New developments on solving AC-OPF on sparse networks

Daniel Bienstock and Gonzalo Muñoz, Columbia University

January 2015

Optimal power flow problem in rectangular coordinates, simplest form

Variables:

- Complex voltages $\boldsymbol{e_k + j f_k}$, power flows $\boldsymbol{P_{km}, Q_{km}}$, auxiliary variables

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

Basic problem

$$\textbf{min} \quad \sum_{k \in V} C_k$$

$$\text{s.t.} \quad \forall km: \quad P_{km} = \boldsymbol{g_{km}}(e_k^2 + f_k^2) - \boldsymbol{g_{km}}(e_k e_m + f_k f_m) + \boldsymbol{b_{km}}(e_k f_m - f_k e_m) \tag{1a}$$

$$\forall km: \quad Q_{km} = -\boldsymbol{b_{km}}(e_k^2 + f_k^2) + \boldsymbol{b_{km}}(e_k e_m + f_k f_m) + \boldsymbol{g_{km}}(e_k f_m - f_k e_m) \tag{1b}$$

$$\forall km: \quad |P_{km}|^2 + |Q_{km}|^2 \leq \boldsymbol{U_{km}} \tag{1c}$$

$$\forall k: \quad \boldsymbol{P_k^{\min}} \leq \sum_{km \in \delta(k)} P_{km} \leq \boldsymbol{P_k^{\max}} \tag{1d}$$

$$\forall k: \quad \boldsymbol{Q_k^{\min}} \leq \sum_{km \in \delta(k)} Q_{km} \leq \boldsymbol{Q_k^{\max}} \tag{1e}$$

$$\forall k: \quad \boldsymbol{V_k^{\min}} \leq e_k^2 + f_k^2 \leq \boldsymbol{V_k^{\max}}, \tag{1f}$$

$$\forall k: \quad C_k = \boldsymbol{F_k}\left(\sum_{km \in \delta(k)} P_{km} \right). \tag{1g}$$

Here, $\boldsymbol{F_k}$ is a quadratic function for each k.

Optimal power flow problem in rectangular coordinates, simplest form

Variables:

- Complex voltages $\boldsymbol{e_k + j f_k}$, power flows $\boldsymbol{P_{km}, Q_{km}}$, auxiliary variables

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

Basic problem

$$\min \quad \sum_{k \in V} C_k$$

$$\text{s.t.} \quad \forall km: \quad P_{km} = \boldsymbol{g_{km}}(e_k^2 + f_k^2) - \boldsymbol{g_{km}}(e_k e_m + f_k f_m) + \boldsymbol{b_{km}}(e_k f_m - f_k e_m) \tag{2a}$$

$$\forall km: \quad Q_{km} = -\boldsymbol{b_{km}}(e_k^2 + f_k^2) + \boldsymbol{b_{km}}(e_k e_m + f_k f_m) + \boldsymbol{g_{km}}(e_k f_m - f_k e_m) \tag{2b}$$

$$\forall km: \quad |P_{km}|^2 + |Q_{km}|^2 \leq \boldsymbol{U_{km}} \tag{2c}$$

$$\forall k: \quad \boldsymbol{P_k^{\min}} \leq \sum_{km \in \delta(k)} P_{km} \leq \boldsymbol{P_k^{\max}} \tag{2d}$$

$$\forall k: \quad \boldsymbol{Q_k^{\min}} \leq \sum_{km \in \delta(k)} Q_{km} \leq \boldsymbol{Q_k^{\max}} \tag{2e}$$

$$\forall k: \quad \boldsymbol{V_k^{\min}} \leq e_k^2 + f_k^2 \leq \boldsymbol{V_k^{\max}}, \tag{2f}$$

$$\forall k: \quad C_k = \boldsymbol{G_k}\left(\sum_{km \in \delta(k)} Q_{km} \right). \tag{2g}$$

Here, $\boldsymbol{G_k}$ is a quadratic function for each k.

Optimal power flow problem in rectangular coordinates, simplest form

Variables:

- Complex voltages $\boldsymbol{e_k + j f_k}$, power flows $\boldsymbol{P_{km}, Q_{km}}$, auxiliary variables

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

Basic problem

$$\textbf{min} \quad \sum_{k \in V} C_k$$

$$\text{s.t.} \quad \forall km: \quad P_{km} = \boldsymbol{g_{km}}(e_k^2 + f_k^2) - \boldsymbol{g_{km}}(e_k e_m + f_k f_m) + \boldsymbol{b_{km}}(e_k f_m - f_k e_m) \tag{3a}$$

$$\forall km: \quad Q_{km} = -\boldsymbol{b_{km}}(e_k^2 + f_k^2) + \boldsymbol{b_{km}}(e_k e_m + f_k f_m) + \boldsymbol{g_{km}}(e_k f_m - f_k e_m) \tag{3b}$$

$$\forall km: \quad |P_{km}|^2 + |Q_{km}|^2 \leq \boldsymbol{U_{km}} \tag{3c}$$

$$\forall k: \quad \boldsymbol{P_k^{\min}} \leq \sum_{km \in \delta(k)} P_{km} \leq \boldsymbol{P_k^{\max}} \tag{3d}$$

$$\forall k: \quad \boldsymbol{Q_k^{\min}} \leq \sum_{km \in \delta(k)} Q_{km} \leq \boldsymbol{Q_k^{\max}} \tag{3e}$$

$$\forall k: \quad \boldsymbol{V_k^{\min}} \leq e_k^2 + f_k^2 \leq \boldsymbol{V_k^{\max}}, \tag{3f}$$

$$\forall k: \quad C_k = \boldsymbol{F_k}\left(\sum_{km \in \delta(k)} P_{km}\right) + \boldsymbol{G_k}\left(\sum_{km \in \delta(k)} Q_{km}\right). \tag{3g}$$

Here, $\boldsymbol{F_k, G_k}$ are quadratic functions for each k.

Optimal power flow problem in rectangular coordinates, simplest form

Variables:

- Complex voltages $\boldsymbol{e_k + j f_k}$, power flows $\boldsymbol{P_{km}, Q_{km}}$, auxiliary variables

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

Basic problem

$$\min \quad \sum_{k \in V} C_k$$

$$\text{s.t.} \quad \forall km: \quad P_{km} = \boldsymbol{g_{km}}(e_k^2 + f_k^2) - \boldsymbol{g_{km}}(e_k e_m + f_k f_m) + \boldsymbol{b_{km}}(e_k f_m - f_k e_m) \tag{4a}$$

$$\forall km: \quad Q_{km} = \boldsymbol{-b_{km}}(e_k^2 + f_k^2) + \boldsymbol{b_{km}}(e_k e_m + f_k f_m) + \boldsymbol{g_{km}}(e_k f_m - f_k e_m) \tag{4b}$$

$$\forall km: \quad |P_{km}|^2 + |Q_{km}|^2 \le \boldsymbol{U_{km}} \tag{4c}$$

$$\forall k: \quad \boldsymbol{P_k^{\min}} \le \sum_{km \in \delta(k)} P_{km} \le \boldsymbol{P_k^{\max}} \tag{4d}$$

$$\forall k: \quad \boldsymbol{Q_k^{\min}} \le \sum_{km \in \delta(k)} Q_{km} \le \boldsymbol{Q_k^{\max}} \tag{4e}$$

$$\forall k: \quad \boldsymbol{V_k^{\min}} \le e_k^2 + f_k^2 \le \boldsymbol{V_k^{\max}}, \tag{4f}$$

$$\forall k: \quad C_k = \boldsymbol{F_k}\left(\sum_{km \in \delta(k)} P_{km}\right) + \boldsymbol{G_k}\left(\sum_{km \in \delta(k)} Q_{km}\right). \tag{4g}$$

Here, $\boldsymbol{F_k, G_k}$ are quadratic functions for each k. **Many** possibilities, all structurally similar.

Optimal power flow problem in rectangular coordinates, simplest form

Variables:

- Complex voltages $\boldsymbol{e_k + j f_k}$, power flows $\boldsymbol{P_{km}, Q_{km}}$, auxiliary variables

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

Basic problem

$$\mathbf{min} \quad \sum_{k \in V} C_k$$

$$\text{s.t.} \quad \forall km: \quad P_{km} = \boldsymbol{g_{km}}(e_k^2 + f_k^2) - \boldsymbol{g_{km}}(e_k e_m + f_k f_m) + \boldsymbol{b_{km}}(e_k f_m - f_k e_m) \tag{5a}$$

$$\forall km: \quad Q_{km} = -\boldsymbol{b_{km}}(e_k^2 + f_k^2) + \boldsymbol{b_{km}}(e_k e_m + f_k f_m) + \boldsymbol{g_{km}}(e_k f_m - f_k e_m) \tag{5b}$$

$$\forall km: \quad |P_{km}|^2 + |Q_{km}|^2 \leq \boldsymbol{U_{km}} \tag{5c}$$

$$\forall k: \quad \boldsymbol{P_k^{\mathbf{min}}} \leq \sum_{km \in \delta(k)} P_{km} \leq \boldsymbol{P_k^{\mathbf{max}}} \tag{5d}$$

$$\forall k: \quad \boldsymbol{Q_k^{\mathbf{min}}} \leq \sum_{km \in \delta(k)} Q_{km} \leq \boldsymbol{Q_k^{\mathbf{max}}} \tag{5e}$$

$$\forall k: \quad \boldsymbol{V_k^{\mathbf{min}}} \leq e_k^2 + f_k^2 \leq \boldsymbol{V_k^{\mathbf{max}}}, \tag{5f}$$

$$\forall k: \quad C_k = \boldsymbol{F_k}\left(\sum_{km \in \delta(k)} P_{km}\right) + \boldsymbol{G_k}\left(\sum_{km \in \delta(k)} Q_{km}\right). \tag{5g}$$

Here, $\boldsymbol{F_k, G_k}$ are quadratic functions for each k. **Many** possibilities, all structurally similar.

These are QCQPs, quadratically constrained quadratic programs, with an underlying graph structure.

QCQPs

$$\begin{aligned}
\textbf{min} \quad & x^T M^0 x \ + \ 2c_0^T x \ + \ d_0 && \text{(6a)}\\
\text{s.t.} \quad \forall km: \quad & x^T M^i x \ + \ 2c_i^T x \ + \ d_i \quad \geq \quad 0, \qquad 1 \leq i \leq m, && \text{(6b)}\\
& x \in \mathbb{R}^n. && \text{(6c)}
\end{aligned}$$

Each matrix M^i symmetric.
This description includes linear inequalities, bounds on individual variables, quadratic/linear equations.

QCQPs

$$\begin{aligned} \mathbf{min} \quad & x^T M^0 x \; + \; 2c_0^T x \; + \; d_0 && \text{(7a)} \\ \text{s.t.} \quad \forall km : \quad & x^T M^i x \; + \; 2c_i^T x \; + \; d_i \quad \geq \quad 0, \qquad 1 \leq i \leq m, && \text{(7b)} \\ & x \in \mathbb{R}^n. && \text{(7c)} \end{aligned}$$

Reformulation

observation: $\quad \boldsymbol{x^T M^i x \; + \; 2c_i^T x = (1 \; x^T) \begin{pmatrix} 0 & c_i^T \\ c_i & M^i \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} \; = \; (1 \; x^T) \tilde{M}^i \begin{pmatrix} 1 \\ x \end{pmatrix}}$

definition: for matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\quad \boldsymbol{A \bullet B \; \doteq \; \sum_{i,j} a_{ij} b_{ij}}$

so for vector $\boldsymbol{y}$ and matrix $\boldsymbol{A}$, $\quad \boldsymbol{y^T A y \; = \; A \bullet y y^T}$

So **QCQP** can be rewritten as:

$$\begin{aligned} \boldsymbol{Q^*} \; \doteq \; \mathbf{min} \quad & \tilde{M}^0 \bullet X \; + \; d_0 && \text{(8a)} \\ \text{s.t.} \quad \forall km : \quad & M^i \bullet X \; + \; d_i \quad \geq \quad 0, \qquad 1 \leq i \leq m, && \text{(8b)} \\ & X \in \mathbb{R}^{(n+1)\times(n+1)}, \quad X \succeq 0, \quad \textbf{of rank 1}. && \text{(8c)} \end{aligned}$$

The **semidefinite relaxation** of this problem is:

$$\begin{aligned} \boldsymbol{\tilde{Q}} \; \doteq \; \mathbf{min} \quad & \tilde{M}^0 \bullet X \; + \; d_0 && \text{(9a)} \\ \text{s.t.} \quad \forall km : \quad & M^i \bullet X \; + \; d_i \quad \geq \quad 0, \qquad 1 \leq i \leq m, && \text{(9b)} \\ & X \in \mathbb{R}^{(n+1)\times(n+1)}, \quad X \succeq 0. && \text{(9c)} \end{aligned}$$

$\boldsymbol{\tilde{Q} \; \leq \; Q^*}$

The critical observation

- Lavaei and Low, 2011: the SDP relaxation of AC-OPF **frequently** is very tight
- This spurred much research
- Jabr, Hiskens and Molzahn, others

The critical observation

- Lavaei and Low, 2011: the SDP relaxation of AC-OPF **frequently** is very tight
- This spurred much research
- Jabr, Hiskens and Molzahn, others
- Under constrained conditions, the SDP relaxation can be **weak**

The critical observation

- Lavaei and Low, 2011: the SDP relaxation of AC-OPF **frequently** is very tight
- This spurred much research
- Jabr, Hiskens and Molzahn, others
- Under constrained conditions, the SDP relaxation can be **weak**
- The SDP relaxation can prove unsolvable for larger grids

The critical observation

- Lavaei and Low, 2011: the SDP relaxation of AC-OPF **frequently** is very tight
- This spurred much research
- Jabr, Hiskens and Molzahn, others
- Under constrained conditions, the SDP relaxation can be **weak**
- The SDP relaxation can prove unsolvable for larger grids
- Factoid: there are polynomial-time algorithms for SDP, but require many assumptions
- There is **no** exact algorithm for SDP

The critical observation

- Lavaei and Low, 2011: the SDP relaxation of AC-OPF **frequently** is very tight
- This spurred much research
- Jabr, Hiskens and Molzahn, others
- Under constrained conditions, the SDP relaxation can be **weak**
- The SDP relaxation can prove unsolvable for larger grids
- Factoid: there are polynomial-time algorithms for SDP, but require many assumptions
- There is **no** exact algorithm for SDP
- Lavaei, Low, Hiskens-Molzahn:
  when the underlying network has **low tree-width**, the SDP relaxation can be solved much faster
  why: standard SDP solvers can leverage low tree-width
- What exactly is tree-width?

Tree-width

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

A tree-decomposition of $\boldsymbol{G}$ is a pair $\boldsymbol{(T, Q)}$ where:

- $\boldsymbol{T}$ is a tree. **Not** a subtree of $\boldsymbol{G}$, just a tree

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:

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.

Tree-width

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

A tree-decomposition of $\boldsymbol{G}$ is a pair $\boldsymbol{(T, Q)}$ where:

- $\boldsymbol{T}$ is a tree. **Not** a subtree of $\boldsymbol{G}$, just a tree
- For each vertex $\boldsymbol{t}$ of $\boldsymbol{T}$, $\boldsymbol{Q_t}$ is a subset of $\boldsymbol{V(G)}$. These subsets satisfy the two properties:

  (1) For each vertex $\boldsymbol{v}$ of $\boldsymbol{G}$, the set $\boldsymbol{\{t \in V(T) : v \in Q_t\}}$ is a **subtree** of $\boldsymbol{T}$, denoted $\boldsymbol{T_v}$.

  (2) For each edge $\boldsymbol{\{u, v\}}$ of $\boldsymbol{G}$, the two subtrees $\boldsymbol{T_u}$ and $\boldsymbol{T_v}$ **intersect**.

Tree-width

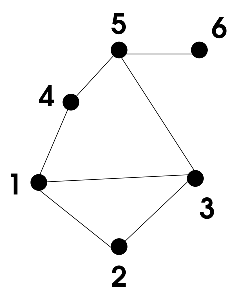
Let $\boldsymbol{G}$ be an undirected graph with vertices $\boldsymbol{V(G)}$ and edges $\boldsymbol{E(G)}$.

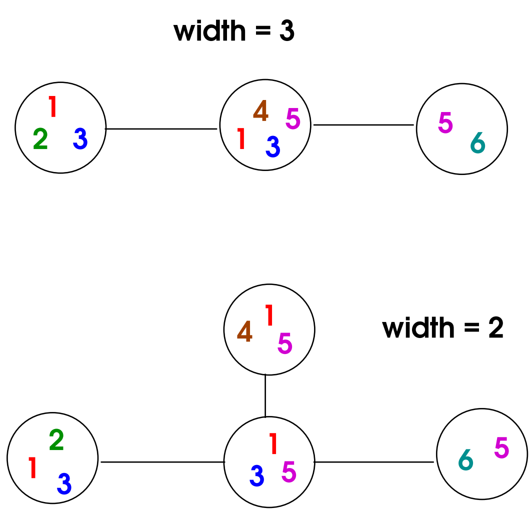
A tree-decomposition of $\boldsymbol{G}$ is a pair $(\boldsymbol{T}, \boldsymbol{Q})$ where:

- $\boldsymbol{T}$ is a tree. **Not** a subtree of $\boldsymbol{G}$, just a tree
- For each vertex $\boldsymbol{t}$ of $\boldsymbol{T}$, $\boldsymbol{Q_t}$ is a subset of $\boldsymbol{V(G)}$. These subsets satisfy the two properties:

  (1) For each vertex $\boldsymbol{v}$ of $\boldsymbol{G}$, the set $\boldsymbol{\{t \in V(T) \ : \ v \in Q_t\}}$ is a **subtree** of $\boldsymbol{T}$, denoted $\boldsymbol{T_v}$.

  (2) For each edge $\boldsymbol{\{u, v\}}$ of $\boldsymbol{G}$, the two subtrees $\boldsymbol{T_u}$ and $\boldsymbol{T_v}$ **intersect**.
- The **width** of $(\boldsymbol{T}, \boldsymbol{Q})$ is $\boldsymbol{\max_{t \in T} |Q_t| - 1}$.



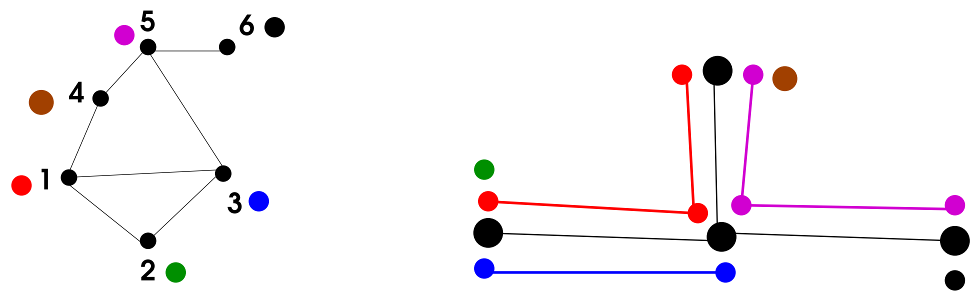


Tree-width

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 $\max_{t \in T} |Q_t| - 1$.



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

History

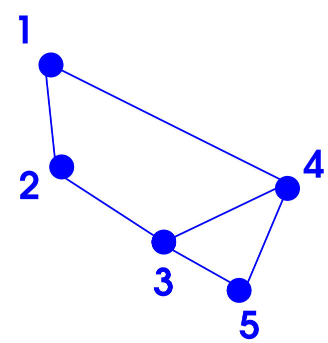
Fulkerson and Gross (1965), binary packing integer programs

$$\begin{aligned} \text{IP} = \max \quad & c^T x && \text{(10a)} \\ \text{s.t.} \quad & Ax \ \leq \ b, && \text{(10b)} \\ & x \in \{0,1\}^n && \text{(10c)} \end{aligned}$$

Here, $\boldsymbol{A}$ is has $\mathbf{0,1}$-valued entries. **Idea:** use the structure of $\boldsymbol{A}$.
The **intersection graph of A**, $\boldsymbol{G_A}$, has:

- A vertex for each column of A.
- An edge between two columns $\boldsymbol{j,k}$ if there is a row $\boldsymbol{i}$ with $\boldsymbol{a_{ij} \neq 0,\ a_{ik} \neq 0}$.

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{bmatrix} 1 & & & 1 & \\ 1 & & & & \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & 1 \end{bmatrix} \end{array}$$



History

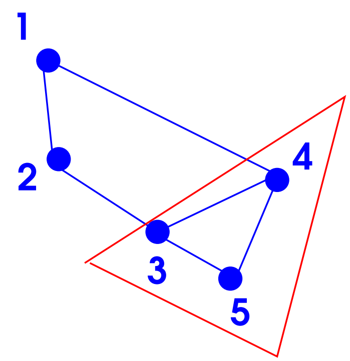
Fulkerson and Gross (1965), binary packing integer programs

$$\mathrm{IP} = \max \quad c^T x \tag{11a}$$
$$\text{s.t.} \quad Ax \leq b, \tag{11b}$$
$$x \in \{0,1\}^n \tag{11c}$$

Here, $\boldsymbol{A}$ is has $\mathbf{0,1}$-valued entries. **Idea:** use the structure of $\boldsymbol{A}$.
The **intersection graph of A**, $\boldsymbol{G_A}$, has:

- A vertex for each column of A.
- An edge between two columns $\boldsymbol{j,k}$ if there is a row $\boldsymbol{i}$ with $\boldsymbol{a_{ij} \neq 0}$, $\boldsymbol{a_{ik} \neq 0}$.

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{bmatrix} 1 & & & 1 & \\ 1 & & & & \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & 1 \end{bmatrix} \end{array}$$



Each row of A induces a clique of $\boldsymbol{G_A}$.

History

Fulkerson and Gross (1965), binary packing integer programs

$$
\begin{aligned}
\mathrm{IP} = \max \quad & c^T x && (12a)\\
\text{s.t.} \quad & Ax \le b, && (12b)\\
& x \in \{0,1\}^n && (12c)
\end{aligned}
$$

Here, $\boldsymbol{A}$ is has $\mathbf{0,1}$-valued entries. **Idea:** use the structure of $\boldsymbol{A}$.
The **intersection graph of A**, $\boldsymbol{G_A}$, has:

- A vertex for each column of A.
- An edge between two columns $\boldsymbol{j,k}$ if there is a row $\boldsymbol{i}$ with $\boldsymbol{a_{ij} \neq 0,\ a_{ik} \neq 0}$.

Theorem. If $\boldsymbol{G_A}$ is an **interval graph**, then

$$
\begin{aligned}
\mathrm{IP} = \mathrm{LP} = \max \quad & c^T x && (13a)\\
\text{s.t.} \quad & Ax \le b, && (13b)\\
& x \in [0,1]^n. && (13c)
\end{aligned}
$$

(so IP = value of its continuous relaxation).

A graph $\boldsymbol{G = (V,E)}$ is an interval graph, if there is a **path** $\boldsymbol{P}$, and a family of subpaths $\boldsymbol{P_v}$ (one for each $\boldsymbol{v \in V}$), such that

- For each **pair of vertices** $\boldsymbol{u}$ and $\boldsymbol{v}$ of $\boldsymbol{G}$, we have $\boldsymbol{\{u,v\} \in E}$ **whenever** $\boldsymbol{P_u}$ and $\boldsymbol{P_v}$ intersect.
- The largest clique size of $\boldsymbol{G}$ is $\boldsymbol{\max_{p \in P} |\{v \in V \ : \ p \in P_v\}|}$.
  (The maximum number of subpaths that simultaneously overlap anywere on $\boldsymbol{P}$)

$$\begin{aligned} \text{IP} = \max \quad & c^T x && (14a)\\ \text{s.t.} \quad & Ax \leq b, && (14b)\\ & x \in \{0,1\}^n && (14c) \end{aligned}$$

The **intersection graph of A**, $\boldsymbol{G_A}$, has:

- A vertex for each column of A, an edge between two columns $\boldsymbol{j, k}$ if there is a row $\boldsymbol{i}$ with $\boldsymbol{a_{ij} \neq 0,\ a_{ik} \neq 0}$.

Definition: (Gavril, 1974) A graph $\boldsymbol{G = (V, E)}$ is **chordal**, if there exists

- A **tree** $\boldsymbol{T}$, and a family of trees $\boldsymbol{P_v}$ (one for each $\boldsymbol{v \in V}$), such that
- For each **pair of vertices** $\boldsymbol{u}$ and $\boldsymbol{v}$ of $\boldsymbol{G}$, we have $\boldsymbol{\{u, v\} \in E}$ **whenever** $\boldsymbol{T_u}$ and $\boldsymbol{T_v}$ intersect.
- The largest clique size of $\boldsymbol{G}$ is $\boldsymbol{\max_{t \in T} |\{v \in V\ :\ t \in T_v\}|}$.
  (The maximum number of subtrees that simultaneously overlap anywere on $\boldsymbol{T}$)

(equivalent: a graph is chordal iff every cycle of length > 3 has a chord).

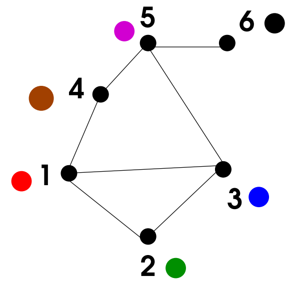
Contrast with tree-decompositions

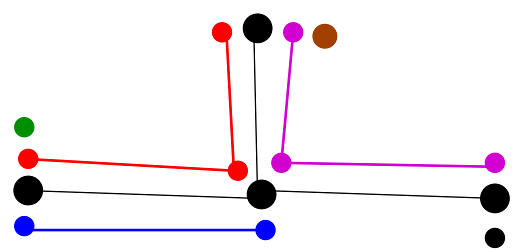
A tree-decomposition of G is a pair (T, Q) where:

- 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 $\max_{t \in T} |Q_t| - 1$.





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

So: A graph G has a tree-decomposition of width w iff there is a **chordal supergraph** of G of clique number $w + 1$.

$$
\begin{aligned}
\text{IP} = \max \quad & c^T x && (15a)\\
\text{s.t.} \quad & Ax \le b, && (15b)\\
& x \in \{0,1\}^n && (15c)
\end{aligned}
$$

The **intersection graph of A**, G_A, has:

- A vertex for each column of A, an edge between two columns j, k if there is a row i with $a_{ij} \neq 0,\ a_{ik} \neq 0$.

Definition: (Gavril, 1974) A graph $G = (V, E)$ is **chordal**, if there exists

- A **tree** T, and a family of subtrees P_v (one for each $v \in V$), such that
- For each **pair of vertices** u and v of G, we have $\{u, v\} \in E$ iff T_u and T_v intersect.
- The largest clique size of G is $\max_{t \in T} |\{v \in V \ : \ t \in T_v\}|$.
  (The maximum number of subtrees that simultaneously overlap anywere on T)

(equivalent: a graph is chordal iff every cycle of length > 3 has a chord).

Theorem. If G_A is **chordal**, then

$$
\begin{aligned}
\text{IP} = \text{LP} = \max \quad & c^T x && (16a)\\
\text{s.t.} \quad & Ax \le b, && (16b)\\
& x \in [0,1]^n. && (16c)
\end{aligned}
$$

(so IP = value of its continuous relaxation).

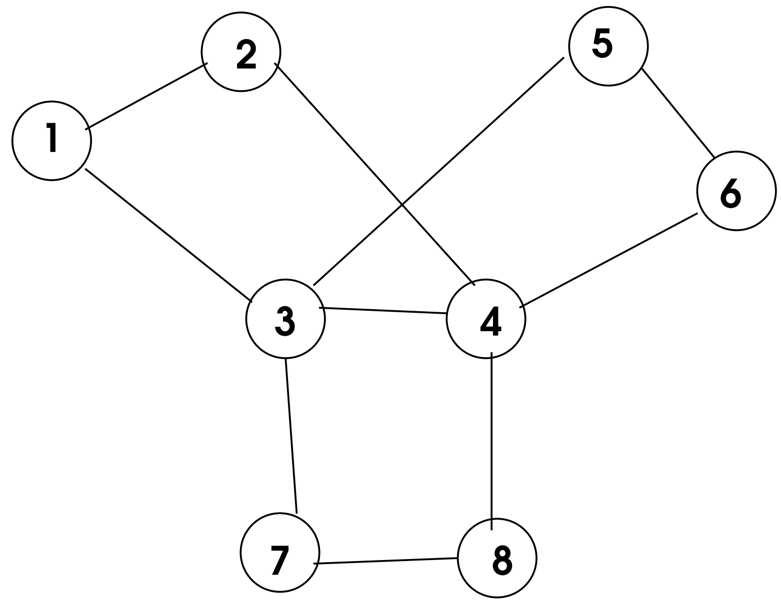
Chordal graphs are “nice.” In fact, they are **perfect**.

Why small tree-width helps

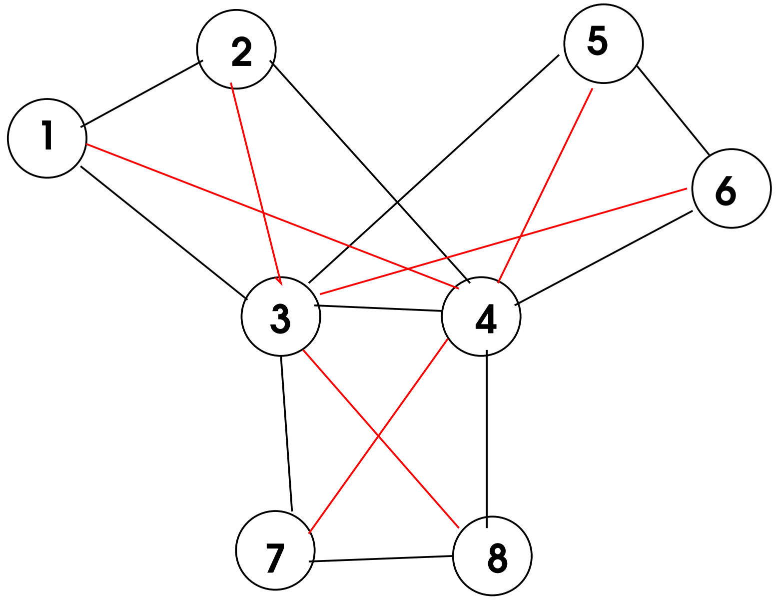
Cholesky factorization of:

$$A = \begin{pmatrix} * & * & * & & & & & \\ * & * & & * & & & & \\ * & & * & * & * & & * & \\ & * & * & * & & * & & * \\ & & * & & * & * & & \\ & & & * & * & * & & \\ & & * & & & & * & * \\ & & & * & & & * & * \end{pmatrix}$$

Cholesky factorization of:



Chordal supergraph:



Pivoting order: 1, 2, 5, 6, 7, 8, 3, 4

Graph Minors Project: Robertson and Seymour, 1983 - 2004

$\longrightarrow$ Tree-width as a measure of the complexity of a graph

CAUTION

CAUTION

sparsity $\neq$ small tree-width

CAUTION

sparsity $\neq$ small tree-width

$\exists$ **graphs of max deg 3 and arbitrarily high tree-width**

Graph Minors Project: Robertson and Seymour, 1983 - 2004

$\longrightarrow$ Tree-width as a measure of the complexity of a graph

- Algorithms community: small tree-width makes hard problems easy (late 1980s)

- Many NP-hard problems can be solved in polynomial time on graphs with small tree-width:
  TSP, max. clique, graph coloring, ...

Graph Minors Project: Robertson and Seymour, 1983 - 2004

$\longrightarrow$ Tree-width as a measure of the complexity of a graph

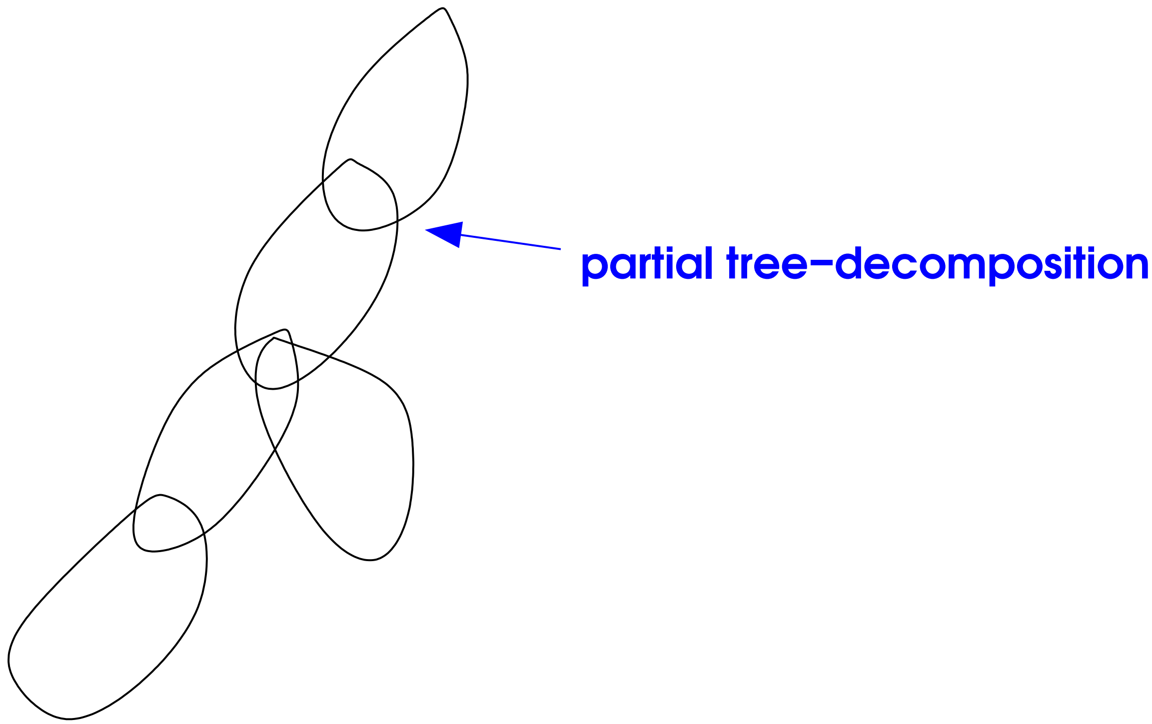
- Algorithms community: small tree-width makes hard problems easy (late 1980s)

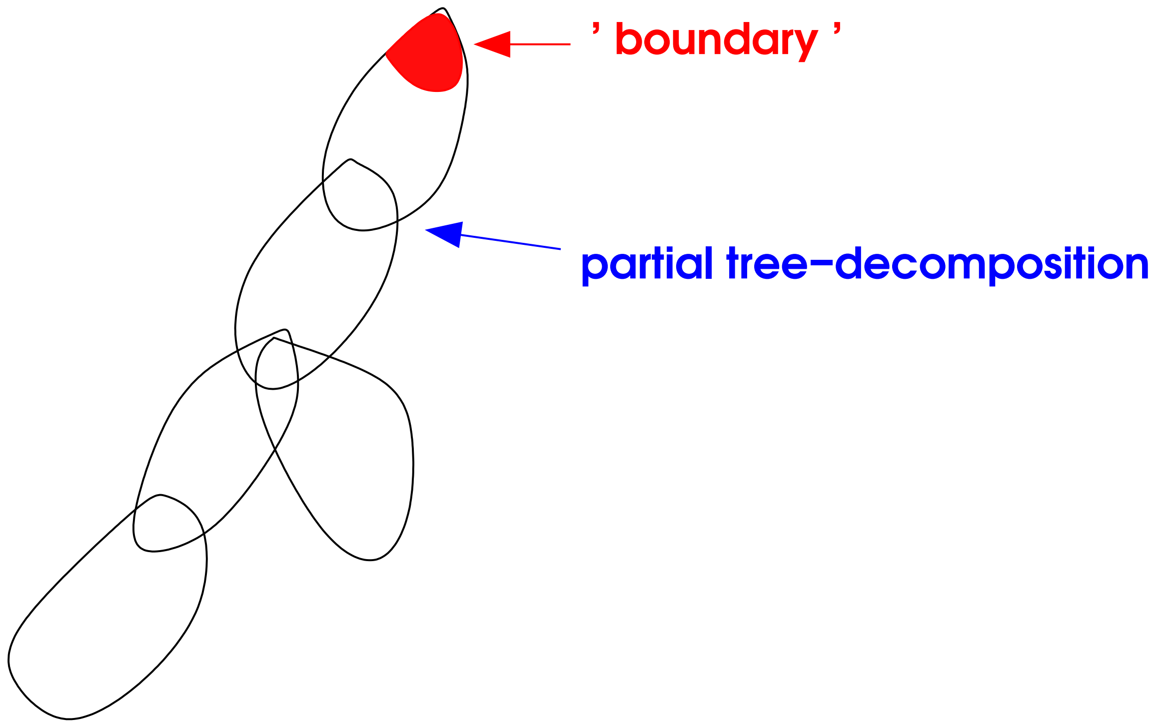
- Many NP-hard problems can be solved in polynomial time on graphs with small tree-width:
  TSP, max. clique, graph coloring, ...

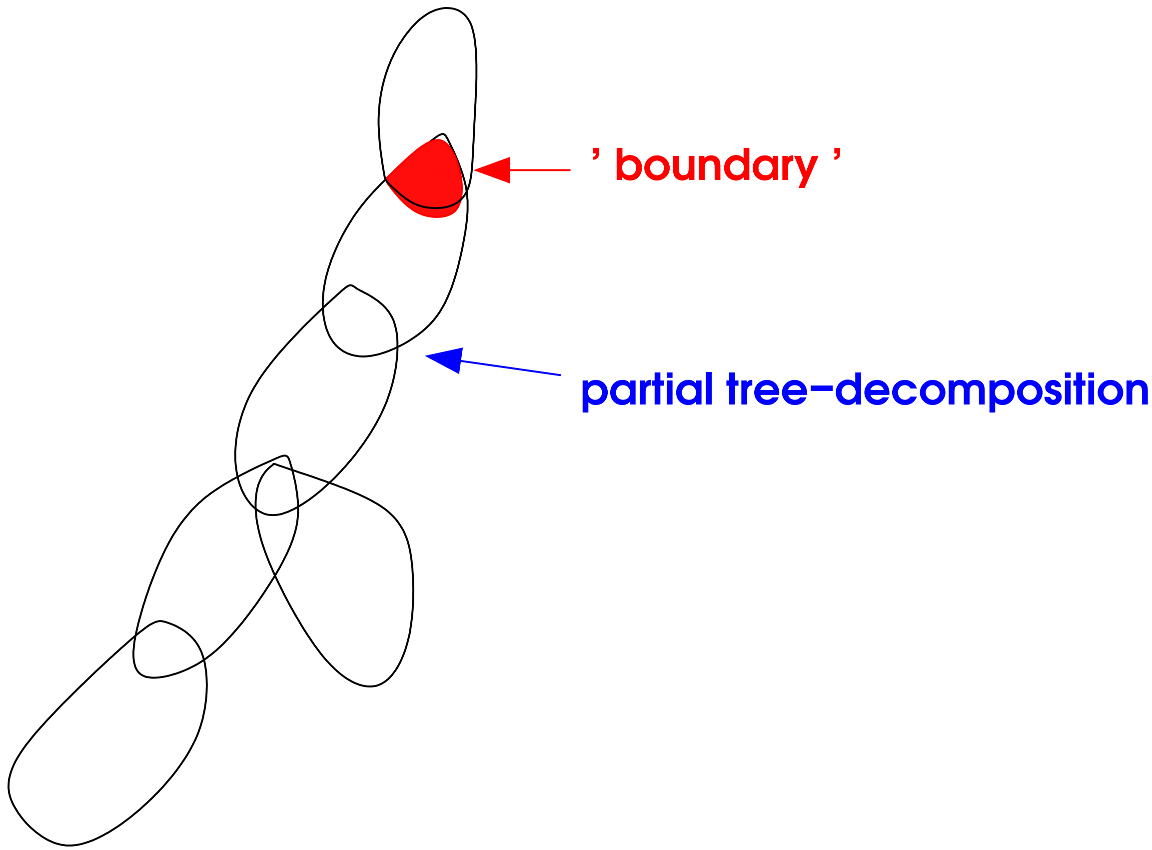
- Fellows & Langston; Bienstock & Langston; Arnborg, Corneil & Proskurowski; many other authors

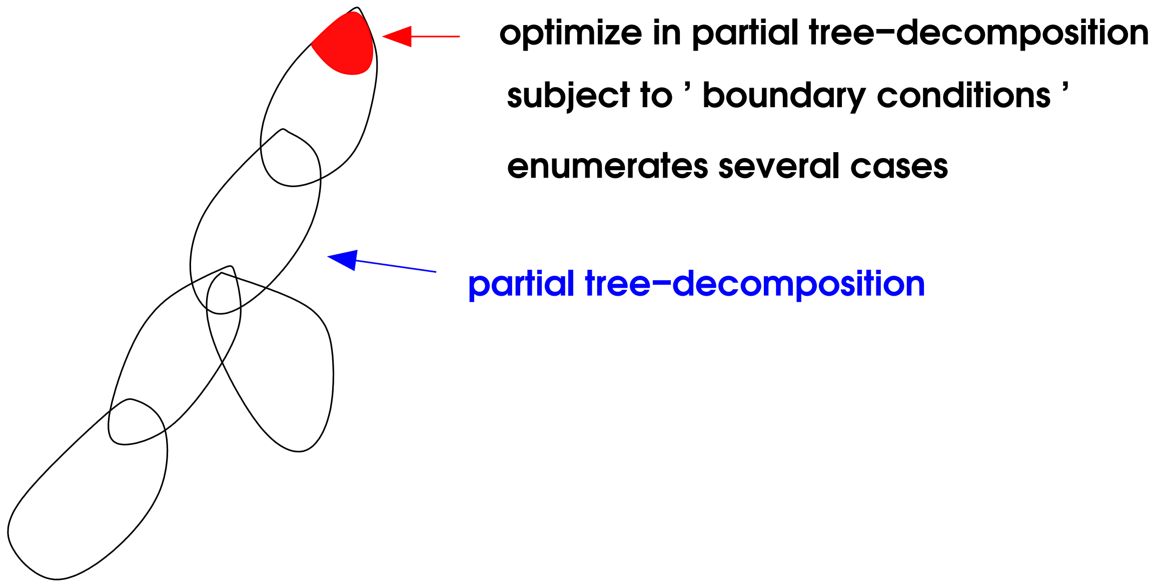
- Common thread: exploit tree-decomposition to obtain good algorithms

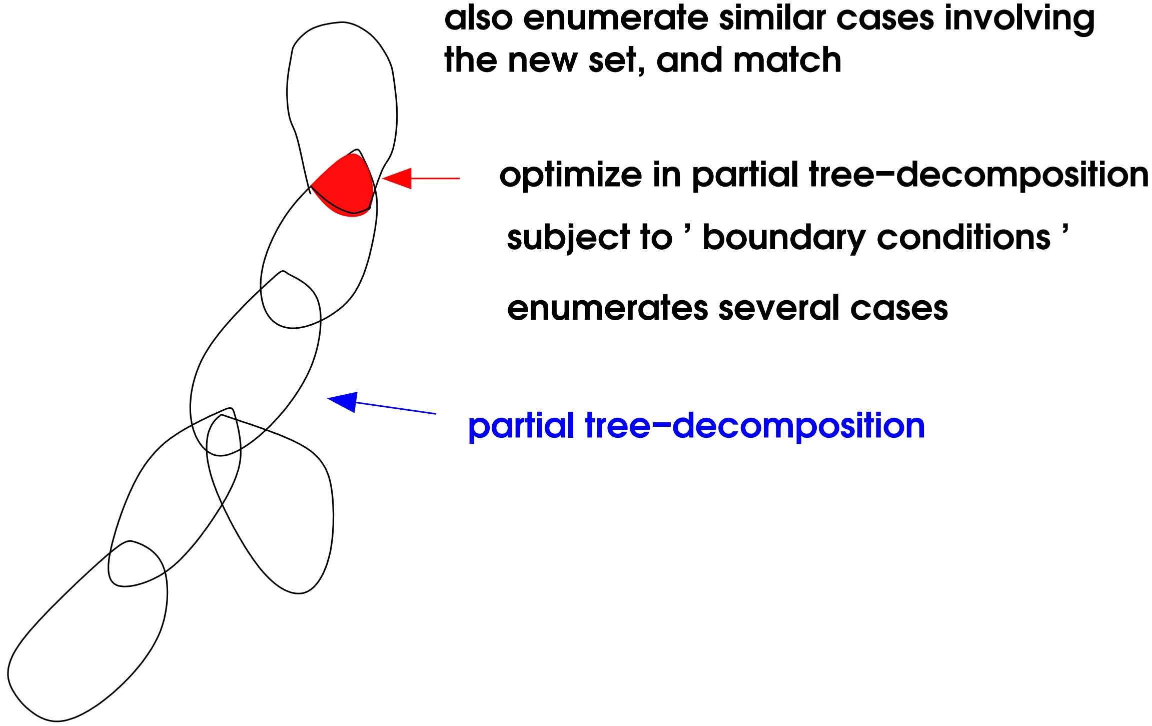
- So-called "non-sequential dynamic programming"


partial tree-decomposition


' boundary '
partial tree-decomposition


' boundary '
partial tree-decomposition


optimize in partial tree−decomposition
subject to ' boundary conditions '
enumerates several cases
partial tree−decomposition


also enumerate similar cases involving
the new set, and match
optimize in partial tree-decomposition
subject to ' boundary conditions '
enumerates several cases
partial tree-decomposition

Graph Minors Project: Robertson and Seymour, 1983 - 2004

$\longrightarrow$ Tree-width as a measure of the complexity of a graph

- Algorithms community: small tree-width makes hard problems easy (late 1980s)

- Many NP-hard problems can be solved in polynomial time on graphs with small tree-width:
  TSP, max. clique, graph coloring, ...

- Fellows & Langston; Bienstock & Langston; Arnborg, Corneil & Proskurowski; many other authors

- Common thread: exploit tree-decomposition to obtain good algorithms

- So-called "non-sequential dynamic programming"

$\longrightarrow$ Can we do the same for OPF ?

Theorem: Given an instance of **AC-OPF** on a graph with a tree-decomposition of width $\boldsymbol{\omega}$, and $\boldsymbol{n}$ buses, and $\mathbf{0 < \epsilon < 1}$,

there is a linear program **LP** such that:

(a) The number of variables and constraints is $\boldsymbol{O(\, 2^{2\omega}\, \omega\, n\, \epsilon^{-1} \log_2 \epsilon^{-1}\,)}$.

(b) An optimal solution to **LP** solves **AC-OPF**, within tolerance $\boldsymbol{\epsilon}$.

More generic statement for AC-OPF

$$
\begin{aligned}
\mathbf{min} \quad & \sum_{k \in V} C_k \\
\text{s.t.} \quad \forall km: \quad & P_{km} \;=\; \boldsymbol{g_{km}}(e_k^2 + f_k^2) - \boldsymbol{g_{km}}(e_k e_m + f_k f_m) + \boldsymbol{b_{km}}(e_k f_m - f_k e_m) \\
\forall km: \quad & Q_{km} \;=\; \boldsymbol{-b_{km}}(e_k^2 + f_k^2) + \boldsymbol{b_{km}}(e_k e_m + f_k f_m) + \boldsymbol{g_{km}}(e_k f_m - f_k e_m) \\
\forall km: \quad & |P_{km}|^2 + |Q_{km}|^2 \;\le\; \boldsymbol{U_{km}} \\
\forall k: \quad & P_k \;=\; \sum_{km \,\in\, \delta(k)} P_{km}; \qquad \boldsymbol{P_k^{\mathbf{min}}} \;\le\; P_k \;\le\; \boldsymbol{P_k^{\mathbf{max}}} \\
\forall k: \quad & Q_k \;=\; \sum_{km \,\in\, \delta(k)} Q_{km}; \qquad \boldsymbol{Q_k^{\mathbf{min}}} \;\le\; Q_k \;\le\; \boldsymbol{Q_k^{\mathbf{max}}} \\
\forall k: \quad & \left(\boldsymbol{V_k^{\mathbf{min}}}\right)^{\mathbf{2}} \;\le\; e_k^2 + f_k^2 \;\le\; \left(\boldsymbol{V_k^{\mathbf{max}}}\right)^{\mathbf{2}} \\
\forall k: \quad & C_k \;=\; \boldsymbol{F_k}\,(P_k, Q_k, e_k, f_k) \;+\; \sum_{km \,\in\, \delta(k)} \boldsymbol{H_{km}}(P_{km}, Q_{km}, e_k, f_k, e_m, f_m)
\end{aligned}
$$

Here, the $\boldsymbol{F_k}$ and $\boldsymbol{H_{km}}$ are quadratics.

A generalization: graphical QCQPs (abridged)

Inputs:

(1) An undirected graph $\boldsymbol{H}$.

(2) For each vertex $\boldsymbol{v}$ of $\boldsymbol{H}$ a set $\boldsymbol{J(v)}$, and for $\boldsymbol{j \in J(v)}$ there is a real variable $\boldsymbol{x_j}$.
Write $\boldsymbol{\mathcal{V}} = \cup_{\boldsymbol{v \in V(H)}} \boldsymbol{J(v)}$.

(3) For each edge $\boldsymbol{\{v, u\}}$ denote by $\boldsymbol{x^{v,u}}$ the vector of all $\boldsymbol{x_j}$ for $\boldsymbol{j \in J(v) \cup J(u)}$.

(4) For each vertex $\boldsymbol{v}$, and each edge $\boldsymbol{\{v, u\}}$ a family of quadratics $\boldsymbol{p^k_{v,u}(x^{v,u})}$ for $\boldsymbol{k = 1, \ldots, N(v)}$.

(5) A vector $\boldsymbol{c} \in \mathbb{R}^{\boldsymbol{\mathcal{V}}}$.

A generalization: graphical QCQPs (abridged)

Inputs:

(1) An undirected graph $\boldsymbol{H}$.

(2) For each vertex $\boldsymbol{v}$ of $\boldsymbol{H}$ a set $\boldsymbol{J(v)}$, and for $\boldsymbol{j \in J(v)}$ there is a real variable $\boldsymbol{x_j}$.
Write $\boldsymbol{\mathcal{V} = \cup_{v \in V(H)} J(v)}$.

(3) For each edge $\boldsymbol{\{v, u\}}$ denote by $\boldsymbol{x^{v,u}}$ the vector of all $\boldsymbol{x_j}$ for $\boldsymbol{j \in J(v) \cup J(u)}$.

(4) For each vertex $\boldsymbol{v}$, and each edge $\boldsymbol{\{v, u\}}$ a family of quadratics $\boldsymbol{p^k_{v,u}(x^{v,u})}$ for $\boldsymbol{k = 1, \ldots, N(v)}$.

(5) A vector $\boldsymbol{c} \in \mathbb{R}^{\boldsymbol{\mathcal{V}}}$.

Problem:

$$\text{(GQCQP):} \qquad \min \ c^T x$$

$$\text{subject to:} \qquad \sum_{u \in \delta(v)} p_{v,u,k}(x^{v,u}) \ \geq \ 0, \quad v \in V(H), \quad k = 1, \ldots, N(v)$$

$$0 \ \leq \ x_j \ \leq \ 1, \quad \forall j \in \mathcal{V}.$$

A generalization: mixed-integer graphical QCQPs (abridged)

Inputs:

(1) An undirected graph $\boldsymbol{H}$.

(2) For each vertex $\boldsymbol{v}$ of $\boldsymbol{H}$ a set $\boldsymbol{J(v)}$, and for $\boldsymbol{j} \in \boldsymbol{J(v)}$ there is a real variable $\boldsymbol{x_j}$.
Write $\boldsymbol{\mathcal{V}} = \cup_{\boldsymbol{v} \in \boldsymbol{V(H)}} \boldsymbol{J(v)}$.

(3) For each edge $\boldsymbol{\{v, u\}}$ denote by $\boldsymbol{x^{v,u}}$ the vector of all $\boldsymbol{x_j}$ for $\boldsymbol{j} \in \boldsymbol{J(v)} \cup \boldsymbol{J(u)}$.

(4) For each vertex $\boldsymbol{v}$, and each edge $\boldsymbol{\{v, u\}}$ a family of quadratics $\boldsymbol{p^k_{v,u}(x^{v,u})}$ for $\boldsymbol{k = 1, \ldots, N(v)}$.

(5) A vector $\boldsymbol{c} \in \mathbb{R}^{\boldsymbol{\mathcal{V}}}$.

(6) A partition $\boldsymbol{\mathcal{V}} = \boldsymbol{V_Z} \cup \boldsymbol{V_R}$.

Problem:

(MGP): $\min\ c^T x$

subject to:

$$\sum_{u \in \delta(v)} p_{v,u,k}(x^{v,u}) \ \geq \ 0, \quad v \in V(H), \quad k = 1, \ldots, N(v)$$

$$0 \ \leq \ x_j \ \leq \ 1 \quad \forall\, j \in \mathcal{V}_R; \quad x_j \ = \ 0 \ \textbf{or} \ 1 \quad \forall\, j \in \mathcal{V}_Z.$$

(1) An undirected graph $\boldsymbol{H}$.

(2) For each vertex $\boldsymbol{v}$ of $\boldsymbol{H}$ a set $\boldsymbol{J(v)}$, and for $\boldsymbol{j \in J(v)}$ there is a real variable $\boldsymbol{x_j}$. Write $\boldsymbol{\mathcal{V} = \cup_{v \in V(H)} J(v)}$.

(3) For each edge $\boldsymbol{\{v, u\}}$ denote by $\boldsymbol{x^{v,u}}$ the vector of all $\boldsymbol{x_j}$ for $\boldsymbol{j \in J(v) \cup J(u)}$.

(4) For each vertex $\boldsymbol{v}$, and each edge $\boldsymbol{\{v, u\}}$ a family of **polynomials** $\boldsymbol{p^k_{v,u}(x^{v,u})}$ for $\boldsymbol{k = 1, \ldots, N(v)}$.

(5) A vector $\boldsymbol{c} \in \mathbb{R}^{\boldsymbol{\mathcal{V}}}$.

(6) A partition $\boldsymbol{\mathcal{V}} = \boldsymbol{V_Z \cup V_R}$.

$$\textbf{(MGP)}: \quad \min \ c^T x \tag{20a}$$

$$\text{subject to:} \quad \sum_{u \in \delta(v)} p_{v,u,k}(x^{v,u}) \ \geq \ 0, \quad v \in V(H), \quad k = 1, \ldots, N(v) \tag{20b}$$

$$0 \leq x_j \leq 1 \quad \forall j \in \mathcal{V}_R; \quad x_j = 0 \ \textbf{or} \ 1 \quad \forall j \in \mathcal{V}_Z. \tag{20c}$$

Theorem: Given an instance of **MGP** on a graph with a tree-decomposition of width $\boldsymbol{\omega}$, there is an equivalent instance of **MGP** on a graph

- With tree-width $\boldsymbol{\leq 2\omega + 1}$

- Of maximum degree $\boldsymbol{3}$.

Remark. If we start with an instance of AC-OPF, the equivalent problem is no longer an AC-OPF problem.

Approximation (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

$$\boldsymbol{x} \approx \sum_{i=1}^{L} \mathbf{2}^{-i} \boldsymbol{y}_i$$

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

$$\boldsymbol{x} \leq \sum_{i=1}^{L} \mathbf{2}^{-i} \boldsymbol{y}_i \leq \boldsymbol{x} + \boldsymbol{\gamma}.$$

So: given an instance of $\boldsymbol{MGP}$, approximate each continuous variable $\boldsymbol{x_j}$ in this manner.

Theorem: Consider an instance $\mathcal{I}$ of problem **MGP**, with n variables. Then there is another instance, $\mathcal{B}$ of **MGP**, with

(1) $\mathcal{B}$ is defined on the same graph as $\mathcal{I}$.

(2) all variables in $\mathcal{B}$ are binary.

(3) For each continuous variable x_j of $\mathcal{I}$, we now have $\log_2 J^* \log \epsilon^{-1}$ binary variables used to approximate x_j.

(4) Solving $\mathcal{B}$ to exact optimality yields a solution to $\mathcal{I}$ within tolerance ϵ.

J^* = size of largest set $J(v)$. (AC-OPF $\Rightarrow$ $J^* = 2$)

Review

(1) A mixed-integer, graphical polynomial optimization problem on a graph with a tree-decomposition of width ω.

Review

(1) A mixed-integer, graphical polynomial optimization problem on a graph with a tree-decomposition of width $\boldsymbol{\omega}$.

$$\downarrow$$

(2) An equivalent mixed-integer, graphical polynomial optimization problem on a graph with a tree-decomposition of width $\boldsymbol{O(\omega)}$ and **degree** $\boldsymbol{\leq 3}$.

Review

(1) A mixed-integer, graphical polynomial optimization problem on a graph with a tree-decomposition of width $\boldsymbol{\omega}$.

$\downarrow$

(2) An equivalent mixed-integer, graphical polynomial optimization problem on a graph with a tree-decomposition of width $\boldsymbol{O(\omega)}$ and **degree** $\leq \mathbf{3}$.

$\downarrow$

(3) An all-binary, graphical polynomial optimization problem on the same graph which is equivalent to the problem in **(2)** within tolerance $\boldsymbol{\epsilon}$. The sets $\boldsymbol{J(v)}$ have grown by a factor of $\mathbf{log_2}\, \boldsymbol{J^*}\ \mathbf{log_2}\, \boldsymbol{\epsilon^{-1}}$.

Ancient History of this Talk

Fulkerson and Gross (1965), binary packing integer programs

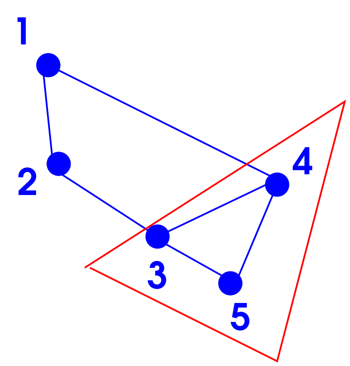
$$
\begin{aligned}
\mathrm{IP} = \max \quad & c^T x && (21a)\\
\text{s.t.} \quad & Ax \leq b, && (21b)\\
& x \in \{0,1\}^n && (21c)
\end{aligned}
$$

Here, $\boldsymbol{A}$ is has $\mathbf{0,1}$-valued entries. **Idea:** use the structure of $\boldsymbol{A}$.

The **intersection graph of A**, $\boldsymbol{G_A}$, has:

- A vertex for each column of A.
- An edge between two columns $\boldsymbol{j,k}$ if there is a row $\boldsymbol{i}$ with $\boldsymbol{a_{ij} \neq 0}$, $\boldsymbol{a_{ik} \neq 0}$.

$$
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array}\\
\begin{bmatrix}
1 & & & 1 & \\
1 & & & & \\
1 & 1 & & & \\
& 1 & 1 & & \\
& & 1 & 1 & 1
\end{bmatrix}
\end{array}
$$



Each row of A induces a clique of $\boldsymbol{G_A}$.

Review

(1) A mixed-integer, graphical polynomial optimization problem on a graph with a tree-decomposition of width $\boldsymbol{\omega}$.

(2) An equivalent mixed-integer, graphical polynomial optimization problem on a graph with a tree-decomposition of width $\boldsymbol{O(\omega)}$ and **degree** $\leq \mathbf{3}$.

(3) An all-binary, graphical polynomial optimization problem on the same graph which is equivalent to the problem in **(2)** within tolerance $\boldsymbol{\epsilon}$. The sets $\boldsymbol{J(v)}$ have grown by a factor of $\boldsymbol{\log_2 J^* \ \log_2 \epsilon^{-1}}$.

(4) Corollary. The intersection graph of the problem in **(3)** has a tree-decomposition of width at most

$$\boldsymbol{O(\omega \, J^* \log_2 J^* \ \log_2 \epsilon^{-1})}$$

Note: There are **two** graphs. The initial graph used to define the problem, and the intersection graph for the constraints in **(3)**.

Pièce de Résistance

Theorem. Given an all-binary problem on n variables and whose intersection graph has a tree-decomposition of width k, then there is an exact linear programming representation using

$$O(2^k n)$$

variables and constraints.

Construction similar to Lovász-Schrijver, Sherali-Adams, Lasserre, Bienstock-Zuckerberg

Pièce de Résistance

Theorem. Given an all-binary problem on n variables and whose intersection graph has a tree-decomposition of width k, then there is an exact linear programming representation using

$$O(2^k n)$$

variables and constraints.

Construction similar to Lovász-Schrijver, Sherali-Adams, Lasserre, Bienstock-Zuckerberg

(A) A mixed-integer, graphical polynomial optimization problem, with N variables, on a graph with a tree-decomposition of width ω.
J^* = size of largest set $J(v)$. (AC-OPF $J^* = 2$)

Pièce de Résistance

Theorem. Given an all-binary problem on $\boldsymbol{n}$ variables and whose intersection graph has a tree-decomposition of width $\boldsymbol{k}$, then there is an exact linear programming representation using

$$\boldsymbol{O(2^k n)}$$

variables and constraints.

Construction similar to Lovász-Schrijver, Sherali-Adams, Lasserre, Bienstock-Zuckerberg

(A) A mixed-integer, graphical polynomial optimization problem, with $\boldsymbol{N}$ variables, on a graph with a tree-decomposition of width $\boldsymbol{\omega}$.
$\boldsymbol{J^*}$ = size of largest set $\boldsymbol{J(v)}$. (AC-OPF $\boldsymbol{J^* = 2}$)

$$\downarrow$$

(B) A linear program that solves the problem in **(A)** within tolerance $\boldsymbol{\epsilon}$, of size

$$\boldsymbol{O(\, 2^{O(\omega J^*)}\, \omega\, J^*\, \epsilon^{-1}\, N)}$$

Should we able to do better?

Probably.

But.

- There are trivial AC-OPF problems where there is a unique feasible solution and it is irrational.
  Under the bit model of computing we cannot produce an "exact" answer.

- AC-OPF is weakly NP-hard on *trees*. Lavaei and Low (2011), a more recent proof by Coffrin and van Hentenryck.

- AC-OPF is strongly NP-hard on general graphs. A. Verma (2009). So no strong approximation algorithms exist unless P = NP.