Some results on polynomial optimization problems

Daniel Bienstock, Columbia University

QCQP:

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

Here,

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

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

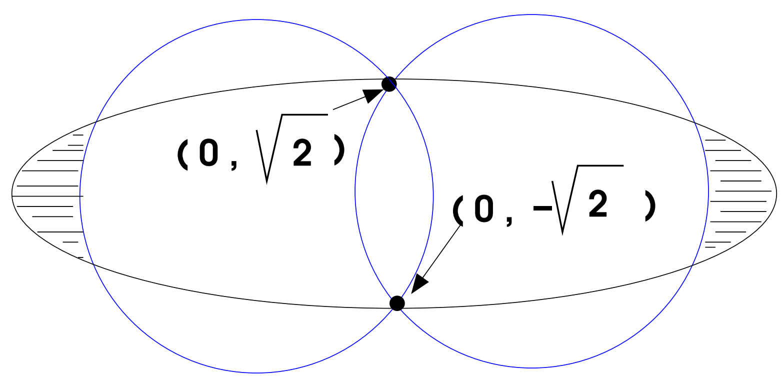
Folklore result: QCQP is Strongly NP-hard

A simple example

$$
\begin{aligned}
\max \quad & x_2 \\
\text{s.t.} \quad & (x_1 - 1)^2 + x_2^2 \geq 3 \\
& (x_1 + 1)^2 + x_2^2 \geq 3 \\
& \frac{x_1^2}{10} + x_2^2 \leq 2
\end{aligned}
$$

A simple example

$$
\begin{aligned}
& \max \quad x_2 \\
\text{s.t.} \quad & (x_1-1)^2 + x_2^2 \geq 3 \\
& (x_1+1)^2 + x_2^2 \geq 3 \\
& \frac{x_1^2}{10} + x_2^2 \leq 2
\end{aligned}
$$



CDT (Celis-Dennis-Tapia) problem

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

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

CDT (Celis-Dennis-Tapia) problem

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

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

Generalization of the trust-region subproblem:

$$\begin{aligned} \min \quad & x^T Q x + c^T x \\ \text{s.t.} \quad & \|x - \mu\|^2 \le r^2 \end{aligned}$$

which is solvable using many techniques

Theorem (Barvinok, 1993)

For each fixed integer $\boldsymbol{p}$ there is a polynomial-time algorithm that given a system

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

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

correctly determines feasibility.

$\rightarrow$ nonconstructive.

Weakening of Barvinok's theorem

For each fixed $\boldsymbol{p \geq 1}$, 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 $\boldsymbol{0 < \epsilon < 1}$, 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)

Theorem (SIOPT, forthcoming).

For each fixed $m \geq 1$ there is an algorithm that given

$$\begin{aligned} \min \quad & f_0(x) \doteq x^T A_0 x + c_0^T x \\ \text{s.t.} \quad & x^T A_i x + c_i^T x + d_i \leq 0 \quad 1 \leq i \leq m, \end{aligned}$$

where $A_1 \succ 0$, 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 $f_0(x) < f(\hat{x}) - \epsilon$

in time polynomial in the number of bits in the data and $\log \epsilon^{-1}$

Sketch:

Given a system

$$\boldsymbol{x^T A_i x + c_i^T x + d_i \ \le \ 0 \quad 1 \le i \le m,}$$

where $\boldsymbol{A_1} \succ \boldsymbol{0}$, how to prove infeasibility or feasibility?

Assume

$$x^T A_1 x + c_1^T x + d_1 \ = \ \|x\|^2 - 1,$$

and $|\boldsymbol{f_i(x)}| \le \boldsymbol{U_i}$.

Sketch:

Given a system

$$\boldsymbol{x^T A_i x + c_i^T x + d_i \;\le\; 0 \quad 1 \le i \le m,}$$

with $\boldsymbol{x^T A_1 x + c_1^T x + d_1 \;=\; \|x\|^2 - 1}$, and $\boldsymbol{|f_i(x)| \le U_i}$.

$$\begin{aligned} x^T A_i x + c_i^T v_0 x + d_i v_0^2 + s_i^2 &= 0 \qquad 1 \le i \le m, && \text{(1a)} \\ \frac{s_i^2 + w_i^2}{U_i} - v_0^2 &= 0 \qquad 2 \le i \le m, && \text{(1b)} \\ \|x\|^2 + s_1^2 + \sum_{i=2}^{m} \frac{s_i^2 + w_i^2}{U_i} + v_0^2 &= m + 1. && \text{(1c)} \end{aligned}$$

$$
\begin{aligned}
x^T A_i x + c_i^T v_0 x + d_i v_0^2 + s_i^2 &= 0 \qquad 1 \le i \le m, \quad \text{(2a)} \\
\frac{s_i^2 + w_i^2}{U_i} - v_0^2 &= 0 \qquad 2 \le i \le m, \quad \text{(2b)} \\
\|x\|^2 + s_1^2 + \sum_{i=2}^{m} \frac{s_i^2 + w_i^2}{U_i} + v_0^2 &= m + 1. \qquad \text{(2c)}
\end{aligned}
$$

→ (2a) for $i = 1$ is $\|x\|^2 - v_0^2 + s_1^2 \;=\; 0.$

Adding it and all of (2b) yields

$$
\|x\|^2 + s_1^2 + \sum_{i=2}^{m} \frac{s_i^2 + w_i^2}{U_i} - m v_0^2 \;=\; 0
$$

Together with (2c) this implies $\boldsymbol{v_0^2 = 1}$.

If $\boldsymbol{v_0 = 1}$ then (2a) means that x is feasible.

New result on "true" version of CDT problem

$$\begin{aligned} \min \quad & x^T Q_0 x + c_0^T x \\ \text{s.t.} \quad & x^T Q_i x + c_i^T x + d_i \;\leq\; 0, \qquad i = 1, 2 \end{aligned}$$

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

Sakaue, Nakatsukasa, Takeda, Iwata (2015); "simple" algorithm.

Assume KKT conditions hold.

$$
\begin{aligned}
H(\lambda_1, \lambda_2)x &= y \\
x^T Q_i x + c_i^T x + d_i &\leq 0, \qquad i = 1, 2 \\
\lambda_i (x^T Q_i x + c_i^T x + d_i) &= 0, \qquad i = 1, 2 \\
\lambda_i &\geq 0, \qquad i = 1, 2
\end{aligned}
$$

Here

$$
\begin{aligned}
H &\doteq Q_0 + \lambda_1 Q_1 + \lambda_2 Q_2 \\
y &\doteq -(c_0 + \lambda_1 c_1 + \lambda_2 c_2)
\end{aligned}
$$

1. Compute a polynomially large set of candidates for λ_1, λ_2.

2. Given λ_1, λ_2, solve $Hx = y$ to obtain x.

$$\lambda_i(x^T Q_i x + c_i^T x + d_i) \;=\; 0, \qquad i = 1, 2$$

is equivalent to

$$\lambda_i \det \begin{bmatrix} Q_i & -H & c_i \\ -H & 0 & y \\ c_i^T & y^T & d_i \end{bmatrix} \;=\; 0$$

So, two determinantal equations

$$\lambda_1 \det M_1(\lambda_1, \lambda_2) \;=\; \lambda_2 \det M_2(\lambda_1, \lambda_2) \;=\; 0.$$

$$\lambda_i(x^T Q_i x + c_i^T x + d_i) \;=\; 0, \qquad i = 1, 2$$

is equivalent to

$$\lambda_i \det \begin{bmatrix} Q_i & -H & c_i \\ -H & 0 & y \\ c_i^T & y^T & d_i \end{bmatrix} \;=\; 0$$

So, two determinantal equations

$$\lambda_1 \det M_1(\lambda_1, \lambda_2) \;=\; \lambda_2 \det M_2(\lambda_1, \lambda_2) \;=\; 0.$$

Recall $H \;=\; Q_0 + \lambda_1 Q_1 + \lambda_2 Q_2, \quad y \;=\; -(c_0 + \lambda_1 c_1 + \lambda_2 c_2)$

$$\lambda_i(x^T Q_i x + c_i^T x + d_i) \;=\; 0, \qquad i = 1, 2$$

is equivalent to

$$\lambda_i \det \begin{bmatrix} Q_i & -H & c_i \\ -H & 0 & y \\ c_i^T & y^T & d_i \end{bmatrix} \;=\; 0$$

So, two determinantal equations

$$\lambda_1 \det M_1(\lambda_1, \lambda_2) \;=\; \lambda_2 \det M_2(\lambda_1, \lambda_2) \;=\; 0.$$

Theorem: If the two equations hold then: $\det B(\lambda_1) \;=\; 0.$

Here, $\boldsymbol{B}$, of the form $\boldsymbol{\lambda_1 E + F}$, is the **Bézoutian**.

$\boldsymbol{B}$ is $\boldsymbol{n^2 \times n^2}$.

Smale's 17^{th} problem

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

- 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?

ACOPF

Input: an undirected graph G.

- For every vertex i, **two** variables: $\boldsymbol{e_i}$ and $\boldsymbol{f_i}$
- For every edge $\{k, m\}$, **four** (specific) quadratics:

$$H^P_{k,m}(e_k, f_k, e_m, f_m), \quad H^Q_{k,m}(e_k, f_k, e_m, f_m)$$

$$H^P_{m,k}(e_k, f_k, e_m, f_m), \quad H^Q_{m,k}(e_k, f_k, e_m, f_m)$$

$$
\begin{aligned}
\textbf{min} \quad & \sum_{k} w_k \\
\text{s.t.} \quad & L_k^P \;\leq \sum_{\{k,m\}\in\delta(k)} H_{k,m}^P(e_k, f_k, e_m, f_m) \;\leq\; U_k^P \quad \forall k \\
& L_k^Q \;\leq \sum_{\{k,m\}\in\delta(k)} H_{k,m}^Q(e_k, f_k, e_m, f_m) \;\leq\; U_k^Q \quad \forall k \\
& V_k^L \;\leq\; \|(e_k, f_k)\| \;\leq\; V_k^U \quad \forall k \\
& v_k \;=\; \sum_{\{k,m\}\in\delta(k)} H_{k,m}^P(e_k, f_k, e_m, f_m) \quad \forall k \\
& w_k \;=\; F_k(v_k)
\end{aligned}
$$

Complexity

Theorem (2011) Lavaei and Low: OPF is (weakly) NP-hard on trees.

Theorem (2014) van Hentenryck et al: OPF is (weakly) NP-hard on trees.

Theorem (2007) B. and Verma (2009): OPF is strongly NP-hard on general graphs.

Recent insight: use the SDP relaxation (Lavaei and Low, 2009 + many others)

SDP Relaxation of OPF:

Fact: The SDP relaxation sometimes has a rank-1 solution!!
Fact: And when not, sometimes it gives a good bound.

But: the SDP relaxation is always slow on large graphs

- Real-life grids $\longrightarrow > 10^4$ vertices
- SDP relaxation of OPF does not terminate

But...

But: the SDP relaxation is always slow on large graphs

- Real-life grids $\rightarrow > 10^4$ vertices

- SDP relaxation of OPF does not terminate

But...

Fact? Real-life grids have **small tree-width**

Definition 1: A graph has treewidth $\leq \boldsymbol{w}$ if it has a chordal supergraph with clique number $\leq \boldsymbol{w} + \mathbf{1}$

But: the SDP relaxation is always slow on large graphs

- Real-life grids $\rightarrow > 10^4$ vertices

- SDP relaxation of OPF does not terminate

But...

Fact? Real-life grids have **small tree-width**

Definition 2: A graph has treewidth $\leq \boldsymbol{w}$ if it is a subgraph of an intersection graph of subtrees of a tree, with $\leq \boldsymbol{w+1}$ subtrees overlapping at any vertex

But: the SDP relaxation is always slow on large graphs

- Real-life grids $\longrightarrow > 10^4$ vertices

- SDP relaxation of OPF does not terminate

But...

Fact? Real-life grids have **small tree-width**

Definition 2: A graph has treewidth $\leq \boldsymbol{w}$ if it is a subgraph of an intersection graph of subtrees of a tree, with $\leq \boldsymbol{w} + \mathbf{1}$ subtrees overlapping at any vertex

(Seymour and Robertson, early 1980s)

But: the SDP relaxation is always slow on large graphs

- Real-life grids $\rightarrow > 10^4$ vertices

- SDP relaxation of OPF does not terminate

But...

Fact? Real-life grids have **small tree-width**

Matrix-completion Theorem

gives fast SDP implementations:

Real-life grids with $\approx 3 \times 10^3$ vertices: $\rightarrow$ 20 minutes runtime

Much previous work using treewidth

- Bienstock and Özbay (Sherali-Adams + treewidth)
- Wainwright and Jordan (Sherali-Adams + treewidth)
- Grimm, Netzer, Schweighofer
- Laurent (Sherali-Adams + treewidth)
- Lasserre et al (moment relaxation + treewidth)
- Waki, Kim, Kojima, Muramatsu

older work ...

- Lauritzen (1996): tree-junction theorem
- Bertele and Brioschi (1972) (Nemhauser 1960s): nonserial dynamic programming
- Bounded tree-width in combinatorial optimization (early 1980s) (Arnborg et al plus too many authors)

But: the SDP relaxation is always slow on large graphs

- Real-life grids $\longrightarrow > 10^4$ vertices
- SDP relaxation of OPF does not terminate

But...

Fact? Real-life grids have **small tree-width**

Matrix-completion Theorem

gives fast SDP implementations:

Real-life grids with $\approx 3 \times 10^3$ vertices: $\rightarrow$ 20 minutes runtime

$\longrightarrow$ Perhaps low tree-width yields **direct** algorithms for ACOPF itself?

That is to say, not for a relaxation?

A classical problem: fixed-charge network flows

Setting: a directed graph G, and

- $\forall$ arc (i, j) a *capacity* u_{ij}, a *fixed cost* k_{ij} and a *variable cost* c_{ij}.
- At each vertex i, a *net supply* b_i. We assume $\sum_i b_i = 0$.
- By paying k_{ij} the capacity of (i, j) becomes u_{ij} – else zero.
- The per-unit flow cost on (i, j) is c_{ij}.

Problem: At minimum cost, send flow b_i out of each node i.

Knapsack problem (subset sum) is a special case where G is a caterpillar.

Mixed-integer Network Polynomial Optimization problems

Input: an undirected graph G.

- Each variable is associated with some vertex.
  X_u = variables associated with u

Mixed-integer Network Polynomial Optimization problems

Input: an undirected graph G.

- Each variable is associated with some vertex.
  X_u = variables associated with u
- Each constraint is associated with some vertex.
  A constraint associated with $u \in V(G)$ is of the form

$$\sum_{\{u,v\} \in \delta(u)} p_{uv}(X_u \cup X_v) \ \geq \ 0$$

where $p_{uv}()$ is a polynomial

Mixed-integer Network Polynomial Optimization problems

Input: an undirected graph G.

- Each variable is associated with some vertex.
  X_u = variables associated with u
- Each constraint is associated with some vertex.
  A constraint associated with $u \in V(G)$ is of the form

$$\sum_{\{u,v\} \in \delta(u)} p_{uv}(X_u \cup X_v) \ \geq \ 0$$

  where $p_{uv}()$ is a polynomial
- For any x_j, $\{u \in V(G) \, : \, x_j \in X_u\}$ induces a *connected* subgraph of G
- All variables in $[0, 1]$, or binary
- Linear objective

Mixed-integer Network Polynomial Optimization problems

Input: an undirected graph G.

- Each variable is associated with some vertex.
  X_u = variables associated with u

- Each constraint is associated with some vertex.
  A constraint associated with $u \in V(G)$ is of the form

$$\sum_{\{u,v\} \in \delta(u)} p_{uv}(X_u \cup X_v) \ \geq \ 0$$

  where $p_{uv}()$ is a polynomial

- For any x_j, $\{u \in V(G) \, : \, x_j \in X_u\}$ induces a *connected* subgraph of G

- All variables in $[0, 1]$, or binary

- Linear objective

Density: max number of variables + constraints at any vertex

ACOPF: density = 4, FCNF: density = 4

Theorem

Given a problem on a graph with

- **treewidth** $\boldsymbol{w}$,
- **density** $\boldsymbol{d}$,
- **max. degree** of a polynomial p_{uv}: $\boldsymbol{\pi}$,
- $\boldsymbol{n}$ vertices,

and any fixed $\mathbf{0 < \epsilon < 1}$,

there is a **linear program** of size (rows + columns) $\boldsymbol{O(\pi^{wd}\epsilon^{-w}\, n)}$ whose feasibility and optimality error is $\boldsymbol{O(\epsilon)}$

Theorem

Given a problem on a graph with

- **treewidth** $\boldsymbol{w}$,
- **density** $\boldsymbol{d}$,
- **max. degree** of a polynomial p_{uv}: $\boldsymbol{\pi}$,
- $\boldsymbol{n}$ vertices,

and any fixed $\mathbf{0 < \epsilon < 1}$,

there is a **linear program** of size (rows + columns) $\boldsymbol{O(\pi^{wd}\epsilon^{-w}\, n)}$
whose feasibility and optimality error is $\boldsymbol{O(\epsilon)}$

- Problem feasible $\longrightarrow$ LP ϵ-feasible
  additive error $= \epsilon$ times L_1 norm of constraint
  and objective value changes by ϵ times L_1 norm of objective
- And viceversa
- Unless P = NP, need $\boldsymbol{\Omega(\epsilon)}$ error and $\boldsymbol{\Omega(\epsilon^{-1})}$ complexity

More general: (Basic polynomially-constrained mixed-integer LP)

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & p_i(x) \ \geq \ 0 & 1 \leq i \leq m \\ & x_j \in \{0,1\} \quad \forall j \in I, & 0 \leq x_j \leq 1, \quad \text{otherwise} \end{array}$$

Each $\boldsymbol{p_i(x)}$ is a polynomial.

Theorem

For any instance where

- the **intersection graph** has treewidth $\boldsymbol{w}$,
- **max. degree** of any $p_i(x)$ is $\boldsymbol{\pi}$,
- $\boldsymbol{n}$ variables,

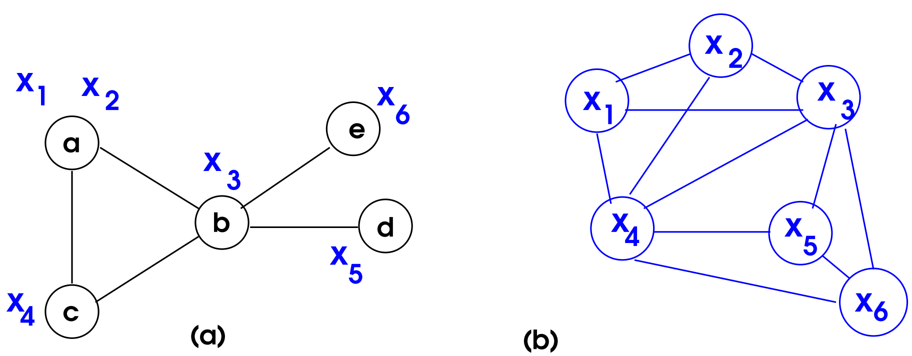
and any fixed $\boldsymbol{0 < \epsilon < 1}$, there is a **linear program** of size (rows + columns) $\boldsymbol{O(\pi^w \epsilon^{-w-1} n)}$ whose feasibility and optimality error is $\boldsymbol{O(\epsilon)}$ (abridged).

Intersection graph of a constraint system: (Fulkerson? (1962?))

- Has a **vertex** for every variably $\boldsymbol{x_j}$
- Has an **edge** $\boldsymbol{\{x_i, x_j\}}$ whenever x_i and x_j appear in the same constraint

Example. Consider the NPO

$$
\begin{aligned}
x_1^2 + x_2^2 + 2x_3^2 &\leq 1 \\
x_1^2 - x_3^2 + x_4 &\geq 0, \\
x_3 x_4 + x_5^3 - x_6 &\geq 1/2 \\
0 \leq x_j \leq 1, \quad 1 \leq j \leq 5, &\quad x_6 \in \{0, 1\}.
\end{aligned}
$$



Main technique: approximation through pure-binary problems

Glover, 1975 (abridged)

Let $\boldsymbol{x}$ be a variable, with bounds $\mathbf{0} \leq \boldsymbol{x} \leq \mathbf{1}$. Let $\mathbf{0} < \boldsymbol{\gamma} < \mathbf{1}$. Then we can approximate

$$\boldsymbol{x} \approx \sum_{h=1}^{L} 2^{-h} y_h$$

where each $\boldsymbol{y_h}$ is a **binary variable**. In fact, choosing $\boldsymbol{L} = \lceil \log_2 \gamma^{-1} \rceil$, we have

$$\boldsymbol{x} \leq \sum_{h=1}^{L} 2^{-h} y_h \leq \boldsymbol{x} + \boldsymbol{\gamma}.$$

$\rightarrow$ Given a mixed-integer polynomially constrained LP apply this technique to each continuous variable x_j

Mixed-integer polynomially-constrained LP:

$$
\textbf{(P)} \quad \min \quad c^T x
$$

$$
\text{s.t.} \quad p_i(x) \ \geq \ 0 \quad 1 \leq i \leq m
$$

$$
x_j \in \{0,1\} \quad \forall j \in I, \quad 0 \leq x_j \leq 1, \quad \text{otherwise}
$$

substitute: $\forall j \notin I,\ \ \boldsymbol{x_j} \ \rightarrow \ \sum_{h=1}^{L} \mathbf{2}^{-h}\, \boldsymbol{y_{h,j}}$, where each $\boldsymbol{y_{h,j}} \in \{\mathbf{0}, \mathbf{1}\}$

$\boldsymbol{L} \approx \mathbf{log_2}\, \boldsymbol{\gamma}^{-1}$

Mixed-integer polynomially-constrained LP:

$$\textbf{(P)} \quad \min \quad c^T x$$
$$\text{s.t.} \quad p_i(x) \;\geq\; 0 \quad 1 \leq i \leq m$$
$$x_j \in \{0,1\} \quad \forall j \in I, \quad 0 \leq x_j \leq 1, \quad \text{otherwise}$$

substitute: $\forall j \notin I, \quad \boldsymbol{x_j} \;\rightarrow\; \sum_{h=1}^{L} 2^{-h}\, \boldsymbol{y_{h,j}}$, where each $\boldsymbol{y_{h,j}} \in \{0,1\}$

$$\boldsymbol{L \approx \log_2 \gamma^{-1}}$$

$$\boldsymbol{p(\hat{x}) \;\geq\; 0,\; |\hat{x}_j - \sum_{h=1}^{L} 2^{-h}\, \hat{y}_{h,j}| \;\leq\; \gamma \;\Rightarrow\; p(\hat{y}) \;\geq\; -\|p\|_1(1-(1-\gamma)^{\pi})}$$

- $\boldsymbol{\pi}$ = degree of $p(x)$
- $\|\boldsymbol{p}\|_1$ = 1-norm of coefficients of $p(x)$
- $\boldsymbol{-\|p\|_1(1-(1-\gamma)^{\pi}) \;\approx\; -\|p\|_1 \pi\, \gamma}$

Mixed-integer polynomially-constrained LP:

$$\textbf{(P)} \quad \min \quad c^T x$$
$$\text{s.t.} \quad p_i(x) \;\geq\; 0 \quad 1 \leq i \leq m$$
$$x_j \in \{0,1\} \quad \forall j \in I, \quad 0 \leq x_j \leq 1, \quad \text{otherwise}$$

substitute: $\forall j \notin I$, $\boldsymbol{x_j} \;\rightarrow\; \sum_{h=1}^{L} \mathbf{2}^{-h}\, \boldsymbol{y_{h,j}}$, where each $\boldsymbol{y_{h,j}} \in \{\mathbf{0}, \mathbf{1}\}$

$$\boldsymbol{L} \approx \mathbf{log_2}\, \boldsymbol{\gamma^{-1}}$$

Approximation: pure-binary polynomially-constrained LP:

$$\textbf{(Q)} \quad \min \quad \bar{c}^T y$$
$$\text{s.t.} \quad \bar{p}_i(z) \;\geq\; -\|p_i\|_1(1-(1-\gamma)^{\pi}) \quad 1 \leq i \leq m$$
$$z \doteq \text{vector consisting of } x_j \text{ for } j \in I \text{ and all added } y \text{ variables}$$
$$z_j \in \{0,1\} \quad \forall j$$

Mixed-integer polynomially-constrained LP:

$$\textbf{(P)} \quad \min \quad c^T x$$
$$\text{s.t.} \quad p_i(x) \ \geq \ 0 \quad 1 \leq i \leq m$$
$$x_j \in \{0,1\} \quad \forall j \in I, \quad 0 \leq x_j \leq 1, \quad \text{otherwise}$$

substitute: $\forall j \notin I,\ \ \boldsymbol{x_j} \ \rightarrow \ \sum_{h=1}^{L} 2^{-h} \boldsymbol{y_{h,j}}$, where each $\boldsymbol{y_{h,j}} \in \{0,1\}$

$$\boldsymbol{L \approx \log_2 \pi \epsilon^{-1}}$$

Approximation: pure-binary polynomially-constrained LP:

$$\textbf{(Q)} \quad \min \quad \bar{c}^T y$$
$$\text{s.t.} \quad \bar{p}_i(y) \ \geq \ -\|p_i\|_1 (1 - (1-\gamma)^{\pi}) \quad 1 \leq i \leq m$$
$$z \doteq \text{vector consisting of } x_j \text{ for } j \in I \text{ and all added } y \text{ variables}$$
$$z_j \in \{0,1\} \quad \forall j$$

Intersection graph of **P** has treewidth $\leq \boldsymbol{\omega} \ \Rightarrow$
Intersection graph of **Q** has treewidth $\leq \boldsymbol{L\omega}$

Pure binary problems

- $\boldsymbol{n}$ binary variables and $\boldsymbol{m}$ constraints.
- Constraint i is given by $\boldsymbol{k[i] \subseteq \{1, \ldots, n\}}$ and $\boldsymbol{S^i \subseteq \{0,1\}^{k[i]}}$.
  1. Constraint states: subvector $\boldsymbol{x_{k[i]} \in S^i}$.
  2. $\boldsymbol{S^i}$ given by a *membership oracle*
- The problem is to minimize a linear function $c^T x$, over $x \in \{0,1\}^n$, and subject to all constraints $\boldsymbol{i}$, $\boldsymbol{1 \leq i \leq m}$.

Pure binary problems

- $\boldsymbol{n}$ binary variables and $\boldsymbol{m}$ constraints.
- Constraint i is given by $\boldsymbol{k[i] \subseteq \{1, \ldots, n\}}$ and $\boldsymbol{S^i \subseteq \{0,1\}^{k[i]}}$.
  1. Constraint states: subvector $\boldsymbol{x_{k[i]} \in S^i}$.
  2. $\boldsymbol{S^i}$ given by a *membership oracle*
- The problem is to minimize a linear function $c^T x$, over $x \in \{0,1\}^n$, and subject to all constraints $\boldsymbol{i}$, $\boldsymbol{1 \leq i \leq m}$.

Theorem. If intersection graph has treewidth $\leq \boldsymbol{W}$, then: there is an LP formulation with $\boldsymbol{O(2^W n)}$ variables and constraints.

Pure binary problems

- $\boldsymbol{n}$ binary variables and $\boldsymbol{m}$ constraints.
- Constraint i is given by $\boldsymbol{k[i] \subseteq \{1,\ldots,n\}}$ and $\boldsymbol{S^i \subseteq \{0,1\}^{k[i]}}$.
  1. Constraint states: subvector $\boldsymbol{x_{k[i]} \in S^i}$.
  2. $\boldsymbol{S^i}$ given by a *membership oracle*
- The problem is to minimize a linear function $c^T x$, over $x \in \{0,1\}^n$, and subject to all constraint $\boldsymbol{i, \quad 1 \leq i \leq m}$.

Theorem. If intersection graph has treewidth $\leq \boldsymbol{W}$, then: there is an LP formulation with $\boldsymbol{O(2^W n)}$ variables and constraints.

- Not explicitly stated, but can be obtained using methods from Laurent (2010)
- "Cones of zeta functions" approach of Lovasz and Schrijver.
- Poly-time algorithm: **old result**.

Pure binary problems

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x_{k[i]} \in S^i \quad 1 \leq i \leq m, \\ & x \in \{0,1\}^n \end{aligned}$$

Theorem. If intersection graph has treewidth $\leq \boldsymbol{W}$, then: there is an LP formulation with $\boldsymbol{O(2^W n)}$ variables and constraints.

An alternative approach?

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & x_{k[i]} \in S^i \quad 1 \leq i \leq m, \\
& x \in \{0,1\}^n
\end{aligned}
$$

An alternative approach?

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x_{k[i]} \in S^i \quad 1 \le i \le m, \\ & x \in \{0,1\}^n \end{aligned}$$

$$\text{conv}\{y \in \{0,1\}^{k[i]} \,:\, y \in S^i\} \quad \text{given by} \quad \boldsymbol{A^i x \ge b^i}$$

An alternative approach?

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x_{k[i]} \in S^i \quad 1 \le i \le m, \\ & x \in \{0,1\}^n \end{aligned}$$

$$\text{conv}\{y \in \{0,1\}^{k[i]} \, : \, y \in S^i\} \quad \text{given by} \quad \boldsymbol{A^i x \geq b^i}$$

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & A^i x_{k[i]} \ge b^i \quad 1 \le i \le m, \\ & x \in \{0,1\}^n \end{aligned}$$

An alternative approach?

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x_{k[i]} \in S^i \quad 1 \leq i \leq m, \\ & x \in \{0,1\}^n \end{aligned}$$

$$\text{conv}\{y \in \{0,1\}^{k[i]} : y \in S^i\} \quad \text{given by} \quad \boldsymbol{A^i x \geq b^i}$$

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & A^i x_{k[i]} \geq b^i \quad 1 \leq i \leq m, \\ & x \in \{0,1\}^n \end{aligned}$$

But: Bárany, Pór (2001):

for d large enough, there exist 0,1-polyhedra in $\mathbb{R}^d$ with

$$\left(\frac{d}{\log d}\right)^{d/4} \quad \text{facets}$$

Corollary: (polynomially-constrained mixed-integer LP)

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & p_i(x) \ \geq \ 0 \quad 1 \leq i \leq m \\ & x_j \in \{0,1\} \quad \forall j \in I, \quad 0 \leq x_j \leq 1, \quad \text{otherwise} \end{aligned}$$

Each $\boldsymbol{p_i(x)}$ is a polynomial.

Theorem

For any instance where

- the **intersection graph** has treewidth $\boldsymbol{w}$,
- **max. degree** of any $p_i(x)$ is $\boldsymbol{\pi}$,
- $\boldsymbol{n}$ variables,

and any fixed $\mathbf{0 < \epsilon < 1}$, there is a **linear program** of size (rows + columns) $\boldsymbol{O(\pi^{w}\epsilon^{-w-1}\, n)}$ whose feasibility and optimality error is $\boldsymbol{O(\epsilon)}$ (abridged).

Application? Mixed-integer Network Polynomial Optimization problems

Input: an undirected graph G.

- Variables and constraints associated with vertices.
- X_u = variables associated with u.
- A constraint associated with $u \in V(G)$ is of the form

$$\sum_{\{u,v\} \in \delta(u)} p_{uv}(X_u \cup X_v) \ \geq \ 0$$

  where $p_{uv}()$ is a polynomial
- All variables in $[0, 1]$, or binary.
- Linear objective
- **Interesting case:** $\boldsymbol{G}$ of bounded treewidth.

Application? Mixed-integer Network Polynomial Optimization problems

Input: an undirected graph G.

- Variables and constraints associated with vertices.
- X_u = variables associated with u.
- A constraint associated with $u \in V(G)$ is of the form

$$\sum_{\{u,v\} \in \delta(u)} p_{uv}(X_u \cup X_v) \ \geq \ 0$$

  where $p_{uv}()$ is a polynomial
- All variables in $[0, 1]$, or binary.
- Linear objective
- **Interesting case:** $\boldsymbol{G}$ of bounded treewidth.

Trouble! Treewidth of $\boldsymbol{G}$ $\neq$ treewidth of intersection graph of constraints

Application? Mixed-integer Network Polynomial Optimization problems

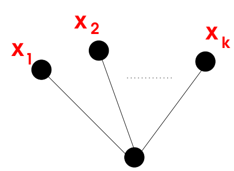
Input: an undirected graph G.

- Variables and constraints associated with vertices.
- X_u = variables associated with u.
- A constraint associated with $u \in V(G)$ is of the form

$$\sum_{\{u,v\} \in \delta(u)} p_{uv}(X_u \cup X_v) \ \geq \ 0$$

  where $p_{uv}()$ is a polynomial

- All variables in $[0,1]$, or binary.
- Linear objective
- **Interesting case:** $\boldsymbol{G}$ of bounded treewidth.



$$\sum_{j=1}^{k} a_j x_j \ \geq \ a_0, \qquad \rightarrow \text{k-clique}$$

Vertex splitting

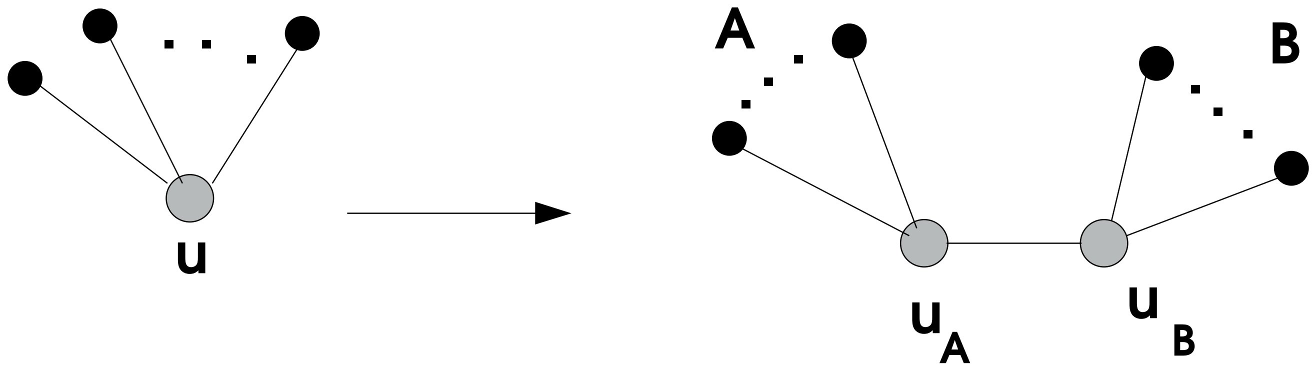
How do we deal with

$\sum_{\{u,v\}\in\delta(u)} p_{uv}(X_u \cup X_v) \geq 0$ when $|\boldsymbol{\delta(u)}|$ large?

Vertex splitting

How do we deal with

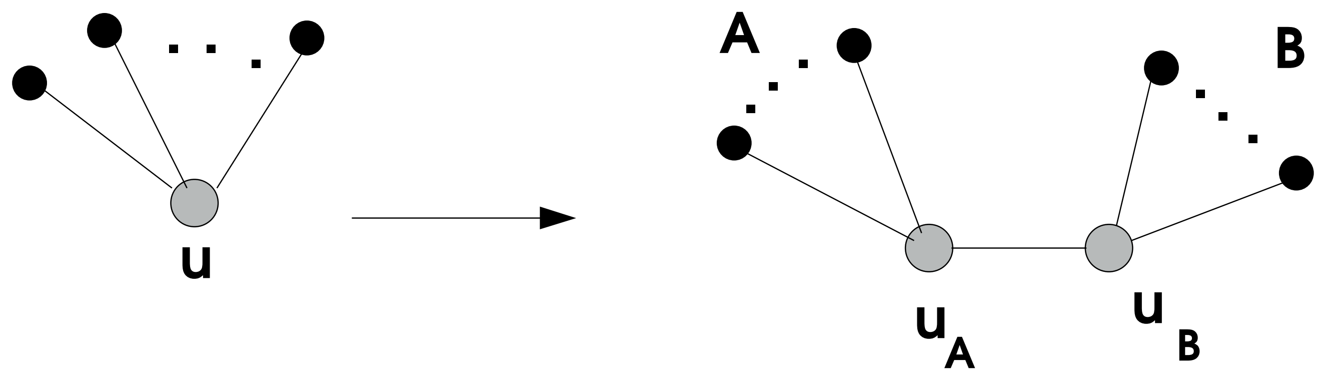
$\sum_{\{u,v\}\in\delta(u)} p_{uv}(X_u \cup X_v) \geq 0$ when $|\boldsymbol{\delta(u)}|$ large?



Vertex splitting

How do we deal with

$\sum_{\{u,v\}\in\delta(u)} p_{uv}(X_u \cup X_v) \geq 0$ when $|\boldsymbol{\delta(u)}|$ large?

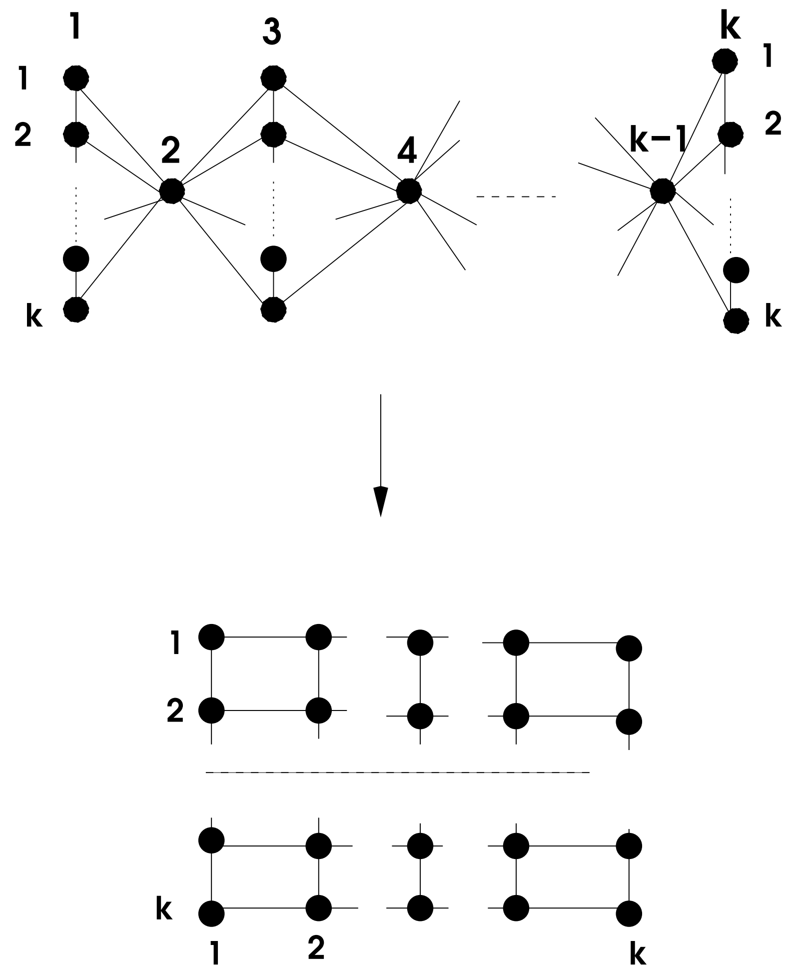


$$\sum_{\{u,v\}\in A} p_{u,v}(X_u \cup X_v) + y \geq 0 \quad \text{assoc. with } u_A$$

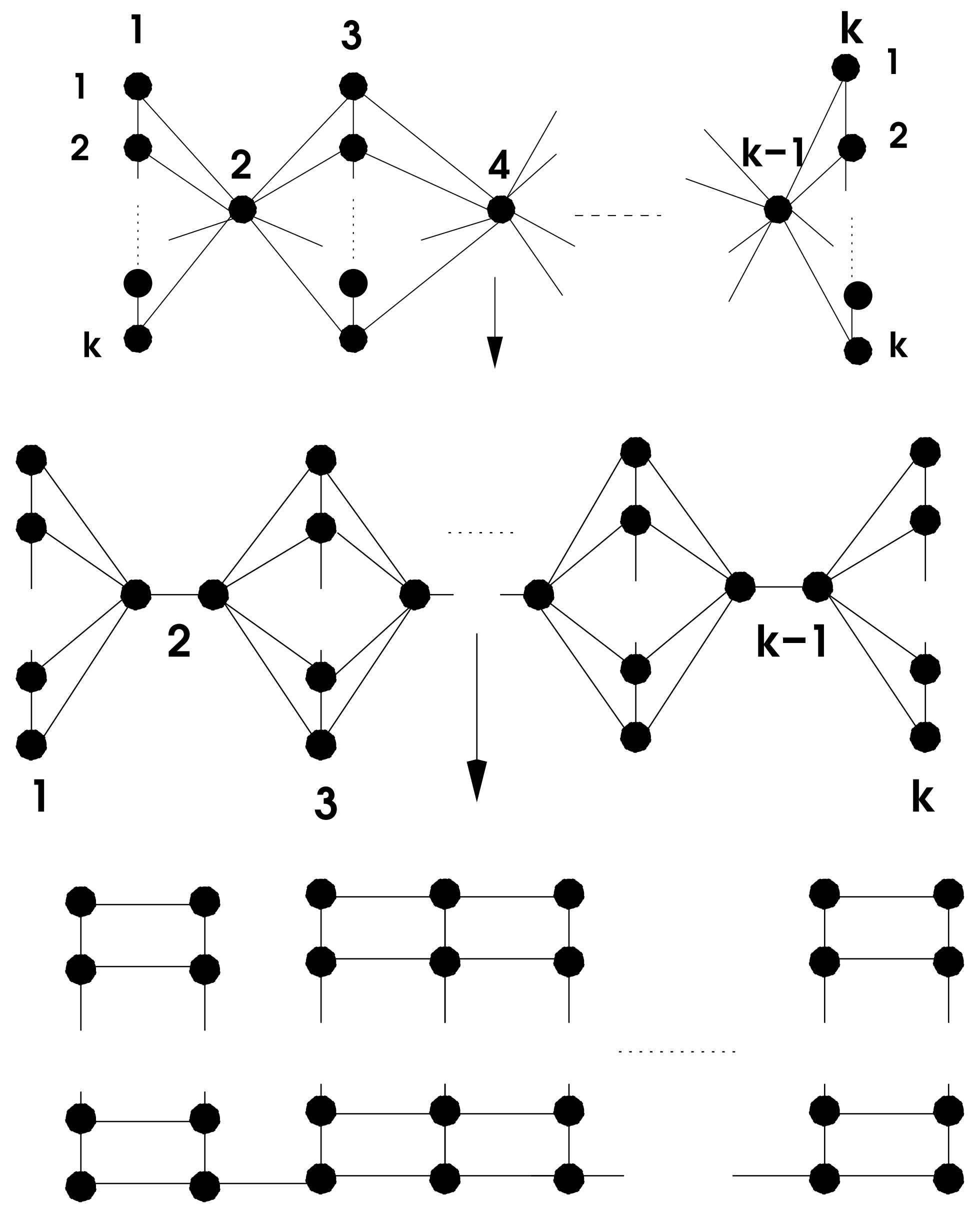
$$\sum_{\{u,v\}\in B} p_{u,v}(X_u \cup X_v) - y = 0. \quad \text{assoc. with } u_B$$

(y is a new variable associated with either u_A or u_B)

Does not work



A better idea



Theorem

Given a graph of treewidth $\leq \omega$, there is a sequence of vertex splittings such that the resulting graph

- Has treewidth $\leq O(\omega)$
- Has maximum degree ≤ 3.

Theorem

Given a graph of treewidth $\leq \omega$, there is a sequence of vertex splittings such that the resulting graph

- Has treewidth $\leq O(\omega)$
- Has maximum degree ≤ 3.

Perhaps known to graph minors people?

Corollary (abridged)

Given a network polynomial optimization problem on a graph G, with treewidth $\leq \omega$ there is an **equivalent** problem on a graph H with treewidth $\leq O(\omega)$ and max degree 3.

Corollary. The intersection graph has treewidth $\leq O(\omega)$.