c^T x & & \\ & \text{s.t.} \quad \|x\| & \leq & 1 \\ & a_i^T x & \leq & b_i \quad i = 1, 2 \end{array}$$

provided the two linear constraints are parallel:



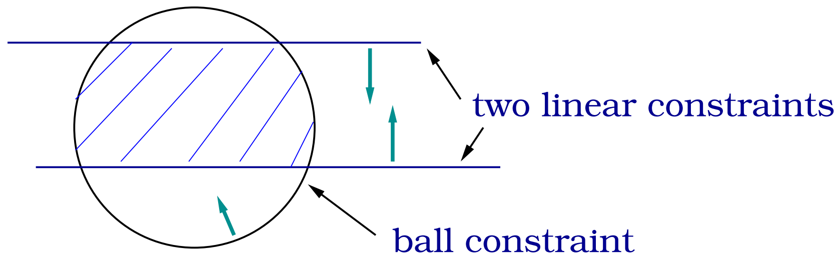
$$\longrightarrow \min \{ \, x^T Q x + c^T x \, : \, l \leq x_1 \leq u, \, \|x\| \leq 1 \, \}$$

Anstreicher-Burer (2012)

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

$$\begin{aligned} \min \quad & x^TQx + c^Tx \\ \text{s.t.} \quad & \|x\| \leq 1 \\ & a_i^T x \leq b_i \quad i = 1,2 \end{aligned}$$

provided the two linear constraints are parallel:



$$\longrightarrow \min \{ x^TQx + c^Tx \, : \, l \leq x_1 \leq u, \, \|x\| \leq 1 \} \qquad (*)$$

restate as:
$$\begin{aligned} \min \quad & \sum_{i,j} q_{ij} X_{ij} \; + \; c^T x \\ \text{s.t.} \quad & X_{11} + lu \leq (l+u)x_1 \\ & \|X_{.1} - lx\| \leq x_1 - l \\ & \|ux - X_{.1}\| \leq u - x_1 \\ & \sum_j X_{jj} \leq 1 \\ & X \succeq xx^T \end{aligned}$$

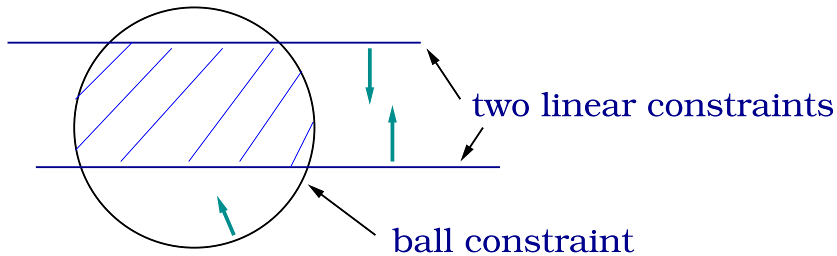
Equivalent to problem (*) ? Yes, if $\boldsymbol{X = xx^T}$, i.e. a **rank-1 solution**

Anstreicher-Burer (2012)

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

$$\begin{array}{rrcl} \min & x^T Qx + c^T x & & \\ \text{s.t.} & \|x\| & \le & 1 \\ & a_i^T x & \le & b_i \quad i = 1, 2 \end{array}$$

provided the two linear constraints are parallel:



$$\rightarrow \min \{\, x^T Qx + c^T x \,:\, l \le x_1 \le u,\ \|x\| \le 1 \,\} \qquad (*)$$

$$\begin{array}{rrcl} \text{restate as:} \quad \min & \displaystyle\sum_{i,j} q_{ij} X_{ij} & + & c^T x \quad \rightarrow \quad \boldsymbol{\sum_{i,j} q_{ij} x_i x_j \ + \ c^T x} \\ \text{s.t.} & X_{11} + lu & \le & (l+u)x_1 \\ & \|X_{.1} - lx\| & \le & x_1 - l \\ & \|ux - X_{.1}\| & \le & u - x_1 \\ & \displaystyle\sum_j X_{jj} \le 1 & & \quad \rightarrow \quad \boldsymbol{\sum_j x_{jj}^2 \ \le \ 1} \\ & X \succeq xx^T & & \end{array}$$

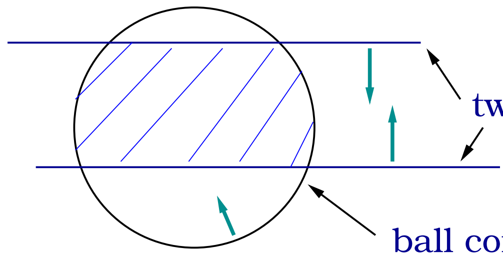
Equivalent to problem (*) ? Yes, if $\boldsymbol{X = xx^T}$, i.e. a **rank-1 solution**

Anstreicher-Burer (2012)

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

$$\begin{array}{rrcl} \min & x^TQx + c^Tx & & \\ & \text{s.t.} \quad \|x\| & \leq & 1 \\ & a_i^Tx & \leq & b_i \quad i = 1,2 \end{array}$$

provided the two linear constraints are parallel:



$$\longrightarrow \min\{\, x^TQx + c^Tx \,:\, l \leq x_1 \leq u,\ \|x\| \leq 1\,\} \qquad (*)$$

$$\begin{array}{lrcl} \text{restate as:} \quad \min & \sum_{i,j} q_{ij}X_{ij} & + & c^Tx \\ \text{s.t.} & X_{11} + lu & \leq & (l+u)x_1 \\ & \|X_{.1} - lx\| & \leq & x_1 - l \\ & \|ux - X_{.1}\| & \leq & u - x_1 \\ & \sum_j X_{jj} \leq 1 & & \\ & X \succeq xx^T & & \end{array}$$

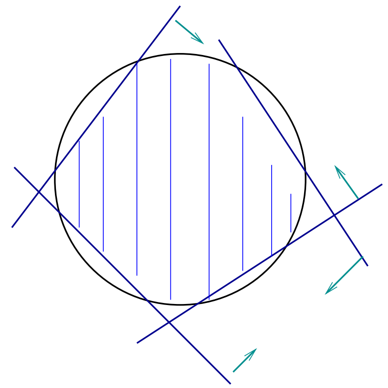
Lemma: This problem has an optimal solution with $X = xx^T$, i.e. a **rank-1** solution.

Burer-Yang (2012)

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

$$\begin{aligned} \min \quad & x^T Q x + c^T x \\ \text{s.t.} \quad & \|x\| \leq 1 \\ & a_i^T x \leq b_i \quad 1 \leq i \leq m \end{aligned}$$

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

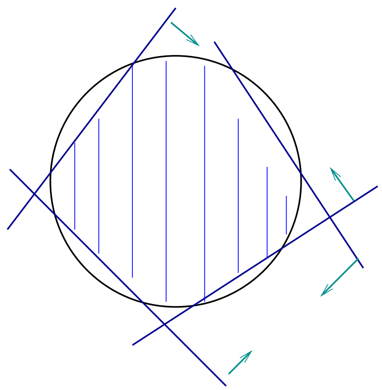


Burer-Yang (2012)

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

$$\begin{array}{rrcl} \min & x^TQx + c^Tx & & \\ & \text{s.t.} \quad \|x\| & \le & 1 \\ & a_i^Tx & \le & b_i \quad 1 \le i \le m \end{array}$$

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



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

$$\begin{array}{rrcl} \min & \displaystyle\sum_{i,j} q_{ij}X_{ij} & + & c^Tx \\ & \text{s.t.} \quad X_{11} + lu & \le & (l+u)x_1 \\ & \|b_ix - Xa_i\| & \le & b_i - a_i^Tx \qquad i \le m \\ & b_ib_j - b_ja_i^Tx - b_ia_j^Tx + a_i^TXa_j & \le & 0 \qquad i < j \le m \\ & \displaystyle\sum_j X_{jj} \le 1 & , & X \succeq xx^T \end{array}$$

Generalizations?

(B. and Alex Michalka, SODA 2014)

$$\begin{aligned} \min \quad & x^T Q x + c^T x \\ \text{s.t.} \quad & \|x - \mu_h\| \le r_h, \quad h \in S, \\ & \|x - \mu_h\| \ge r_h, \quad h \in K, \\ & x \in P \doteq \{ x \in \mathbb{R}^n \,:\, Ax \le b \} \end{aligned}$$

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

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

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

or

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

Generalizations?

(B. and Alex Michalka, SODA 2014)

$$
\begin{aligned}
\min \quad & x^T Q x + c^T x \\
\text{s.t.} \quad & \|x - \mu_h\| \le r_h, \quad h \in S, \\
& \|x - \mu_h\| \ge r_h, \quad h \in K, \\
& x \in P \doteq \{\, x \in \mathbb{R}^n \,:\, Ax \le b \,\}
\end{aligned}
$$

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

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

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

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?

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

Why not general QCQP?

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

$\longrightarrow$ form the semidefinite relaxation

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

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

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

Why not general QCQP?

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

$\longrightarrow$ form the semidefinite relaxation

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

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

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

Why do we call it a relaxation?

Why not general QCQP?

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

$\longrightarrow$ form the semidefinite relaxation

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

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

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

Why do we call it a relaxation?

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

Why not general QCQP?

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

$\longrightarrow$ form the semidefinite relaxation

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

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

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

Why do we call it a relaxation?

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

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

Why not general QCQP?

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

$\longrightarrow$ form the semidefinite relaxation

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

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

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

Why do we call it a relaxation?

Given $\boldsymbol{x}$ feasible for **QCQP**, the matrix $(\mathbf{1}, \boldsymbol{x}^T) \begin{pmatrix} \mathbf{1} \\ \boldsymbol{x} \end{pmatrix}$ 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**.

Why not general QCQP?

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

$\longrightarrow$ form the semidefinite relaxation

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

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

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

Why do we call it a relaxation?

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

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

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

Generalizations?

(B. and Alex Michalka, SODA 2014)

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

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:.

$$\begin{aligned} \min \quad & x^T Q x + c^T x \\ \text{s.t.} \quad & \|x - \mu\| \leq r \end{aligned}$$

The trust-region subproblem:.

$$\begin{array}{ll} \min & x^T Q x + c^T x \\ \text{s.t.} & \|x - \mu\| \le r \end{array}$$

Generalization: CDT (Celis-Dennis-Tapia) problem

$$\begin{array}{lrcl} \min & x^T Q_0 x + c_0^T x & & \\ \text{s.t.} & x^T Q_1 x + c_1^T x + d_1 & \le & 0 \\ & x^T Q_2 x + c_2^T x + d_2 & \le & 0 \end{array}$$

where $\boldsymbol{Q_1 \succ 0}$, $\boldsymbol{Q_2 \succ 0}$

Even more general than QCQPs

Barvinok (STOC 1992):

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

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

where the M_i are general matrices.

Even more general than QCQPs

Barvinok (STOC 1992):

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

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

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 $\boldsymbol{p \geq 1}$, there is a polynomial-time algorithm for deciding feasibility of a system

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

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: ϵ-feasibility

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

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

and given $\boldsymbol{0 < \epsilon < 1}$, either

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

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

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

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

A (better?) alternative: ϵ-feasibility

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

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

and given $\mathbf{0 < \boldsymbol{\epsilon} < 1}$, either

- **Prove** that the system is **infeasible**, or
- **Output** $\hat{\boldsymbol{x}} \in \mathbb{R}^{\boldsymbol{n}}$ with

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

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

Two issues: Constructiveness, and ϵ-feasibility

Modification to Barvinok's result

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

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

and given $\boldsymbol{0 < \epsilon < 1}$, either

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

- **Proves** that is $\boldsymbol{\epsilon}$-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 $\boldsymbol{p \geq 1}$, there is an algorithm that given a system

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

and given $\boldsymbol{0 < \epsilon < 1}$, either

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

- **Proves** that is $\boldsymbol{\epsilon}$-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

$$\begin{array}{lrcl} \min & f_0(x) & \doteq & x^T Q_0 x + c_0^T x \\ \text{s.t.} & x^T Q_i x + c_i^T x + d_i & \leq & 0 \quad 1 \leq i \leq m, \end{array}$$

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

(1) proves that the problem is infeasible,

or

(2) computes an ϵ-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

$$\begin{array}{lrcl} \min & f_0(x) & \doteq & x^T Q_0 x + c_0^T x \\ \text{s.t.} & x^T Q_i x + c_i^T x + d_i & \leq & 0 \quad 1 \leq i \leq m, \end{array}$$

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

(1) proves that the problem is infeasible,

or

(2) computes an ϵ-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 $\log \epsilon^{-1}$

$\longrightarrow$ Related algebraic geometry work by Grigoriev, Pasechnik, other Russians

Back to S-Lemma, +

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

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

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

Back to S-Lemma, +

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

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

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

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

Back to S-Lemma, +

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

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

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

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

in other words, **Hilbert (1888)**: iff exists $\gamma \geq \mathbf{0}$, $\boldsymbol{S}_{\mathbf{0}}(\boldsymbol{x})$ s.t. $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{S}_{\mathbf{0}}(\boldsymbol{x}) + \gamma \boldsymbol{g}(\boldsymbol{x})$

where $\boldsymbol{S}_{\mathbf{0}}(\boldsymbol{x})$ is a **sum of squares of polynomials**

Back to S-Lemma, +

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

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

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

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

in other words, **Hilbert (1888)**: iff exists $\gamma \geq \mathbf{0}$, $\boldsymbol{S}_0(\boldsymbol{x})$ s.t. $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{S}_0(\boldsymbol{x}) + \gamma \boldsymbol{g}(\boldsymbol{x})$

where $\boldsymbol{S}_0(\boldsymbol{x})$ is a **sum of squares of polynomials !** This paper started the field of algebraic geometry

Back to S-Lemma, +

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

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

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

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

in other words, **Hilbert (1888)**: iff exists $\gamma \ge 0,$ $S_0(x)$ 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 $Q_0(x), Q_1(x), \dots, Q_m(x)$ with $m \ge 2$, is it true that

$Q_0(x) \ge 0$ whenever $Q_i(x) \ge 0,\ 1 \le i \le m,$

iff exist $\gamma_i \ge 0$ s.t. $Q_0(x) \ge \sum_{i=1}^m \gamma_i Q_i(x)$ for all $x \in \mathbb{R}^n$?

Back to S-Lemma, +

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

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

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

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

$Q_0(x) \geq 0$ whenever $Q_i(x) \geq 0$, $1 \leq i \leq m$,

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 $\boldsymbol{f}, \boldsymbol{g} : \mathbb{R}^{\boldsymbol{n}} \longrightarrow \mathbb{R}$ be quadratic functions (degree ≤ 2 polynomials).

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

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

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

in other words, **Hilbert (1888)**: iff exists $\boldsymbol{\gamma} \geq \boldsymbol{0}$, $\boldsymbol{S}_0(\boldsymbol{x})$ s.t. $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{S}_0(\boldsymbol{x}) + \boldsymbol{\gamma g}(\boldsymbol{x})$

where $\boldsymbol{S}_0(\boldsymbol{x})$ 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 $\boldsymbol{Q}_0(\boldsymbol{x}), \boldsymbol{Q}_1(\boldsymbol{x}), \ldots, \boldsymbol{Q}_m(\boldsymbol{x})$ with $\boldsymbol{m} \geq \boldsymbol{2}$, is it true that

$\boldsymbol{Q}_0(\boldsymbol{x}) \geq \boldsymbol{0}$ whenever $\boldsymbol{Q}_i(\boldsymbol{x}) \geq \boldsymbol{0}$, $\boldsymbol{1} \leq \boldsymbol{i} \leq \boldsymbol{m}$,

iff exist $\boldsymbol{\gamma}_i \geq \boldsymbol{0}$, $\boldsymbol{S}_0(\boldsymbol{x})$, s.t. $\boldsymbol{Q}_0(\boldsymbol{x}) = \boldsymbol{S}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{\gamma}_i \boldsymbol{Q}_i(\boldsymbol{x})$, $\boldsymbol{S}_0(\boldsymbol{x}) =$ a sum of squares?

No.

Back to S-Lemma, +

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

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

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

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

$Q_0(x) \geq 0$ whenever $Q_i(x) \geq 0$, $1 \leq i \leq m$,

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?

No.

However*: $Q_0(x) > 0$ whenever $Q_i(x) \geq 0$, $1 \leq i \leq m$,

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).

Back to S-Lemma, +

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

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

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

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

in other words, **Hilbert (1888)**: iff exists $\gamma \geq \mathbf{0}$, $\boldsymbol{S}_0(\boldsymbol{x})$ s.t. $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{S}_0(\boldsymbol{x}) + \gamma \boldsymbol{g}(\boldsymbol{x})$

where $\boldsymbol{S}_0(\boldsymbol{x})$ 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 $\boldsymbol{Q}_0(\boldsymbol{x}), \boldsymbol{Q}_1(\boldsymbol{x}), \ldots, \boldsymbol{Q}_m(\boldsymbol{x})$ with $\boldsymbol{m} \geq \mathbf{2}$, is it true that

$\boldsymbol{Q}_0(\boldsymbol{x}) \geq \mathbf{0}$ whenever $\boldsymbol{Q}_i(\boldsymbol{x}) \geq \mathbf{0}, \ \mathbf{1} \leq \boldsymbol{i} \leq \boldsymbol{m}$,

iff exist $\gamma_i \geq \mathbf{0}$, $\boldsymbol{S}_0(\boldsymbol{x})$, s.t. $\boldsymbol{Q}_0(\boldsymbol{x}) = \boldsymbol{S}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \gamma_i \boldsymbol{Q}_i(\boldsymbol{x})$, $\boldsymbol{S}_0(\boldsymbol{x}) =$ a sum of squares?

No.

However*: $\boldsymbol{Q}_0(\boldsymbol{x}) > \mathbf{0}$ whenever $\boldsymbol{Q}_i(\boldsymbol{x}) \geq \mathbf{0}, \ \mathbf{1} \leq \boldsymbol{i} \leq \boldsymbol{m}$,

iff exist $\boldsymbol{S}_0(\boldsymbol{x}), \boldsymbol{S}_1(\boldsymbol{x}), \ldots \boldsymbol{S}_m(\boldsymbol{x})$ s.t. $\boldsymbol{Q}_0(\boldsymbol{x}) = \boldsymbol{S}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{S}_i(\boldsymbol{x}) \boldsymbol{Q}_i(\boldsymbol{x})$

where each $\boldsymbol{S}_i(\boldsymbol{x})$ 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** $\boldsymbol{P_0(x), G_1(x), \ldots, G_m(x)},\ \boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$,

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

More complete statement of Putinar's theorem – still abridged

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

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

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

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

where each $\boldsymbol{S_i(x)}$ is a sum of squares of polynomials

More complete statement of Putinar's theorem – still abridged

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

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

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

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

where each $\boldsymbol{S_i(x)}$ is a sum of squares of polynomials

Questions

- What are the $\boldsymbol{S_i(x)}$? Can we compute them efficiently?

More complete statement of Putinar's theorem – still abridged

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

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

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

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

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

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

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

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

where each $\boldsymbol{S_i(x)}$ is a sum of squares of polynomials

Questions

- What are the $\boldsymbol{S_i(x)}$? 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** $\boldsymbol{P_0(x), G_1(x), \ldots, G_m(x),}$ $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$,

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

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

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

where each $\boldsymbol{S_i(x)}$ is a sum of squares of polynomials

Questions

- What are the $\boldsymbol{S_i(x)}$? 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

$$\begin{array}{rl} \textbf{min} & \boldsymbol{P_0(x)} \\ \textbf{s.t.} & \boldsymbol{G_i(x) \ \geq \ 0, \qquad 1 \leq i \leq m}? \end{array}$$

$$
\begin{aligned}
\boldsymbol{P^*} \quad \doteq \quad \min \quad & \boldsymbol{P_0(x)} \\
\textbf{s.t.} \quad & \boldsymbol{G_i(x)} \;\geq\; \mathbf{0}, \qquad \mathbf{1} \leq \boldsymbol{i} \leq \boldsymbol{m}?
\end{aligned}
$$

$$\begin{array}{rlll} P^* \doteq & \min & P_0(x) & \\ & \text{s.t.} & G_i(x) \geq 0, & 1 \leq i \leq m? \end{array}$$

Idea: constrain the degrees of the sum-of-square "certificate" polynomials $S_i(x)$

$$\begin{aligned} P^* \doteq \min \quad & P_0(x) \\ \text{s.t.} \quad & G_i(x) \geq 0, \quad 1 \leq i \leq m? \end{aligned}$$

Idea: constrain the degrees of the sum-of-square "certificate" polynomials $S_i(x)$

Pick an integer $t > 0$, and define

$$\begin{aligned} P^{(t)} \doteq \quad & \sup \ \rho \\ \text{s.t.} \quad & P_0(x) - \rho = S_0(x) + \sum_{i=1}^{m} S_i(x) G_i(x) \\ & \text{each } S_i(x) \quad SOS \\ & \deg(S_0(x)) \leq 2t, \quad \deg(S_i(x) g_i(x)) \leq 2t. \end{aligned}$$

$$\begin{array}{rcll} \boldsymbol{P^*} & \doteq & \mathbf{min} & \boldsymbol{P_0(x)} \\ & & \textbf{s.t.} & \boldsymbol{G_i(x) \ \geq \ 0, \qquad 1 \leq i \leq m}? \end{array}$$

Idea: constrain the degrees of the sum-of-square "certificate" polynomials $\boldsymbol{S_i(x)}$

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

$$\boldsymbol{P^{(t)}} \ \doteq \ \mathbf{sup} \ \boldsymbol{\rho}$$

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

$$\text{each} \ \ \boldsymbol{S_i(x)} \quad \boldsymbol{SOS}$$

$$\mathbf{deg}(\boldsymbol{S_0(x)}) \ \leq \ \boldsymbol{2t}, \quad \mathbf{deg}(\boldsymbol{S_i(x) g_i(x)}) \ \leq \ \boldsymbol{2t}.$$

- $\boldsymbol{P^{(t)} \ \leq \ P^*}$
- $\boldsymbol{P^{(t)} \ \longrightarrow \ P^*}$ as $\boldsymbol{t} \longrightarrow +\infty$ (finite convergence)
- Does this help?

$$
\begin{aligned}
\boldsymbol{P^{(t)}} \;\doteq\; & \quad \textbf{sup}\;\; \boldsymbol{\rho} \\
\text{s.t.} \quad & \quad \boldsymbol{P_0(x) \;-\; \rho \;=\; S_0(x) \;+\; \sum_{i=1}^{m} S_i(x) G_i(x)} \\
& \quad \text{each} \;\; \boldsymbol{S_i(x)} \quad \boldsymbol{SOS} \\
& \quad \mathbf{deg}(\boldsymbol{S_0(x)}) \;\leq\; \boldsymbol{2t}, \quad \mathbf{deg}(\boldsymbol{S_i(x) g_i(x)}) \;\leq\; \boldsymbol{2t}.
\end{aligned}
$$

$$\boldsymbol{P}^{(t)} \doteq \quad \textbf{sup} \;\; \boldsymbol{\rho}$$

$$\text{s.t.} \quad \boldsymbol{P}_0(\boldsymbol{x}) - \boldsymbol{\rho} = \boldsymbol{S}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{S}_i(\boldsymbol{x})\, \boldsymbol{G}_i(\boldsymbol{x})$$

Here, **blue** polynomials are known, **black** polynomials are unknown

Example:

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

$$= \ \ \alpha x_1^3 \ + \ (2\alpha + \beta) x_1^2 x_2 +$$

$$2\beta x_1 x_2^2 \ + \ (\alpha + \gamma) x_1^2 \ + \ (\beta + 2\gamma) x_1 x_2 \ + \ \gamma x_1$$

$$
\begin{aligned}
\boldsymbol{P^{(t)}} \doteq \quad & \mathbf{sup}\ \boldsymbol{\rho} \\
\text{s.t.} \quad & \boldsymbol{P_0(x) - \rho = S_0(x) + \textstyle\sum_{i=1}^{m} S_i(x)\, G_i(x)} \\
& \text{each } \boldsymbol{S_i(x)} \quad \boldsymbol{SOS} \\
& \boldsymbol{\deg(S_0(x)) \le 2t, \quad \deg(S_i(x) g_i(x)) \le 2t}.
\end{aligned} \tag{2}
$$

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

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

$$
\begin{aligned}
P^{(t)} \doteq \quad & \sup \ \rho \\
\text{s.t.} \quad & P_0(x) - \rho = S_0(x) + \textstyle\sum_{i=1}^m S_i(x)G_i(x) \\
& \text{each } \ S_i(x) \quad SOS \\
& \deg(S_0(x)) \leq 2t, \quad \deg(S_i(x)g_i(x)) \leq 2t.
\end{aligned}
$$

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:

$$
\left(x_1^2 + 2x_1 + x_2\right)^2 = \left(x_1^2 + 2x_1 + x_2\right)\left(x_1^2 + 2x_1 + x_2\right)
$$

$$
\left(x_1^2 + 2x_1 + x_2\right) = \left(x_1^2 + 0x_2^2 + 0x_1x_2 + 2x_1 + x_2 + 0\right) =
$$

$$
(x_1^2,\ x_2^2,\ x_1x_2,\ x_1,\ x_2,\ 1)\begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix}
$$

$$\left(x_1^2 + 2x_1 + x_2\right) \quad = \quad \boldsymbol{(x_1^2,\ x_2^2,\ x_1 x_2,\ x_1,\ x_2,\ 1)} \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{2} \\ \mathbf{1} \\ \mathbf{0} \end{pmatrix}$$

$$\left(x_1^2 + 2x_1 + x_2\right) \quad = \quad (\boldsymbol{x_1^2},\ \boldsymbol{x_2^2},\ \boldsymbol{x_1 x_2},\ \boldsymbol{x_1},\ \boldsymbol{x_2},\ \mathbf{1}) \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{2} \\ \mathbf{1} \\ \mathbf{0} \end{pmatrix}$$

So $\color{blue}{\left(\boldsymbol{x_1^2} + \mathbf{2}\boldsymbol{x_1} + \boldsymbol{x_2}\right)^{\mathbf{2}}} = (\boldsymbol{x_1^2},\ \boldsymbol{x_2^2},\ \boldsymbol{x_1 x_2},\ \boldsymbol{x_1},\ \boldsymbol{x_2},\ \mathbf{1}) \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{2} \\ \mathbf{1} \\ \mathbf{0} \end{pmatrix} (\mathbf{1},\ \mathbf{0},\ \mathbf{0},\ \mathbf{2},\ \mathbf{1},\ \mathbf{0}) \begin{pmatrix} \boldsymbol{x_1^2} \\ \boldsymbol{x_2^2} \\ \boldsymbol{x_1 x_2} \\ \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \mathbf{1} \end{pmatrix} =$

$$\left(x_1^2 + 2x_1 + x_2\right) \;=\; \left(\boldsymbol{x_1^2},\ \boldsymbol{x_2^2},\ \boldsymbol{x_1x_2},\ \boldsymbol{x_1},\ \boldsymbol{x_2},\ \mathbf{1}\right)\begin{pmatrix}\mathbf{1}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{2}\\ \mathbf{1}\\ \mathbf{0}\end{pmatrix}$$

So $\left(\boldsymbol{x_1^2} + \mathbf{2}\boldsymbol{x_1} + \boldsymbol{x_2}\right)^{\mathbf{2}} = \left(\boldsymbol{x_1^2},\ \boldsymbol{x_2^2},\ \boldsymbol{x_1x_2},\ \boldsymbol{x_1},\ \boldsymbol{x_2},\ \mathbf{1}\right)\begin{pmatrix}\mathbf{1}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{2}\\ \mathbf{1}\\ \mathbf{0}\end{pmatrix}\left(\mathbf{1},\ \mathbf{0},\ \mathbf{0},\ \mathbf{2},\ \mathbf{1},\ \mathbf{0}\right)\begin{pmatrix}\boldsymbol{x_1^2}\\ \boldsymbol{x_2^2}\\ \boldsymbol{x_1x_2}\\ \boldsymbol{x_1}\\ \boldsymbol{x_2}\\ \mathbf{1}\end{pmatrix} =$

$$\left(\boldsymbol{x_1^2},\ \boldsymbol{x_2^2},\ \boldsymbol{x_1x_2},\ \boldsymbol{x_1},\ \boldsymbol{x_2},\ \mathbf{1}\right)\begin{pmatrix}\mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{4} & \mathbf{2} & \mathbf{0}\\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{x_1^2}\\ \boldsymbol{x_2^2}\\ \boldsymbol{x_1x_2}\\ \boldsymbol{x_1}\\ \boldsymbol{x_2}\\ \mathbf{1}\end{pmatrix}$$

$$\left(x_1^2 + 2x_1 + x_2\right) \;=\; \left(\boldsymbol{x_1^2,\ x_2^2,\ x_1x_2,\ x_1,\ x_2,\ 1}\right)\begin{pmatrix}\mathbf{1}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{2}\\ \mathbf{1}\\ \mathbf{0}\end{pmatrix}$$

So $\boldsymbol{\left(x_1^2 + 2x_1 + x_2\right)^2} = \left(\boldsymbol{x_1^2,\ x_2^2,\ x_1x_2,\ x_1,\ x_2,\ 1}\right)\begin{pmatrix}\mathbf{1}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{2}\\ \mathbf{1}\\ \mathbf{0}\end{pmatrix}\left(\mathbf{1,\ 0,\ 0,\ 2,\ 1,\ 0}\right)\begin{pmatrix}\boldsymbol{x_1^2}\\ \boldsymbol{x_2^2}\\ \boldsymbol{x_1x_2}\\ \boldsymbol{x_1}\\ \boldsymbol{x_2}\\ \mathbf{1}\end{pmatrix} =$

$$\left(\boldsymbol{x_1^2,\ x_2^2,\ x_1x_2,\ x_1,\ x_2,\ 1}\right)\begin{pmatrix}\mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{4} & \mathbf{2} & \mathbf{0}\\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{x_1^2}\\ \boldsymbol{x_2^2}\\ \boldsymbol{x_1x_2}\\ \boldsymbol{x_1}\\ \boldsymbol{x_2}\\ \mathbf{1}\end{pmatrix}$$

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

$$\left(x_1^2 + 2x_1 + x_2\right) \;=\; \left(\boldsymbol{x_1^2,\; x_2^2,\; x_1x_2,\; x_1,\; x_2,\; 1}\right)\begin{pmatrix}\mathbf{1}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{2}\\ \mathbf{1}\\ \mathbf{0}\end{pmatrix}$$

$$\text{So}\;\;\left(\boldsymbol{x_1^2 + 2x_1 + x_2}\right)^{\mathbf{2}} = \left(\boldsymbol{x_1^2,\; x_2^2,\; x_1x_2,\; x_1,\; x_2,\; 1}\right)\begin{pmatrix}\mathbf{1}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{2}\\ \mathbf{1}\\ \mathbf{0}\end{pmatrix}\left(\mathbf{1,\; 0,\; 0,\; 2,\; 1,\; 0}\right)\begin{pmatrix}\boldsymbol{x_1^2}\\ \boldsymbol{x_2^2}\\ \boldsymbol{x_1x_2}\\ \boldsymbol{x_1}\\ \boldsymbol{x_2}\\ \mathbf{1}\end{pmatrix} =$$

$$\left(\boldsymbol{x_1^2,\; x_2^2,\; x_1x_2,\; x_1,\; x_2,\; 1}\right)\begin{pmatrix}\mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{4} & \mathbf{2} & \mathbf{0}\\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{x_1^2}\\ \boldsymbol{x_2^2}\\ \boldsymbol{x_1x_2}\\ \boldsymbol{x_1}\\ \boldsymbol{x_2}\\ \mathbf{1}\end{pmatrix}$$

So: if a **given** polynomial $\boldsymbol{F(x_1, x_2)}$ is a sum of squares of quadratic polynomials in $\boldsymbol{x_1,\; x_2}$, then:

$$\boldsymbol{F(x_1, x_2)} \;=\; \left(\boldsymbol{x_1^2,\; x_2^2,\; x_1x_2,\; x_1,\; x_2,\; 1}\right), \text{ times a PSD matrix, times } \begin{pmatrix}\boldsymbol{x_1^2}\\ \boldsymbol{x_2^2}\\ \boldsymbol{x_1x_2}\\ \boldsymbol{x_1}\\ \boldsymbol{x_2}\\ \mathbf{1}\end{pmatrix}$$

Something different

Consider the optimization problem

$$\boldsymbol{f^*} \quad \doteq \quad \textbf{min} \;\; \boldsymbol{f(x)} \;\; \textbf{:} \;\; \boldsymbol{x \in K}$$

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

Something different

Consider the optimization problem

$$\boldsymbol{f^*} \quad \doteq \quad \mathbf{min} \;\; \boldsymbol{f(x)} \;\; \mathbf{:} \;\; \boldsymbol{x} \in \boldsymbol{K}$$

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

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

Something different

Consider the optimization problem

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

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

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

so if μ is a **measure** over K, i.e. $\int_K d\mu = 1$,

Something different

Consider the optimization problem

$$\boldsymbol{f^*} \quad \doteq \quad \min\ \boldsymbol{f(x)}\ :\ \boldsymbol{x} \in \boldsymbol{K}$$

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

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

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

Something different

Consider the optimization problem

$$\boldsymbol{f^* \quad \doteq \quad \min\ f(x)\ :\ x \in K}$$

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

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

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

and so

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

Something different

Consider the optimization problem

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

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

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

so if μ 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 $f(y) = f^*$,

Something different

Consider the optimization problem

$$\boldsymbol{f^*} \quad \doteq \quad \min \ \boldsymbol{f(x)} \ : \ \boldsymbol{x \in K}$$

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

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

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

and so

$$\boldsymbol{f^*} \quad \leq \quad \mathbf{inf}_{\boldsymbol{\mu}} \, \mathbb{E}_{\boldsymbol{\mu}} \, \boldsymbol{f(x)}$$

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

Something different

Consider the optimization problem

$$\boldsymbol{f^* \quad \doteq \quad \min\ f(x)\ :\ x \in K}$$

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

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

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

and so

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

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

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

Something different

Consider the optimization problem

$$\boldsymbol{f^* \quad \doteq \quad \min\ f(x)\ :\ x \in K}$$

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

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

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

and so

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

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

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

And so

$$\boldsymbol{f^* \quad = \quad \inf_{\mu} \mathbb{E}_{\mu}\, f(x)}$$

Something different

Consider the optimization problem

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

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

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

so if μ 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 δ_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

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

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.

Polynomial optimization

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

- 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.

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

Polynomial optimization

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.

We know $f_0^* = \inf_{\mu} \mathbb{E}_{\mu} f_0(x)$, 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

Polynomial optimization

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.

We know $f_0^* = \inf_{\mu} \mathbb{E}_{\mu} f_0(x)$, 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).

Polynomial optimization

Consider the polynomial optimization problem

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

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

- 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.

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

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

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 $\boldsymbol{\pi}$

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

Polynomial optimization

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

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

Polynomial optimization

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

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

So $\boldsymbol{f_0^*} \quad = \quad \mathbf{inf}_{\boldsymbol{y}}\ \sum_{\pi} \boldsymbol{a_{0,\pi} y_{\pi}}$, over all $\boldsymbol{K}$-moment vectors $\boldsymbol{y}$;

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

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

Polynomial optimization

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

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: $y_0 = 1$.

Polynomial optimization

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

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?

Polynomial optimization

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

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).

Polynomial optimization

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

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 $M[y] \doteq E_\mu v v^T$.

Polynomial optimization

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

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, $M[y]_{\pi,\rho} = \mathbb{E}_\nu\, x^\pi x^\rho = E_\nu x^{\pi+\rho} = y_{\pi+\rho}$

Polynomial optimization

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

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

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

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

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

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

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

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

Polynomial optimization

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

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,

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

Polynomial optimization

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

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

So $\boldsymbol{f_0^*} \quad = \quad \mathbf{inf}_{y} \sum_{\pi} \boldsymbol{a_{0,\pi} y_{\pi}}$, over all $\boldsymbol{K}$-moment vectors $\boldsymbol{y}$;

($\boldsymbol{y}$ 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 $\boldsymbol{\pi}$)

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

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

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

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

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

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

Polynomial optimization

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

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,

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

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

so

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

Polynomial optimization

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

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

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

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

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

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

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

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

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

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

so

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

A **finite-dimensional** semidefinite program!!

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

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

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

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

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

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

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

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

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

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

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

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

- Can be strengthened to account for the constraints $\boldsymbol{f_i(x) \geq 0}$.

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

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

$$\begin{array}{lrcl} \boldsymbol{f_0^*} & \geq & \min \sum_{\pi} a_{0,\pi}\, y_{\pi} & \\ \text{s.t.} & y_0 & = & 1, \\ & M & \succeq & 0, \\ & M_{\pi,\rho} & = & y_{\pi+\rho}, \\ & \multicolumn{3}{l}{\text{the zeroth row and column of } M \text{ both equal } y.} \end{array}$$

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

$$\begin{array}{lrcl} \boldsymbol{f_0^*} & \geq & \min \sum_{\pi} a_{0,\pi}\, y_{\pi} & \\ \text{s.t.} & y_0 & = & 1, \\ & \multicolumn{3}{l}{\text{the rows and columns of } M\text{, and the entries in } y\text{, indexed by tuples of size} \leq d} \\ & M & \succeq & 0, \\ & M_{\pi,\rho} & = & y_{\pi+\rho}, \quad \text{for all appropriate tuples } \pi, \rho \\ & \multicolumn{3}{l}{\text{the zeroth row and column of } M \text{ both equal } y} \end{array}$$

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

- Can be strengthened to account for the constraints $\boldsymbol{f_i(x) \geq 0}$.
- This is the level- $\boldsymbol{d}$ Lasserre relaxation (abridged).

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

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

$$\begin{array}{lll} \boldsymbol{f_0^*} & \geq & \min \sum_{\pi} a_{0,\pi}\, y_{\pi} \\ \text{s.t.} & & y_0 \; = \; 1, \\ & & M \; \succeq \; 0, \\ & & M_{\pi,\rho} \; = \; y_{\pi+\rho}, \\ & & \text{the zeroth row and column of } M \text{ both equal } y. \end{array}$$

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

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

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

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