LP formulations for sparse polynomial optimization problems

Daniel Bienstock and Gonzalo Muñoz, Columbia University

An application: the Optimal Power Flow problem (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:

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

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

$$\min \quad \sum_k F_k \left(\sum_{\{k,m\} \in \delta(k)} H^P_{k,m}(e_k, f_k, e_m, f_m) \right)$$

$$\text{s.t.} \quad L^P_k \le \sum_{\{k,m\} \in \delta(k)} H^P_{k,m}(e_k, f_k, e_m, f_m) \le U^P_k \quad \forall k$$

$$L^Q_k \le \sum_{\{k,m\} \in \delta(k)} H^Q_{k,m}(e_k, f_k, e_m, f_m) \le U^Q_k \quad \forall k$$

$$V^L_k \le \|(e_k, f_k)\| \le V^U_k \quad \forall k.$$

Function F_k in the objective: convex quadratic

Complexity

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

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

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

Complexity

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

Theorem (2014) van Hentenryck et al: OPF is (strongly) 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)

$$
\begin{aligned}
\min \quad & \sum_k F_k \left(\sum_{\{k,m\} \in \delta(k)} H^P_{k,m}(e_k, f_k, e_m, f_m) \right) \\
\text{s.t.} \quad & L^P_k \;\leq\; \sum_{\{k,m\} \in \delta(k)} H^P_{k,m}(e_k, f_k, e_m, f_m) \;\leq\; U^P_k \quad \forall k \\
& L^Q_k \;\leq\; \sum_{\{k,m\} \in \delta(k)} H^Q_{k,m}(e_k, f_k, e_m, f_m) \;\leq\; U^Q_k \quad \forall k \\
& V^L_k \;\leq\; \|(e_k, f_k)\| \;\leq\; V^U_k \quad \forall k.
\end{aligned}
$$

Complexity

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

Theorem (2014) van Hentenryck et al: OPF is (strongly) 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)

Reformulation of ACOPF:

$$\begin{aligned} \min \quad & F \bullet W \\ \text{s.t.} \quad & A_i \bullet W \ \le \ b_i \quad i = 1, 2, \ldots \\ & W \succeq 0, \quad W \text{ of rank } 1. \end{aligned}$$

Complexity

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

Theorem (2014) van Hentenryck et al: OPF is (strongly) 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:

$$\begin{aligned} \min \quad & F \bullet W \\ \text{s.t.} \quad & A_i \bullet W \ \leq \ b_i \quad i = 1, 2, \ldots \\ & W \succeq 0. \end{aligned}$$

Complexity

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

Theorem (2014) van Hentenryck et al: OPF is (strongly) 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:

$$\begin{aligned} \min \quad & F \bullet W \\ \text{s.t.} \quad & A_i \bullet W \ \leq \ b_i \quad i = 1, 2, \ldots \\ & W \succeq 0. \end{aligned}$$

Fact: The SDP relaxation almost always has a rank-1 solution!!

Complexity

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

Theorem (2014) van Hentenryck et al: OPF is (strongly) 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:

$$\begin{aligned} \min \quad & F \bullet W \\ \text{s.t.} \quad & A_i \bullet W \ \leq \ b_i \quad i = 1, 2, \ldots \\ & W \succeq 0. \end{aligned}$$

Fact: The SDP relaxation sometimes has a rank-1 solution!!

Complexity

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

Theorem (2014) van Hentenryck et al: OPF is (strongly) 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:

$$\begin{aligned} \min \quad & F \bullet W \\ \text{s.t.} \quad & A_i \bullet W \ \leq \ b_i \quad i = 1, 2, \ldots \\ & W \succeq 0. \end{aligned}$$

Fact: The SDP relaxation sometimes has a rank-1 solution!!
Fact: But it is always very tight!!

Complexity

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

Theorem (2014) van Hentenryck et al: OPF is (strongly) 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:

$$\begin{aligned} \min \quad & F \bullet W \\ \text{s.t.} \quad & A_i \bullet W \ \leq \ b_i \quad i = 1, 2, \ldots \\ & W \succeq 0. \end{aligned}$$

Fact: The SDP relaxation sometimes has a rank-1 solution!!
Fact: But it is frequently rather tight!!

Complexity

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

Theorem (2014) van Hentenryck et al: OPF is (strongly) 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:

$$\begin{aligned} \min \quad & F \bullet W \\ \text{s.t.} \quad & A_i \bullet W \;\leq\; b_i \quad i = 1, 2, \ldots \\ & W \succeq 0. \end{aligned}$$

Fact: The SDP relaxation sometimes has a rank-1 solution!!
Fact: But it is usually good!!

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 $\longrightarrow > 10^4$ 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 $\leq \boldsymbol{w+1}$

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

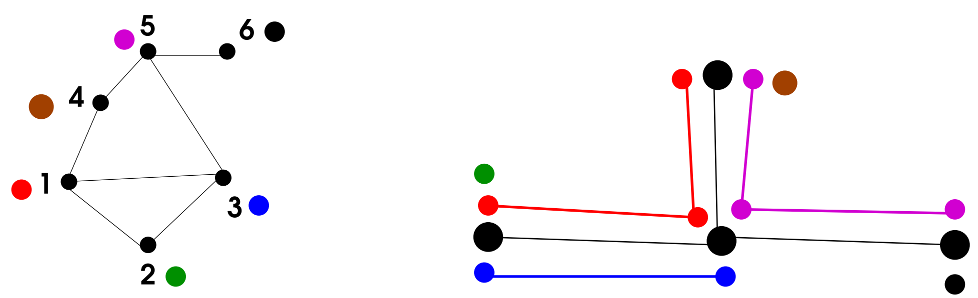
(Seymour and Robertson, early 1980s)

Tree-width

Let G be an undirected graph with vertices $V(G)$ and edges $E(G)$.

A tree-decomposition of G is a pair (T, Q) where:

- T is a tree. **Not** a subtree of G, just a tree
- For each vertex t of T, Q_t is a subset of $V(G)$. These subsets satisfy the two properties:
  - (1) For each vertex v of G, the set $\{t \in V(T) : v \in Q_t\}$ is a **subtree** of T, denoted T_v.
  - (2) For each edge $\{u, v\}$ of G, the two subtrees T_u and T_v **intersect**.
- The **width** of (T, Q) is $\max_{t \in T} |Q_t| - 1$.



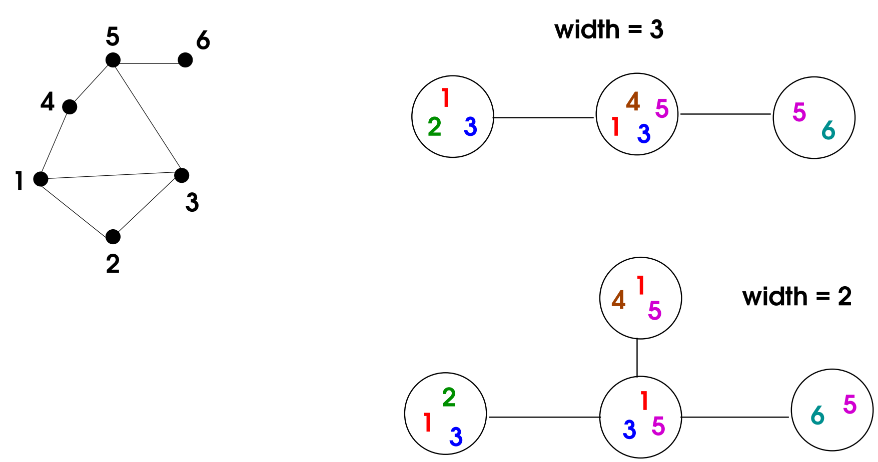
$\rightarrow$ two subtrees T_u, T_v may overlap even if $\{u, v\}$ is **not** an edge of G

Tree-width

Let G be an undirected graph with vertices $V(G)$ and edges $E(G)$.

A tree-decomposition of G is a pair (T, Q) where:

- T is a tree. **Not** a subtree of G, just a tree
- For each vertex t of T, Q_t is a subset of $V(G)$. These subsets satisfy the two properties:
  (1) For each vertex v of G, the set $\{t \in V(T) \, : \, v \in Q_t\}$ is a **subtree** of T, denoted T_v.
  (2) For each edge $\{u, v\}$ of G, the two subtrees T_u and T_v **intersect**.
- The **width** of (T, Q) is $\max_{t \in T} |Q_t| - 1$.



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**

Matrix-completion Theorem

gives fast SDP implementations:

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

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

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

That is to say, not for a relaxation?

Much previous work using structured sparsity

- 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): nonserial dynamic programming
- Bounded tree-width in combinatorial optimization (early 1980s) (Arnborg et al plus too many authors)
- Fulkerson and Gross (1965): matrices with consecutive ones

ACOPF, again

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:

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

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

$$\textbf{min} \quad \sum_k F_k \left(\sum_{\{k,m\} \in \delta(k)} H^P_{k,m}(e_k, f_k, e_m, f_m) \right)$$

$$\text{s.t.} \quad L^P_k \le \sum_{\{k,m\} \in \delta(k)} H^P_{k,m}(e_k, f_k, e_m, f_m) \le U^P_k \quad \forall k$$

$$L^Q_k \le \sum_{\{k,m\} \in \delta(k)} H^Q_{k,m}(e_k, f_k, e_m, f_m) \le U^Q_k \quad \forall k$$

$$V^L_k \le \|(e_k, f_k)\| \le V^U_k \quad \forall k.$$

Function F_k in the objective: convex quadratic

ACOPF, again

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:

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

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

$$\textbf{min} \quad \sum_k w_k$$

$$\text{s.t.} \quad L^P_k \ \leq \sum_{\{k,m\} \in \delta(k)} H^P_{k,m}(e_k, f_k, e_m, f_m) \ \leq \ U^P_k \quad \forall k$$

$$L^Q_k \ \leq \sum_{\{k,m\} \in \delta(k)} H^Q_{k,m}(e_k, f_k, e_m, f_m) \ \leq \ U^Q_k \quad \forall k$$

$$V^L_k \ \leq \ \|(e_k, f_k)\| \ \leq \ V^U_k \quad \forall k$$

$$v_k \ = \sum_{\{k,m\} \in \delta(k)} H^P_{k,m}(e_k, f_k, e_m, f_m) \quad \forall k$$

$$w_k \ = \ F_k(v_k)$$

A classical problem: fixed-charge network flows

Setting: a directed graph G, and

- At each 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 $\sum_i b_i = 0$ (so $b_i < 0$ means i has demand).

- By paying k_{ij} the capacity of (i, j) becomes u_{ij} – else it is 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 $\mathbf{0} < \boldsymbol{\epsilon} < \mathbf{1}$,

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 $\mathbf{0} < \boldsymbol{\epsilon} < \mathbf{1}$,

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 = ϵ times L_1 norm of constraint
  and objective value changes by ϵ times L_1 norm of objective
- And viceversa

Simple example: subset-sum problem

Input: positive integers $p_1, p_2, \ldots, p_n$.

Problem: find a solution to:

$$\sum_{j=1}^{n} p_j x_j \; = \; \frac{1}{2} \sum_{j=1}^{n} p_j$$

$$x_j(1 - x_j) \; = \; 0, \quad \forall j$$

(weakly) NP-hard

This is a network polynomial problem on a **star** – so treewidth 1.

But

$\mathbf{\{0, 1\}}$ solutions with error $\left(\frac{1}{2} \sum_{j=1}^{n} p_j \right) \epsilon$ in time polynomial in ϵ^{-1}

More general: (Basic 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,

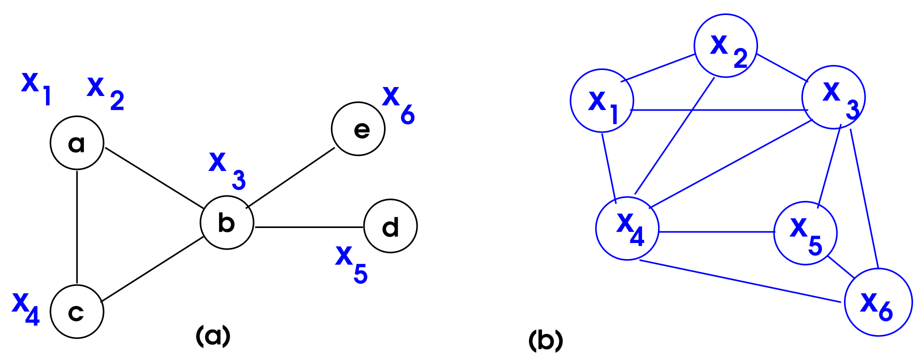
and any fixed $\mathbf{0 < \epsilon < 1}$, there is a **linear program** of size (rows + columns) $\boldsymbol{O(\pi^w \epsilon^{-w-1} n)}$ whose feasibility and optimality error is $\boldsymbol{O(\epsilon)}$ (abridged).

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

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

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

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

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

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

Approximation: pure-binary polynomially-constrained LP:

$$\textbf{(Q)} \quad \min \quad \bar{c}^T y$$
$$\text{s.t.} \quad \bar{p}_i(y) \ \geq \ -\|p_i\|_1 (1 - (1-\gamma)^{\pi}) \quad 1 \leq i \leq m$$
$$x_j \in \{0,1\} \quad \forall j \in I, \quad 0 \leq x_j \leq 1, \quad \text{otherwise}$$

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

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

Approximation: pure-binary polynomially-constrained LP:

$$\textbf{(Q)} \quad \min \quad \bar{c}^T y$$
$$\text{s.t.} \quad \bar{p}_i(y) \ \geq \ -\|p_i\|_1(1 - (1-\gamma)^{\pi}) \quad 1 \leq i \leq m$$
$$x_j \in \{0,1\} \quad \forall j \in I, \quad 0 \leq x_j \leq 1, \quad \text{otherwise}$$

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 constraint $\boldsymbol{i}$, $\boldsymbol{1 \leq i \leq m}$.

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 constraint $\boldsymbol{i}$, $\boldsymbol{1 \leq i \leq m}$.

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 $\boldsymbol{i}$, $\boldsymbol{1 \le i \le m}$.

Theorem. If intersection graph has treewidth $\boldsymbol{\le W}$, 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?

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

An alternative approach?

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

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

An alternative approach?

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

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

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

An alternative approach?

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

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

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

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 $\mathbf{0 < \boldsymbol{\epsilon} < 1}$, 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

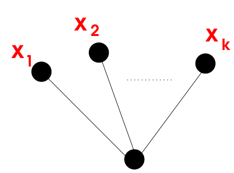
Input: an undirected graph G.

- 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

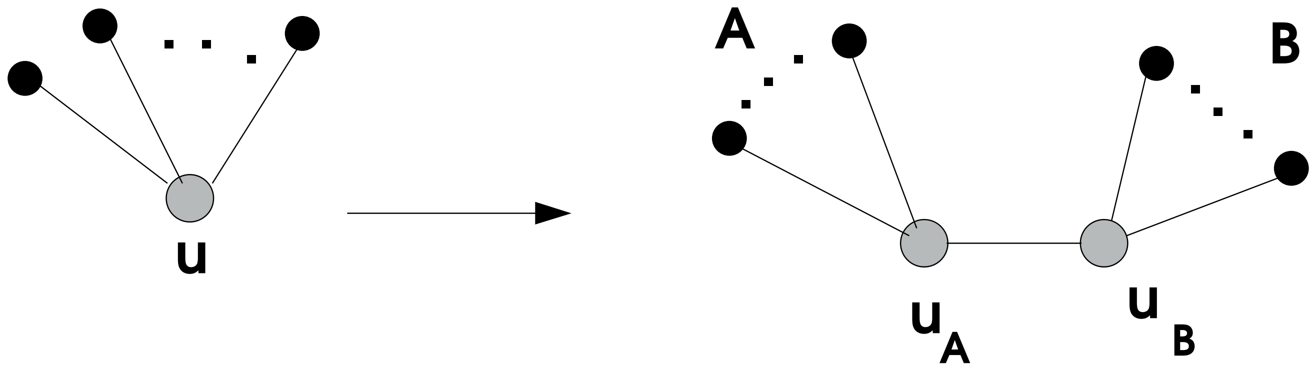
How do we deal with

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

Vertex splitting

How do we deal with

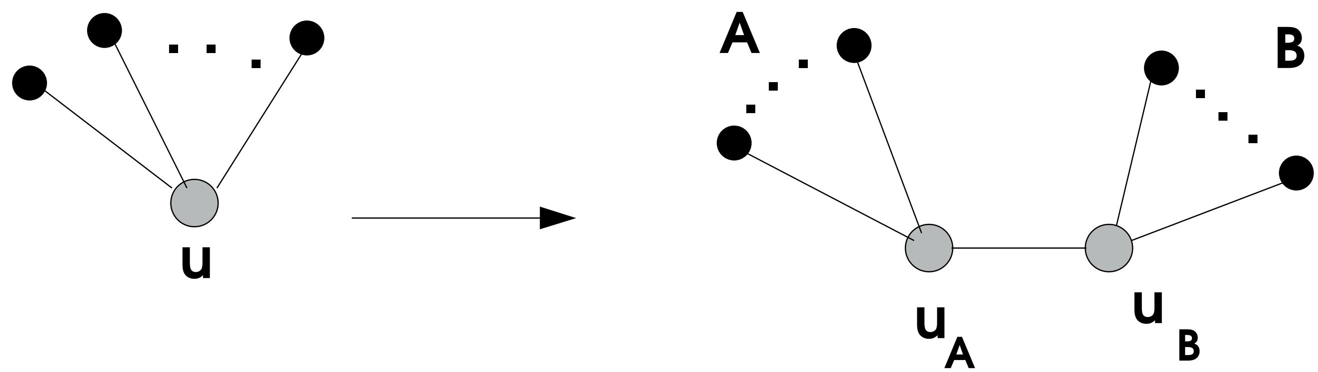
$\sum_{\{u,v\}\in\delta(u)} p_{uv}(X_u \cup X_v) \geq 0$ when $|\boldsymbol{\delta(u)}|$ large?



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?

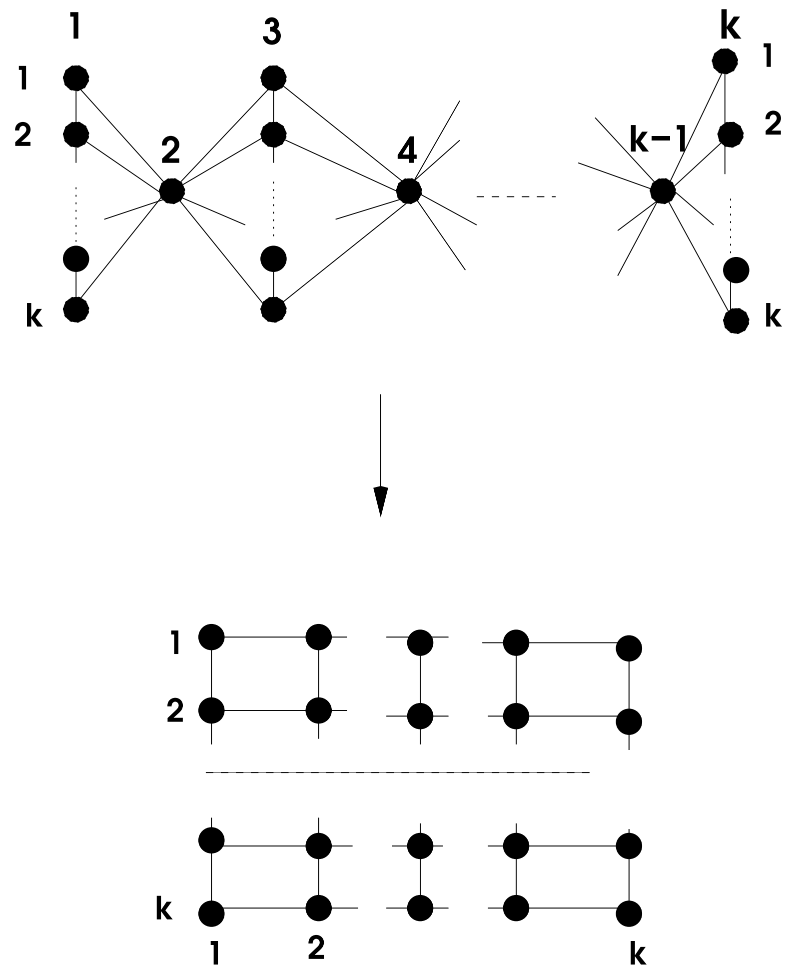


$$\sum_{\{u,v\}\in A} p_{u,v}(X_u \cup X_v) + y \geq 0 \quad \text{assoc. with } u_A$$

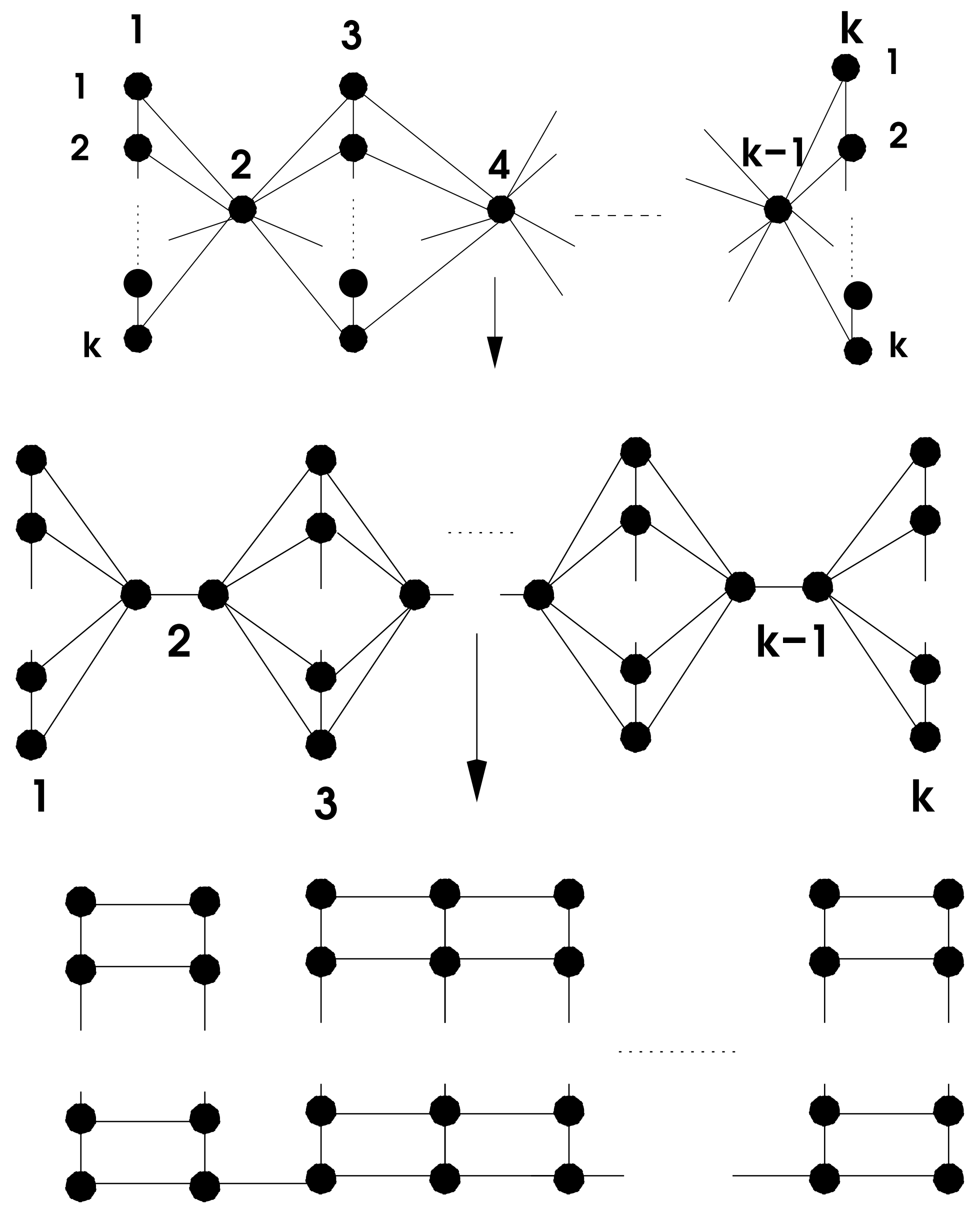
$$\sum_{\{u,v\}\in B} p_{u,v}(X_u \cup X_v) - y = 0. \quad \text{assoc. with } u_B$$

(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.

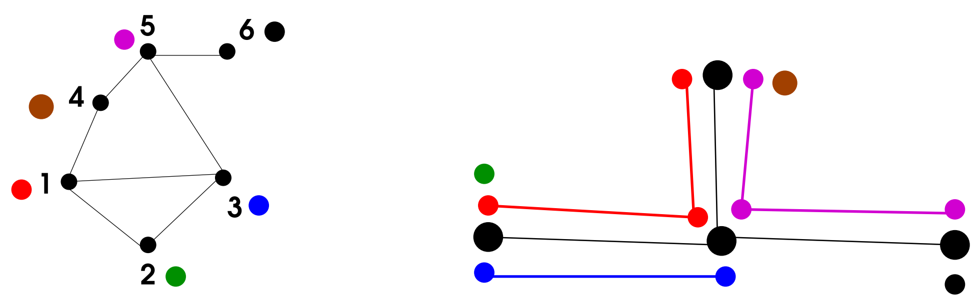
Corollary. The intersection graph has treewidth $\leq O(\omega)$.

Tree-width

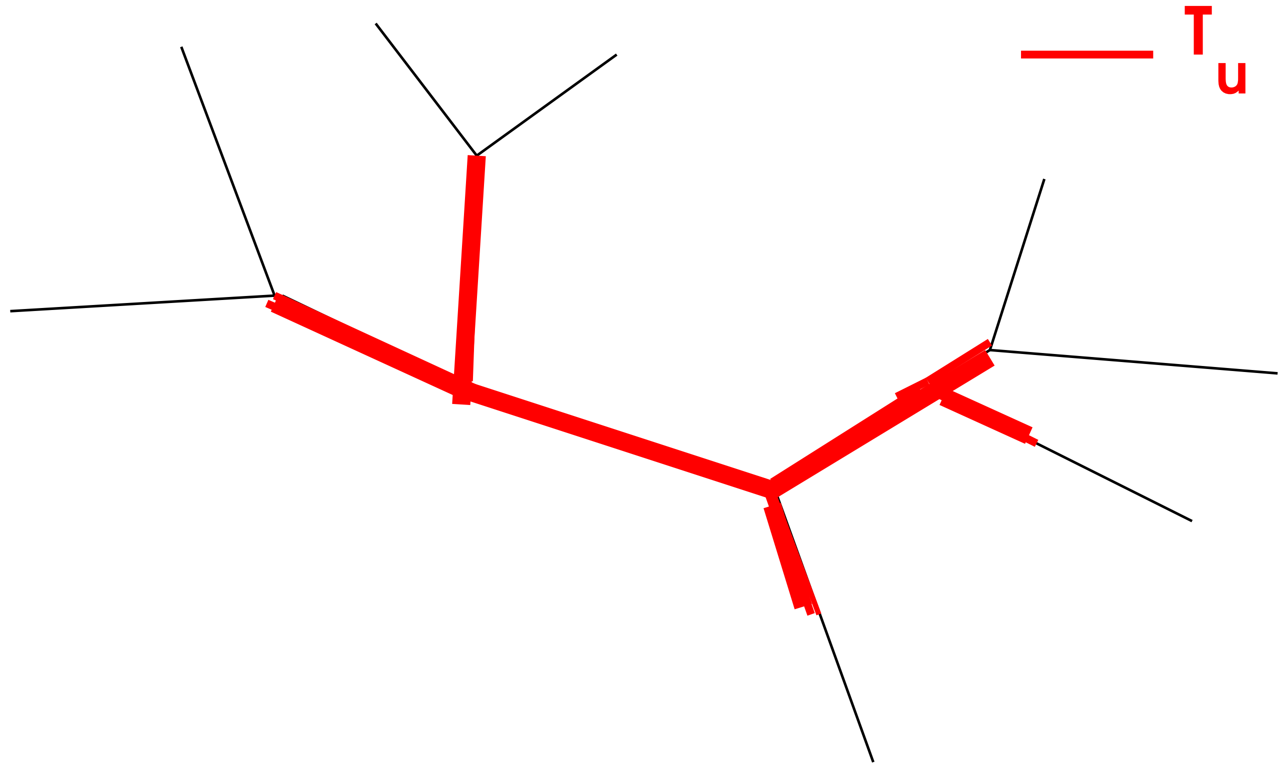
Let G be an undirected graph with vertices $V(G)$ and edges $E(G)$.

A tree-decomposition of G is a pair (T, Q) where:

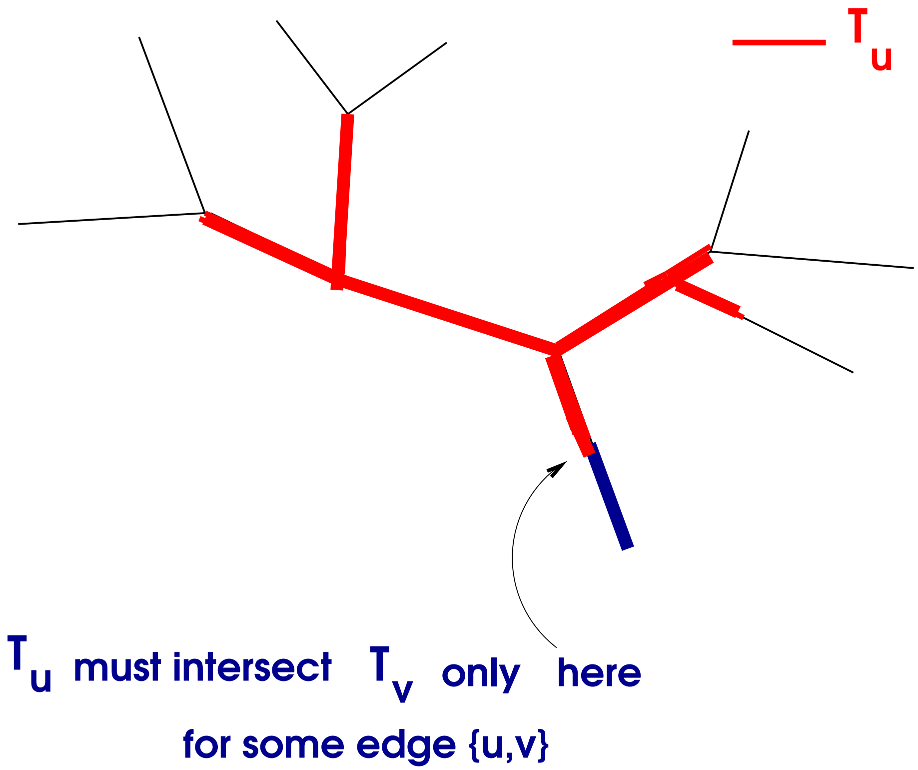
- T is a tree. **Not** a subtree of G, just a tree
- For each vertex t of T, Q_t is a subset of $V(G)$. These subsets satisfy the two properties:
  (1) For each vertex v of G, the set $\{t \in V(T) : v \in Q_t\}$ is a **subtree** of T, denoted T_v.
  (2) For each edge $\{u, v\}$ of G, the two subtrees T_u and T_v **intersect**.
- The **width** of (T, Q) is $\max_{t \in T} |Q_t| - 1$.



→ two subtrees T_u, T_v may overlap even if $\{u, v\}$ is **not** an edge of G



each edge $\boldsymbol{\{u, v\}} \in \boldsymbol{E(G)}$ found in some vertex of $\boldsymbol{T_u}$



wlog **every** edge $\{u, v\} \in E(G)$ found in some **leaf** of T_u

Sat.Nov..7.102812.2015@rockadoodle