Progress on solution of OPF problems

Daniel Bienstock, Columbia University
Optimal Power Flow

• Used to operate electrical power transmission systems ("grids").

• Primary goal: economic and secure operation.

• Run as often as every five minutes.

• Inputs for the computation: the current state of the grid, and estimates of demands ("loads") in the next time window.

• First proposed by Carpentier in 1962.
Brief introduction

- Grid modeled as a network; nodes = “buses”, edges = “lines”
- Steady-state operation: each bus $k$ has a voltage (potential energy) $V_k = |V_k|e^{j\theta_k}$
- Each line $\{k, m\}$ has physical attributes: e.g. resistance $r$, reactance $x$, shunt admittance $y^{sh}$
  
  $z = r + jx$, (series impedance)
  $y = z^{-1} = g + jb$, (admittance)
  $g = \frac{r}{r^2 + x^2}$ and $b = -\frac{x}{r^2 + x^2}$,

- A transformer with $N = \tau e^{j\sigma}$ scales voltages by $N$. 

![Diagram of a grid network with transformer and lines](diagram.png)
\(N = \tau e^{j\sigma}\)

\[
V = \begin{pmatrix} V_k \\ V_m \end{pmatrix} = \begin{pmatrix} |V_k| e^{j\theta_k} \\ |V_m| e^{j\theta_m} \end{pmatrix} = \begin{pmatrix} e_k + j f_k \\ e_m + j f_m \end{pmatrix} \quad \text{(voltages at } k \text{ and } m) 
\]

\[
I = \begin{pmatrix} I_{km} \\ I_{mk} \end{pmatrix} \quad \text{(complex current injections at } k \text{ and } m) 
\]

\[
S = \begin{pmatrix} S_{km} \\ S_{mk} \end{pmatrix} = \begin{pmatrix} P_{km} + j Q_{km} \\ P_{mk} + j Q_{mk} \end{pmatrix} \quad \text{(complex power injections at } k \text{ and } m) 
\]

Then

\[S_{km} = V_k I_{km}^*, \quad S_{mk} = V_m I_{mk}^* \quad \text{and} \quad I = \Im V,
\]

where

\[
\Im = \begin{pmatrix} (y + \frac{y^{sh}}{2}) \tau^2 & -y \frac{1}{\tau e^{-j\sigma}} \\ -y \frac{1}{\tau e^{j\sigma}} & y + \frac{y^{sh}}{2} \end{pmatrix}.
\]
\[ S_{km} = P_{km} + jQ_{km} \]

\[ P_{km} = |V_k|^2 g - |V_k||V_m|g \cos \theta_{km} - |V_k||V_m|b \sin \theta_{km} \]
(Active power injected by \( k \) into \( km \))

\[ Q_{km} = -|V_k|^2 b + |V_k||V_m|b \cos \theta_{km} - |V_k||V_m|b \sin \theta_{km} \]
(Reactive power injected by \( k \) into \( km \))

\( (\theta_{km} \doteq \theta_k - \theta_m) \)
\[ S_{km} = P_{km} + jQ_{km} \]

\[ P_{km} = |V_k|^2 g - |V_k||V_m| g \cos \theta_{km} - |V_k||V_m| b \sin \theta_{km} \]

(\text{active power injected by } k \text{ into } km)

\[ Q_{km} = -|V_k|^2 b + |V_k||V_m| b \cos \theta_{km} - |V_k||V_m| b \sin \theta_{km} \]

(\text{reactive power injected by } k \text{ into } km)

\[ (\theta_{km} = \theta_k - \theta_m) \]

\[ P_k = \sum_{km} P_{km} \quad \text{total active power injection by } k \]

\[ Q_k = \sum_{km} Q_{km} \quad \text{total reactive power injection by } k \]
OPF problem, simple version

Choose $|V_k|$ and $\theta_k$ for each bus $k$, so that

\[
\min \sum_{g \in \mathcal{G}} F_g(P_g)
\]

s.t. $L_k \leq P_k \leq U_k$ all $k$

$V_k^{min} \leq |V_k| \leq V_k^{max}$ all $k$

$|S_{km}| \leq S_{km}^{max}$ all $km$

$|\theta_{km}| \leq \theta_{km}^{max}$ all $km$, sometimes

$F_g$ convex quadratic, usually.
**OPF problem, simple version**

Choose $|V_k|$ and $\theta_k$ for each bus $k$, so that

$$\min \sum_{g \in G} F_g(P_g)$$

s.t. $L_k \leq P_k \leq U_k$ \hspace{1cm} all $k$

$$V_k^{min} \leq |V_k| \leq V_k^{max} \hspace{1cm} all \hspace{1cm} k$$

$$|S_{km}| \leq S_{km}^{max} \hspace{1cm} all \hspace{1cm} km$$

$$|\theta_{km}| \leq \theta_{km}^{max} \hspace{1cm} all \hspace{1cm} km, \text{ sometimes}$$

$F_g$ convex quadratic, usually.

In principle, this is a difficult, nonconvex optimization problem
How does the industry handle this problem?

- Techniques borrowed from convex optimization, i.e. logarithmic barrier methods
- Sequential linearization
- Other heuristics
- If everything fails, change the problem
- Some software is quite old
- Works very well on routine problems – runs in seconds
- May not work well on grids under distress
\[ P_{km} = |V_k|^2 g - |V_k||V_m| g \cos \theta_{km} - |V_k||V_m| b \sin \theta_{km} \]
\[ P_{km} = |V_k|^2 g - |V_k||V_m|g \cos \theta_{km} - |V_k||V_m|b \sin \theta_{km} \]

**Practical adaptation for routine operation**

- \( r = 0 \) for each line (zero resistance).
  So \( g = \frac{r}{r^2 + x^2} = 0, \quad b = -\frac{x}{r^2 + x^2} = -x^{-1} \)

- \(|V_k| = 1\) for all buses \( k \) (after scaling)

- \( \theta_k - \theta_m \approx 0 \) for all lines \( km \), so \( \sin(\theta_k - \theta_m) \approx \theta_k - \theta_m \)

- Only focus on active power
  \[ P_{km} = |V_k|^2 g - |V_k||V_m|g \cos \theta_{km} - |V_k||V_m|b \sin \theta_{km} \]
  \[ \approx \frac{\theta_k - \theta_m}{x} = y(\theta_k - \theta_m) \]
\[ P_{km} = |V_k|^2 g - |V_k||V_m|g \cos \theta_{km} - |V_k||V_m|b \sin \theta_{km} \]

**“DC Approximation”**

- \( r = 0 \) for each line (zero resistance).
  
  So \( g = \frac{r}{r^2 + x^2} = 0, \quad b = -\frac{x}{r^2 + x^2} = -x^{-1} \)

- \(|V_k| = 1\) for all buses \( k \) (after scaling)

- \( \theta_k - \theta_m \approx 0 \) for all lines \( km \), so \( \sin(\theta_k - \theta_m) \approx \theta_k - \theta_m \)

- Only focus on active power

\[
P_{km} = |V_k|^2 g - |V_k||V_m|g \cos \theta_{km} - |V_k||V_m|b \sin \theta_{km}
\]

\[
\approx \frac{\theta_k - \theta_m}{x} = y(\theta_k - \theta_m)
\]

So, get

\[
\min \sum_{g \in G} F_g(P_g)
\]

s.t.

\[
\sum_{km} y_{km}(\theta_k - \theta_m) = P_k \quad \text{all} \; k
\]

\[
L_k \leq P_k \leq U_k \quad \text{all} \; k, \quad |y_{km}(\theta_k - \theta_m)| \leq U_{km} \quad \text{all} \; km
\]
OPF using rectangular coordinates

\[ V = \begin{pmatrix} V_k \\ V_m \end{pmatrix} = \begin{pmatrix} |V_k| e^{j\theta_k} \\ |V_m| e^{j\theta_m} \end{pmatrix} = \begin{pmatrix} e_k + jf_k \\ e_m + jf_m \end{pmatrix} \quad \text{(voltages at } k \text{ and } m) \]

\[ I = \begin{pmatrix} I_{km} \\ I_{mk} \end{pmatrix} \quad \text{(complex current injections at } k \text{ and } m) \]

\[ S = \begin{pmatrix} S_{km} \\ S_{mk} \end{pmatrix} = \begin{pmatrix} P_{km} + jQ_{km} \\ P_{mk} + jQ_{mk} \end{pmatrix} \quad \text{(complex power injections at } k \text{ and } m) \]

Then

\[ S_{km} = V_k I_{km}^*, \quad S_{mk} = V_m I_{mk}^* \quad \text{and} \quad I = \mathbb{Y} V, \]

where

\[ \mathbb{Y} = \begin{pmatrix} \left( y + \frac{y^s}{2} \right) \frac{1}{\tau^2} & -y \frac{1}{\tau e^{-j\sigma}} \\ -y \frac{1}{\tau e^{j\sigma}} & y + \frac{y^s}{2} \end{pmatrix}. \]
**OPF using rectangular coordinates**

\[
V = \begin{pmatrix} V_k \\ V_m \end{pmatrix} = \begin{pmatrix} |V_k|e^{j\theta_k} \\ |V_m|e^{j\theta_m} \end{pmatrix} = \begin{pmatrix} e_k + jf_k \\ e_m + jf_m \end{pmatrix} \quad \text{(voltages at } k \text{ and } m)\
\]

\[
I = \begin{pmatrix} I_{km} \\ I_{mk} \end{pmatrix} \quad \text{(complex current injections at } k \text{ and } m)\
\]

\[
S = \begin{pmatrix} S_{km} \\ S_{mk} \end{pmatrix} = \begin{pmatrix} P_{km} + jQ_{km} \\ P_{mk} + jQ_{mk} \end{pmatrix} \quad \text{(complex power injections at } k \text{ and } m)\
\]

Then

\[
S_{km} = V_k I_{km}^*, \quad S_{mk} = V_m I_{mk}^* \quad \text{and} \quad I = \mathbb{Y}V,
\]

where

\[
\mathbb{Y} = \begin{pmatrix} (y + \frac{y^h}{2}) \frac{1}{\tau^2} & -y \frac{1}{\tau e^{-j\sigma}} \\ -y \frac{1}{\tau e^{j\sigma}} & y + \frac{y^h}{2} \end{pmatrix}.
\]

→ \(P_{km}\) and \(Q_{km}\) are **bilinear** functions of \(e_k, e_m, f_k, f_m\), e.g.

\[
P_{km} = e_k g(e_k - e_m) - e_k b(f_k - f_m) + f_k g(f_k - f_m) + f_k b(e_k - e_m)
\]

in the no shunt, no transformer case.
OPF in rectangular coordinates, simple case

Choose $|e_k|$ and $f_k$ for each bus $k$, so that

$$K^{OPF} = \min \sum_{g \in G} F_g(P_g)$$

s.t. $w^T A_k w = P_k$, \hspace{1cm} all $k$

$$w^T B_k w = Q_k, \hspace{1cm} all \hspace{1cm} k$$

box constraints on $P_k$, $Q_k$, for all $k$

$$V_{k}^{min} \leq w^T M_k w \leq V_{k}^{max} \hspace{1cm} all \hspace{1cm} k$$

Here $w = (e_1, e_2, \ldots, e_n, f_1, f_2, \ldots, f_n)^T$. 
OPF in rectangular coordinates, II

\[ K^{OPF} = \min \ w^T F w \]

s.t. \[ L_k \leq w^T A^k w \leq U_k, \quad k = 1, 2, \ldots m \]

\[ w \in \mathbb{R}^n. \]

Here, \( F \succeq 0. \)
OPF in rectangular coordinates, II

\[ K^{OPF} = \min w^T F w \]

s.t. \[ L_k \leq w^T A^k w \leq U_k, \quad k = 1, 2, \ldots m \]

\[ w \in \mathbb{R}^n. \]

Here, \( F \succeq 0 \). A quadratically constrained, quadratic program.
OPF in rectangular coordinates, II

\[ K^{OPF} = \min w^T F w \]

s.t. \[ L_k \leq w^T A^k w \leq U_k, \quad k = 1, 2, \ldots m \]

\[ w \in \mathbb{R}^n. \]

Here, \( F \succeq 0 \). A quadratically constrained, quadratic program.

Write \( W = w w^T \in \mathbb{R}^{n \times n} \). Then \( W \succeq 0 \), rank 1. So:

\[ K^{OPF} = \min \sum_{i,j} F_{ij} W_{ij} \]

s.t. \[ L_k \leq \sum_{i,j} A^{k}_{ij} W_{ij} \leq U_k, \quad k = 1, 2, \ldots m \]

A linear program?
OPF in rectangular coordinates, III

\[
K^{\text{OPF}} = \min w^T Fw
\]

s.t. \[L_k \leq w^T A^k w \leq U_k, \quad k = 1, 2, \ldots m\]

\[w \in \mathbb{R}^n.\]

Here, \(F \succeq 0\). A quadratically constrained, quadratic program.

Semidefinite relaxation:

\[
K^{\text{sdp}} = \min \sum_{i,j} F_{ij} W_{ij}
\]

s.t. \[L_k \leq \sum_{ij} A^k_{ij} W_{ij} \leq U_k, \quad k = 1, 2, \ldots m\]

\[W \succeq 0.\]

A relaxation \(\rightarrow\) proves a lower bound on \(K^{\text{OPF}}\).
OPF in rectangular coordinates, III

\[ K^{OPF} = \min \ w^T F w \]

s.t. \[ L_k \leq w^T A_k^w \leq U_k, \quad k = 1, 2, \ldots m \]

\[ w \in \mathbb{R}^n. \]

Here, \( F \succeq 0 \). A quadratically constrained, quadratic program.

Semidefinite relaxation:

\[ K^{sdp} = \min \ \sum_{i,j} F_{ij} W_{ij} \]

s.t. \[ L_k \leq \sum_{ij} A_{ij}^k W_{ij} \leq U_k, \quad k = 1, 2, \ldots m \]

\[ W \succeq 0. \]

A relaxation \( \rightarrow \) proves a lower bound on \( K^{OPF} \).

Relaxation is exact, i.e. \( K^{sdp} = K^{OPF} \) if optimal \( W \) has rank 1.
Interesting developments

→ Lavaei and Low (many coauthors), 2010 –
  • SDP relaxation is often exact (rank-1 solution)
Interesting developments

→ Lavaei and Low (many coauthors), 2010 –

• SDP relaxation is often exact (rank-1 solution)
• May require perturbing resistances to small positive values
Interesting developments

Lavaei and Low (many coauthors), 2010 –

• SDP relaxation is often exact (rank-1 solution)
• May require perturbing resistances to small positive values
• More nuanced view: the SDP solution is often of small rank
Interesting developments

→ Lavaei and Low (many coauthors), 2010 –
  • SDP relaxation is often exact (rank-1 solution)
  • May require perturbing resistances to small positive values
  • More nuanced view: the SDP solution is often of small rank
  • But $K^{sdp} \approx K^{OPF}$, often! (?)
  • Positive results may concern grids that are not under stress
  • And how do we extract a good, feasible rank-1 solution from the SDP solution?
Interesting developments

→ Lavaei and Low (many coauthors), 2010 –
  • SDP relaxation is often exact (rank-1 solution)
  • May require perturbing resistances to small positive values
  • More nuanced view: the SDP solution is often of small rank
  • But $K^{sdp} \approx K^{OPF}$, often! (?)
  • Positive results may concern grids that are not under stress
  • And how do we extract a good, feasible rank-1 solution from the SDP solution?
  • Despite quibbles, this is a very interesting development!
  • Many groups trying to “make it work”
  • Practical challenges?
Practical challenges?

- Solving large SDPs is not easy!
Practical challenges?

- Solving large SDPs is **not** easy!

- Matrix completion theorem (Laurent): SDPs involving sparse constraints can be kept sparse – but involves a combinatorial computation

- Lavaei, Molzahn, others(?) (2014) – Polish grid (about 3000 buses) can be solved in 300 - 900 seconds using matrix completion.
Practical challenges?

- Solving large SDPs is **not** easy!

- Matrix completion theorem (Laurent): SDPs involving sparse constraints can be kept sparse – but involves a combinatorial computation

- Lavaei, Molzahn, others (?) (2014) – Polish grid (about 3000 buses) can be solved in 300 - 900 seconds using matrix completion.

- And how about extracting a low rank near-optimal, feasible solution to OPF?

- And what do we do when there is a large *duality gap*:

  \[ K^{sdp} \ll K^{OPF} \]
Our current work

- Cutting-plane algorithms for computing linear approximations to lifted formulations to OPF

- Goal is to compute tight lower bounds, fast, with linear formulations

- Only linear (not conic) can be extended to handle important features, e.g. binary variables to model optional line switching or generator commitment

- Sample result: near optimal lower bound on Polish grid in \( \approx 5 \) seconds.
Quadratically constrained, quadratic programming:

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

Here,

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

is a general quadratic

Each \( M_i \) is \( n \times n \), wlog symmetric
Folklore result: QCQP is NP-hard
Folklore result: QCQP is NP-hard

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

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

s.t. $\sum_i w_i x_i = 0, -1 \leq x_i \leq 1, 1 \leq i \leq n.$
Folklore result: QCQP is NP-hard

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

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

\( W^* = -n \iff \text{there exists a subset } J \subseteq \{1, \ldots, n\} \text{ with } \sum_{j \in J} w_j = \sum_{j \notin J} w_j \)
Take any $\{-1, 1\}$-linear program
Take any \((-1, 1)\)-linear program
\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \in \{-1, 1\}^n.
\end{align*}
\]
Take any $\{-1, 1\}$-linear program

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

→

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

(and many other similar transformations)
Observation

Any instance of QCQP

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

with a \textbf{fixed} number of distinct bilinear terms can be solved in polynomial time.
Even more general

Solving systems of polynomial equations:

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

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. \( p_i(x) = 0, \forall i \)

**Example:** find a root for \( 3v^6w - v^4 + 7 = 0 \).
Even more general

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. $p_i(x) = 0$, $\forall i$

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

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

\[
\begin{align*}
c^2 &= 1 \\
v^2 - cv_2 &= 0 \\
v_2^2 - cv_4 &= 0 \\
v_2v_4 - cv_6 &= 0 \\
v_6w - cy &= 0 \\
3cy - cv_4 &= -7
\end{align*}
\]
Smale’s 17th problem

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

(but we are cheating)
Smale’s 17th problem

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

(but we are cheating)

- Approximately?
- On the average?
- Uniform algorithm?
“Approximately”

Q: How do practitioners and other lesser folk solve systems of nonlinear equations?
“Approximately”

Q: How do practitioners and other lesser folk solve systems of nonlinear equations?

A: Newton-Raphson, of course!

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

Q: How do practitioners and other lesser folk solve systems of nonlinear equations?

A: Newton-Raphson, of course!

→ If we start near a solution, quadratic convergence

“Approximate” solution to a system of polynomials:

a point in the region of quadratic convergence (to a solution)
Smale’s 17th problem

Can a zero of \( n \) polynomial equations on \( n \) unknowns be found \textbf{approximately}, \textcolor{red}{\textbf{on the average}} in polynomial time, with a \textbf{uniform} algorithm?

(but we are cheating)

- Approximately?
- \textbf{On the average}?
- Uniform 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
“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 run in polynomial time, on average, in that ball
Smale’s 17\textsuperscript{th} problem

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

(but we are cheating)

- Approximately?
- On the average?
- \textbf{Uniform algorithm}? When is an algorithm non-uniform?
Smale’s 17th problem

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

(but we are cheating)

- Approximately?
- On the average?
- **Uniform algorithm?** When is an algorithm non-uniform?

Blum, Shub, Smale (89), Blum, Cucker, Shub, Smale (98)

**First version:** A non-uniform algorithm specifies the existence of an algorithm for each input size.

As such, we cannot write a “program” that implements the algorithm.

It is more a proof of existence of an algorithm for each input size.
Smale’s 17\textsuperscript{th} problem

Can a zero of \( n \) polynomial equations on \( n \) unknowns be found \textbf{approximately}, \textbf{on the average} in polynomial time, with a \textbf{uniform} algorithm?

(but we are cheating)

- Approximately?
- On the average?
- \textbf{Uniform algorithm}? When is an algorithm non-uniform?

Blum, Shub, Smale (89), Blum, Cucker, Shub, Smale (98)
Bürgisser, Cucker (2012)

Second version: A \textbf{uniform algorithm}

- allows operations over real numbers
- at unit cost per operation
- with infinite precision
Smale’s 17th problem

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

(but we are cheating)

- Approximately?
- On the average?
- **Uniform algorithm?** When is an algorithm non-uniform?

Blum, Shub, Smale (89), Blum, Cucker, Shub, Smale (98)
Bürgisser, Cucker (2012)

**Second version:** A uniform algorithm

- allows operations over real numbers
- at unit cost per operation
- with infinite precision
Smale’s 17th problem

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

(but we are cheating)

- Approximately?
- On the average?
- Uniform algorithm? When is an algorithm non-uniform?

Blum, Shub, Smale (89), Blum, Cucker, Shub, Smale (98)
Bürgisser, Cucker (2012)

Second version: A uniform algorithm

- allows operations over real numbers
- at unit cost per operation
- with infinite precision
- Not! the usual bit-model of computation
Smale’s 17th problem

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

(but we are cheating)

- Beltrán and Pardo (2009) – a randomized (Las Vegas) uniform algorithm that computes an approximate zero in *expected* polynomial time
- Bürgisser, Cucker (2012) – a deterministic \( O(n^{\log \log n}) \) (uniform) algorithm for computing approximate zeros

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

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

(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) algo-
  rithm 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\textsuperscript{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?

(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

\begin{itemize}
  \item **Techniques:** Homotopy (path-following method solving a sequence of problems), Newton’s method
\end{itemize}

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

**So what can be done over the reals?**
Take any \([-1, 1]-linear program\)

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

\[
\rightarrow
\]

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

- Fixed number of linear constraints?
- Fixed number of quadratic constraints?
- Non-convex quadratic constraints?
The S-Lemma

Let \( f, g : \mathbb{R}^n \to \mathbb{R} \) be quadratic 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 if there exists \( \gamma \geq 0 \) such that

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

Yakubovich (1971), also much earlier, related work

**Corollary:** Can solve

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

in polynomial time (using semidefinite programming)

**Note:** duality may not hold if there is more than one quadratic constraint
Special case: the trust-region subproblem

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

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

Scale, rotate, translate:

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

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

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

How about extensions of the trust-region subproblem?

Where \( f(x) \) is a quadratic,

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

can be solved in polynomial time, as can

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


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

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

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

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

s.t. \  \ ||x|| \leq 1 \quad a_i^T x \leq b_i \quad i = 1, 2$$

provided the two linear constraints are parallel:
Anstreicher-Burer (2012)

In polynomial time, one can solve a problem of the form
\[
\min \ x^T Q x + c^T x \\
s.t. \quad \|x\| \leq 1 \\
a_i^T x \leq b_i \quad i = 1, 2
\]

provided the two linear constraints are parallel:

\[\rightarrow \min \{ x^T Q x + c^T x \ : \ l \leq x_1 \leq u, \ \|x\| \leq 1 \} \]
Anstreicher-Burer (2012)

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

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

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

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

provided the two linear constraints are parallel:

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

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

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

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

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

$$\sum_j X_{jj} \leq 1 \quad , \quad X \succeq xx^T$$

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

Also: Ye-Zhang
Burer-Yang (2012)

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

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

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

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

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

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

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

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

\( a_i^T x \leq b_i \quad 1 \leq i \leq m \)

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

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

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

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

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

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

\( \sum_j X_{jj} \leq 1 \quad X \succeq xx^T \)
This talk (B. and Alex Michalka, SODA 2014)

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

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

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

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

or

(2) $|S| = 0$ and the number of rows of $A$ is bounded.
This talk (B. and Alex Michalka, SODA 2014)

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

**Theorem.**

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

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

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

or

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

**Anstreicher-Burer:** Case (1) with 3 faces of $P$ meeting the feasible region.

**Burer-Yang:** Case (1) with $m + 1$ faces of $P$ meeting the feasible region.
More precise statement for case (1)

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

**Theorem.**
For each fixed $|S| \geq 1$, $|K|$ there is an algorithm that solves the problem, to tolerance $0 < \epsilon < 1$ in time

**(a)** Polynomial in the number of bits in the data and $\log \epsilon^{-1}$

**(b)** Linear in the number of faces of $P$ that intersect

\[
\bigcap_{h \in S} \{ x \in \mathbb{R}^n : \|x - \mu_h\| \leq r_h \}.
\]
Not hard **Lemma**

Given a collection of balls $B_h \subset \mathbb{R}^n$ ($h \in S$)

and a polyhedron

$$P = \{ x \in \mathbb{R}^n : Ax \leq b \},$$

there is an algorithm that lists the faces of $P$ that intersect $\bigcap_{h \in S} B_h$

In time

**(a)** polynomial in the number of bits in the data

**(b)** linear in the number of intersecting faces
Basic Idea

\[
\min \{ x^T Q x + c^T x : \| x - \mu_h \| \leq r_h, \ h \in S, \ \| x - \mu_h \| \geq r_h, \ h \in K, \ Ax \leq b \}
\]
Basic Idea

\[ \min \{ x^T Q x + c^T x : \| x - \mu_h \| \leq r_h, \ h \in S, \ \| x - \mu_h \| \geq r_h, \ h \in K, \ Ax \leq b \} \]

Let \( x^* \) be optimal. Trivial: there exist (possibly empty) subsets \( S^- \) of \( S \), \( K^- \) of \( K \), and \( I^- \) of the rows of \( Ax \leq b \), such that

\[
\| x^* - \mu_h \| = r_h \quad \forall \ h \in S^- \cup K^-, \quad a_i^T x^* = b_i \quad \forall \ i \in I^-
\]

\[
\| x^* - \mu_h \| < r_h \quad \forall \ h \in S - S^-, \quad \| x^* - \mu_h \| > r_h \quad \forall \ h \in K - K^-
\]

\[
a_i^T x^* < b_i \quad \forall \ i \notin I^-.
\]
Basic Idea

\[
\min \{ x^T Q x + c^T x : \| x - \mu_h \| \leq r_h, \ h \in S, \ \| x - \mu_h \| \geq r_h, \ h \in K, \ Ax \leq b \}
\]

Let \( x^* \) be optimal. Trivial: there exist (possibly empty) subsets

\( S^= \) of \( S \), \( K^= \) of \( K \), and \( I^= \) of the rows of \( Ax \leq b \), such that

\[
\| x^* - \mu_h \| = r_h \quad \forall \ h \in S^= \cup K^=, \quad a_i^T x^* = b_i \quad \forall \ i \in I^=
\]

\[
\| x^* - \mu_h \| < r_h \quad \forall \ h \in S - S^=, \quad \| x^* - \mu_h \| > r_h \quad \forall \ h \in K - K^=
\]

\[
a_i^T x^* < b_i \quad \forall \ i \notin I^=.
\]

\((S^=, K^=, I^=)\): an optimal triple.
Basic Idea

$$\min \{ x^T Q x + c^T x : \| x - \mu_h \| \leq r_h, \ h \in S, \ \| x - \mu_h \| \geq r_h, \ h \in K, \ Ax \leq b \}$$

Let $x^*$ be optimal. Trivial: there exist (possibly empty) subsets

$S^-$ of $S$, $K^-$ of $K$, and $I^-$ of the rows of $Ax \leq b$, such that

$$\| x^* - \mu_h \| = r_h \ \forall \ h \in S^- \cup K^-, \ a_i^T x^* = b_i \ \forall \ i \in I^-$$

$$\| x^* - \mu_h \| < r_h \ \forall \ h \in S - S^-, \ \| x^* - \mu_h \| > r_h \ \forall \ h \in K - K^- \ a_i^T x^* < b_i \ \forall \ i \notin I^-.$$

$(S^-, K^-, I^-)$: an optimal triple. $x^*$: tight for $(S^-, K^-, I^-)$
Basic Idea

\[
\begin{align*}
\min \{ x^T Q x + c^T x : \|x - \mu_h\| &\leq r_h, \ h \in S, \quad \|x - \mu_h\| \geq r_h, \ h \in K, \quad Ax \leq b \}\end{align*}
\]

Let \( x^* \) be optimal. Trivial: there exist (possibly empty) subsets

\( S^= \) of \( S \), \( K^= \) of \( K \), and \( I^= \) of the rows of \( Ax \leq b \), such that

\[
\begin{align*}
\|x^* - \mu_h\| &= r_h \quad \forall h \in S^= \cup K^=, \quad a_i^T x^* = b_i \quad \forall i \in I^= \\
\|x^* - \mu_h\| &< r_h \quad \forall h \in S - S^=, \quad \|x^* - \mu_h\| > r_h \quad \forall h \in K - K^= \\
a_i^T x^* &< b_i \quad \forall i \notin I^=.
\end{align*}
\]

\((S^=, K^=, I^=):\) an optimal triple. \( x^* : \text{tight} \) for \((S^=, K^=, I^=)\)

Algorithm will guess \((S^=, K^=, I^=)\) (actually, compute \( I^=\)).
Basic Idea

\[ \min \{ x^T Q x + c^T x : \| x - \mu_h \| \leq r_h, \ h \in S, \ \| x - \mu_h \| \geq r_h, \ h \in K, \ Ax \leq b \} \]

Let \( x^* \) be optimal. Trivial: there exist (possibly empty) subsets

\( S^= \) of \( S \), \( K^= \) of \( K \), and \( I^= \) of the rows of \( Ax \leq b \), such that

\[
\| x^* - \mu_h \| = r_h \quad \forall \ h \in S^= \cup K^=, \quad a_i^T x^* = b_i \quad \forall \ i \in I^=
\]

\[
\| x^* - \mu_h \| < r_h \quad \forall \ h \in S - S^=, \quad \| x^* - \mu_h \| > r_h \quad \forall \ h \in K - K^=\]

\[ a_i^T x^* < b_i \quad \forall \ i \notin I^= . \]

\( (S^=, K^=, I^=) \): an optimal triple. \( x^* \): tight for \( (S^=, K^=, I^=) \)

Algorithm will guess \( (S^=, K^=, I^=) \) (actually, compute \( I^= \)).

For each enumerated triple \( (\hat{S}, \hat{K}, \hat{I}) \), it will (in polynomial time) either

(a) Compute a finite set of vectors tight for \( (\hat{S}, \hat{K}, \hat{I}) \), one of which must be \( x^* \) if the guess is right, or
**Basic Idea**

\[ \min \{ x^T Q x + c^T x : \| x - \mu_h \| \leq r_h, \ h \in S, \ \| x - \mu_h \| \geq r_h, \ h \in K, \ Ax \leq b \} \]

Let \( x^* \) be optimal. Trivial: there exist (possibly empty) subsets

\[ S^= \] of \( S, \ K^= \] of \( K, \] and \( I^= \] of the rows of \( Ax \leq b, \] such that

\[
\| x^* - \mu_h \| = r_h \ \forall \ h \in S^= \cup K^=, \ \ a_i^T x^* = b_i \ \forall \ i \in I^= \\
\| x^* - \mu_h \| < r_h \ \forall \ h \in S - S^=, \ \| x^* - \mu_h \| > r_h \ \forall \ h \in K - K^= \\
a_i^T x^* < b_i \ \forall \ i \notin I^=.
\]

\((S^=, \ K^=, \ I^=)\): an optimal triple. \( x^* \): tight for \((S^=, \ K^=, \ I^=)\)

Algorithm will guess \((S^=, \ K^=, \ I^=)\) (actually, compute \( I^= \)).

For each enumerated triple \((\hat{S}, \hat{K}, \hat{I})\), it will (in polynomial time) either

(a) Compute a finite set of vectors tight for \((\hat{S}, \hat{K}, \hat{I})\), one of which must be \( x^* \) if the guess is right, or

(b) Prove that if \((\hat{S}, \hat{K}, \hat{I})\) is optimal, there is a different optimal triple \((\tilde{S}, \tilde{K}, \tilde{I})\) with

\[ \tilde{S} \supseteq \hat{S}, \ \tilde{K} \supseteq \hat{K}, \ \tilde{I} \supseteq \hat{I} \] and \[ |\tilde{S}| + |\tilde{K}| + |\tilde{I}| > |\hat{S}| + |\hat{K}| + |\hat{I}|. \]
Geometry, 1

Notation. Given a ball $B = \{ x \in \mathbb{R}^n : \| x - \hat{\mu}_i \| \leq \hat{r} \}$,

$$\partial B \triangleq \{ x \in \mathbb{R}^n : \| x - \hat{\mu}_i \| = \hat{r} \}$$
Geometry, 1

Notation. Given a ball $B = \{ x \in \mathbb{R}^n : \| x - \hat{\mu}_i \| \leq \hat{r} \}$,

$$\partial B \doteq \{ x \in \mathbb{R}^n : \| x - \hat{\mu}_i \| = \hat{r} \}$$

Lemma. Let $B_i = \{ x \in \mathbb{R}^n : \| x - \mu_i \| \leq r_i \}$, $i = 1, 2$, be distinct and intersecting.
Geometry, 1

Notation. Given a ball $B = \{ x \in \mathbb{R}^n : \| x - \hat{\mu}_i \| \leq \hat{r} \}$,

$$\partial B \doteq \{ x \in \mathbb{R}^n : \| x - \hat{\mu}_i \| = \hat{r} \}$$

Lemma. Let $B_i = \{ x \in \mathbb{R}^n : \| x - \mu_i \| \leq r_i \}, i = 1, 2$, be distinct and intersecting.

There exists an $(n-1)$-dim hyperplane $H$, a point $v \in H$, and $r \geq 0$ such that

$$\partial B_1 \cap \partial B_2 = \{ x \in H : \| x - v \| = r \}$$

and

$$\partial B_i \cap H = \{ x \in H : \| x - v \| = r \}, \quad i = 1, 2$$
Geometry, 1

Corollary Given balls $B_i$, $i \in I$, not all equal, with

$$\bigcap_{i \in I} B_i \neq \emptyset,$$

there exists an $(n - t)$-dim hyperplane $H$ ($t \geq 1$), $v \in H$ and $r \geq 0$

s.t.

$$\bigcap_{i \in I} \partial B_i = \{x \in H : \|x - v\| = r\}$$
Corollary  Given balls $B_i, i \in I$, not all equal, with

$$\bigcap_{i \in I} B_i \neq \emptyset,$$

there exists an $(n - t)$-dim hyperplane $H \ (t \geq 1)$, \ $v \in H$ and \ $r \geq 0$ s.t.

$$\bigcap_{i \in I} \partial B_i = \{ x \in H : \| x - v \| = r \}$$

Implication: When guessing an optimal triple \ $(S^=, K^=, I^=)$

$$\| x^* - \mu_h \| = r_h \ \forall \ h \in S^= \cup K^=, \quad a_i^T x^* = b_i \ \forall \ i \in I^=$$

$$\| x^* - \mu_h \| < r_h \ \forall \ h \in S - S^=, \quad \| x^* - \mu_h \| > r_h \ \forall \ h \in K - K^=$$

$$a_i^T x^* < b_i \ \forall \ i \notin I^=.$$
Geometry, 1

Corollary Given balls $B_i, i \in I$, not all equal, with

$$\bigcap_{i \in I} B_i \neq \emptyset,$$

there exists an $(n - t)$-dim hyperplane $H$ ($t \geq 1$), $v \in H$ and $r \geq 0$ s.t.

$$\bigcap_{i \in I} \partial B_i = \{x \in H : \|x - v\| = r\}$$

Implication: When guessing an optimal triple $(S^\approx, K^\approx, I^\approx)$

$$\|x^* - \mu_h\| = r_h \quad \forall h \in S^\approx \cup K^\approx, \quad a_i^T x^* = b_i \quad \forall i \in I^\approx$$

$$\|x^* - \mu_h\| < r_h \quad \forall h \in S - S^\approx, \quad \|x^* - \mu_h\| > r_h \quad \forall h \in K - K^\approx$$

$$a_i^T x^* < b_i \quad \forall i \notin I^\approx.$$

we

(1) Restrict to a lower dimensional space

(2) Obtain a single, binding, ball constraint
The original problem:

\[
\begin{align*}
\min & \quad x^T Q x + c^T x \\
\text{s.t.} & \quad \| x - \mu_h \| \leq r_h, \quad h \in S, \\
& \quad \| x - \mu_h \| \geq r_h, \quad h \in K, \\
& \quad a_i^T x \leq b_i, \quad i \in I
\end{align*}
\]
The original problem:

\[
\begin{align*}
\min & \quad x^T Q x + c^T x \\
\text{s.t.} & \quad \|x - \mu_h\| \leq r_h, \quad h \in S, \\
& \quad \|x - \mu_h\| \geq r_h, \quad h \in K, \\
& \quad a_i^T x \leq b_i, \quad i \in I
\end{align*}
\]

Given a guess, this becomes (ignoring the non-binding constraints):

\[
\begin{align*}
\min & \quad x^T Q x + c^T x \\
\text{s.t.} & \quad \|x - \hat{\mu}\| = \hat{r}, \\
& \quad x \in H
\end{align*}
\]
The original problem:

\[
\begin{align*}
\min & \quad x^T Q x + c^T x \\
\text{s.t.} & \quad \|x - \mu_h\| \leq r_h, \quad h \in S, \\
& \quad \|x - \mu_h\| \geq r_h, \quad h \in K, \\
& \quad a_i^T x \leq b_i, \quad i \in I
\end{align*}
\]

Given a guess, this becomes (ignoring the non-binding constraints):

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

Almost correct: first-order condition restricted to \( H \)
The original problem:

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

s.t. $\|x - \mu_h\| \leq r_h$, $h \in S$,

$\|x - \mu_h\| \geq r_h$, $h \in K$,

$a_i^T x \leq b_i$, $i \in I$

Given a guess, this becomes (ignoring the non-binding constraints):

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

s.t. $\|x - \hat{\mu}\| = \hat{r}$,

$x \in H$

Almost correct: first-order condition restricted to $H$

Better: Use projected quadratic representation
Theorem (abridged).

Given a triple \((\hat{S}, \hat{K}, \hat{I})\) there is polynomially computable list of points \(x^j, \ (j \in J)\) tight for the triple, such that if \((\hat{S}, \hat{K}, \hat{I})\) is optimal, then either
**Theorem** (abridged).

Given a triple \((\hat{S}, \hat{K}, \hat{I})\) there is polynomially computable list of points \(x^j, (j \in J)\) tight for the triple, such that if \((\hat{S}, \hat{K}, \hat{I})\) is optimal, then either

\[(1) \quad x^* = x^j \text{ for some } j \in J, \text{ or} \]

**Theorem** (abridged).

Given a triple \((\hat{S}, \hat{K}, \hat{I})\) there is polynomially computable list of points \(x^j, (j \in J)\) tight for the triple, such that if \((\hat{S}, \hat{K}, \hat{I})\) is optimal, then either

(1) \(x^* = x^j\) for some \(j \in J\), or

(2) There exists infeasible \(y\) and a Jordan curve \(\Theta\) joining \(y\) and \(x^*\), s.t.

\[
z^TQz + c^Tz = x^*^TQx^* + c^Tx^* \quad \forall \ z \in \Theta
\]

\(z\) tight for \((\hat{S}, \hat{K}, \hat{I})\) \(\forall \ z \in \Theta\)
**Theorem** (abridged).

Given a triple \((\hat{S}, \hat{K}, \hat{I})\) there is polynomially computable list of points \(x^j, \ (j \in J)\) tight for the triple, such that if \((\hat{S}, \hat{K}, \hat{I})\) is optimal, then either

(1) \(x^* = x^j\) for some \(j \in J\), or

(2) There exists infeasible \(y\) and a Jordan curve \(\Theta\) joining \(y\) and \(x^*\), s.t.

\[
z^TQz + c^Tz = x^*^TQx^* + c^Tx^* \quad \forall \ z \in \Theta
\]

\(z\) tight for \((\hat{S}, \hat{K}, \hat{I})\) \(\forall \ z \in \Theta\)

**Implication:** In case (2), there is a different optimal triple \((\tilde{S}, \tilde{K}, \tilde{I})\) with

\[
\tilde{S} \supseteq \hat{S}, \ \tilde{K} \supseteq \hat{K}, \ \tilde{I} \supseteq \hat{I} \quad \text{and} \quad |\tilde{S}| + |\tilde{K}| + |\tilde{I}| > |\hat{S}| + |\hat{K}| + |\hat{I}|.
\]
**Theorem** (abridged).

Given a triple \((\hat{S}, \hat{K}, \hat{I})\) there is polynomially computable list of points \(x^j, (j \in J)\) tight for the triple, such that if \((\hat{S}, \hat{K}, \hat{I})\) is optimal, then either

(1) \(x^* = x^j\) for some \(j \in J\), or

(2) There exists infeasible \(y\) and a Jordan curve \(\Theta\) joining \(y\) and \(x^*\), s.t.

\[
z^TQz + c^Tz = x^*^TQx^* + c^Tx^* \quad \forall z \in \Theta
\]

\(z\) tight for \((\hat{S}, \hat{K}, \hat{I})\) \(\forall z \in \Theta\)

**Implication:** In case (2), there is a different optimal triple \((\tilde{S}, \tilde{K}, \tilde{I})\) with

\[\tilde{S} \supseteq \hat{S}, \tilde{K} \supseteq \hat{K}, \tilde{I} \supseteq \hat{I}\]

and \(|\tilde{S}| + |\tilde{K}| + |\tilde{I}| > |\hat{S}| + |\hat{K}| + |\hat{I}|\).
The trust-region subproblem:

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

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

Generalization: CDT (Celis-Dennis-Tapia) problem

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

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

Barvinok (STOC 1992):

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

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

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

Barvinok (STOC 1992):

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

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

where the $M_i$ are general matrices.

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

- **Computational model?** Uniform algorithm? “Real-RAM”?
A (better?) alternative: $\epsilon$-feasibility

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

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

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

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

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

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

$$-\epsilon \leq x^T M_i \leq \epsilon, \quad 1 \leq i \leq p,$$

$$1 - \epsilon \leq \|\hat{x}\| \leq 1 + \epsilon,$$

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

A (better?) alternative: $\epsilon$-feasibility

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(so still nonconstructive)
Modification to Barvinok’s result

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

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

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

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

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

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

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

(so still nonconstructive)

Assuming such an algorithm exists ...
Theorem (2014).

Assume that an algorithm for $\epsilon$-feasibility as indicated above exists.
**Theorem** (2014).

Assume that an algorithm for $\epsilon$-feasibility as indicated above exists.

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

$$\min \quad f_0(x) = x^T Q_0 x + c_0^T x$$

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

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

(1) proves that the problem is infeasible,

or

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

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

The best optimization journal!