Graph-Theoretic Convexification of Polynomial Optimization with Applications to Power Systems and Distributed Control

Javad Lavaei

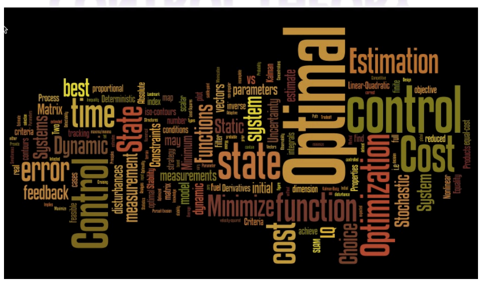
Industrial Engineering and Operations Research
University of California, Berkeley

Optimization

TRANSPORTATION

ROBOTICS

CONTROL THEORY



OPTIMIZATION

MACHINE LEARNING

FINANCE

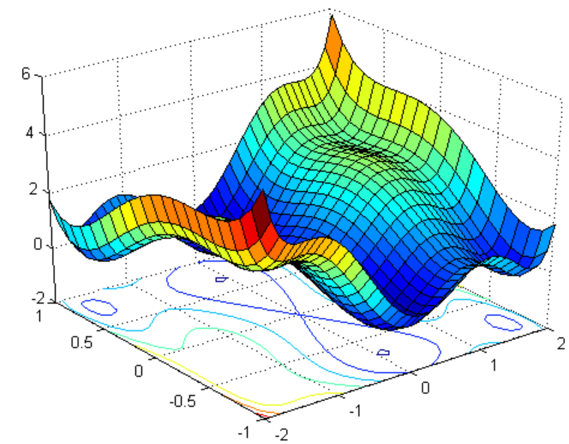
POWER SYSTEMS

Optimization

Optimization problem:

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{Subject to} & x \in \Omega \end{array}$$

- **Universal challenges:**
  1) Continuous nonlinearity
  2) Discrete nonlinearity
  3) Large-scale nature
  4) Unreliable problem data



Optimization

Optimization problem:

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{Subject to} & x \in \Omega \end{array}$$

❑ **Universal challenges:**

1) Continuous nonlinearity
2) Discrete nonlinearity
3) Large-scale nature
4) Unreliable problem data

500 decisions

⇩

Each decision = 0 or 1

⇩

Possibilities = almost infinity

Optimization

Optimization problem:

$$\begin{array}{ll}\text{Minimize} & f(x) \\ \text{Subject to} & x \in \Omega\end{array}$$

- **Universal challenges:**
  1) Continuous nonlinearity
  2) Discrete nonlinearity
  3) Large-scale nature
  4) Unreliable problem data

Energy systems

↓

Tens of thousands of variables

↓

Millions of constraints

Optimization

Optimization problem:

$$\begin{aligned} &\text{Minimize} \quad f(x) \\ &\text{Subject to} \quad x \in \Omega \end{aligned}$$

- **Universal challenges:**
  1) Continuous nonlinearity
  2) Discrete nonlinearity
  3) Large-scale nature
  4) Unreliable problem data

System

↓

Data

↓

Optimization

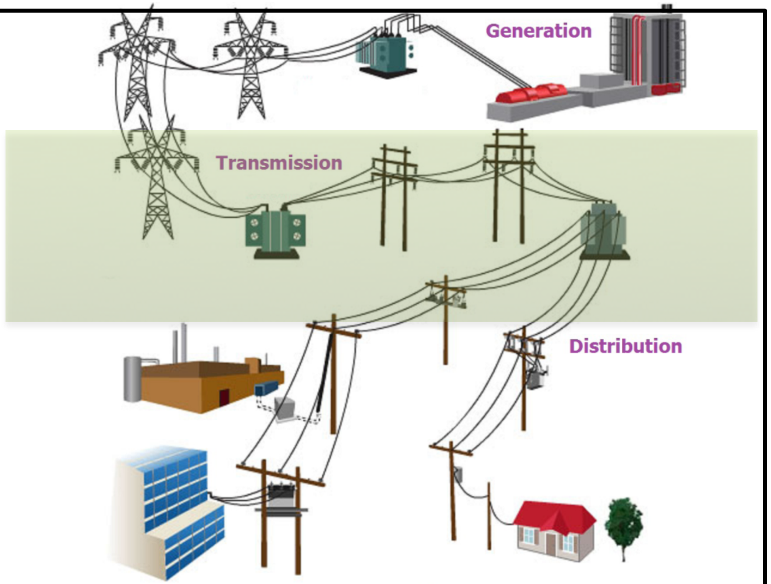
↓

Decision/Action

This talk uses the $400B power grid to illustrate our tools and techniques.

Power Systems

- **Power system:**
  - A large-scale system consisting of generators, loads, lines, etc.
  - Used for generating, transporting and distributing electricity.

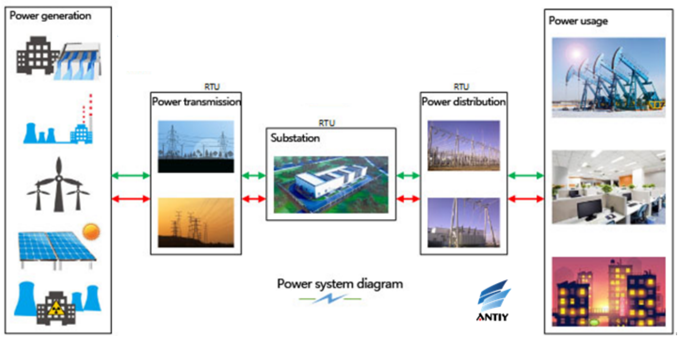


ISO, RTO, TSO

- Unit commitment (UC)
- Optimal power flow (OPF)
- Security analysis
- State estimation

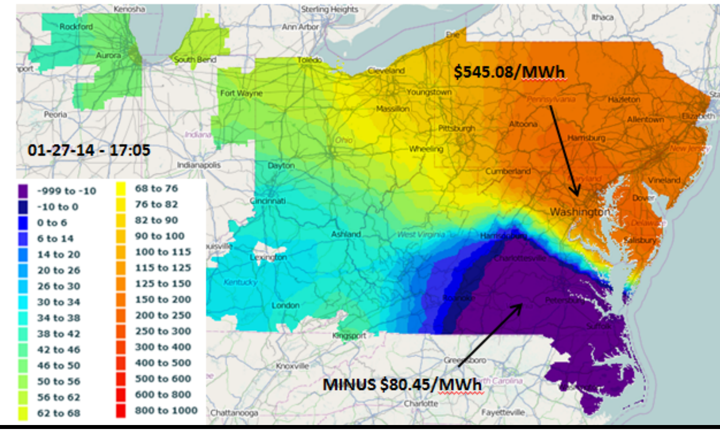
Power Operational Problems

Unit commitment: Optimize the ON/OFF status of each generator.



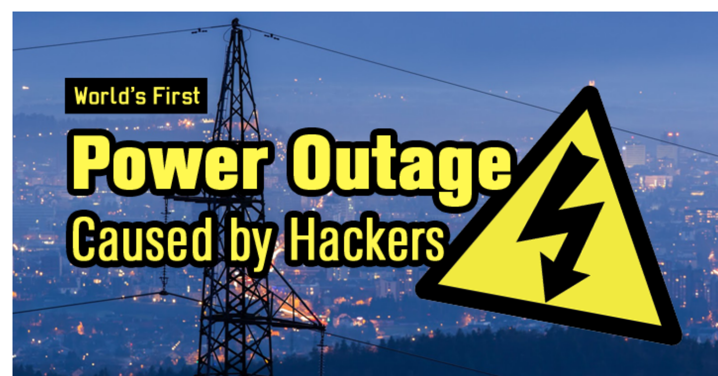
Generators Loads

Optimal power flow: Optimize the flows and parameters.



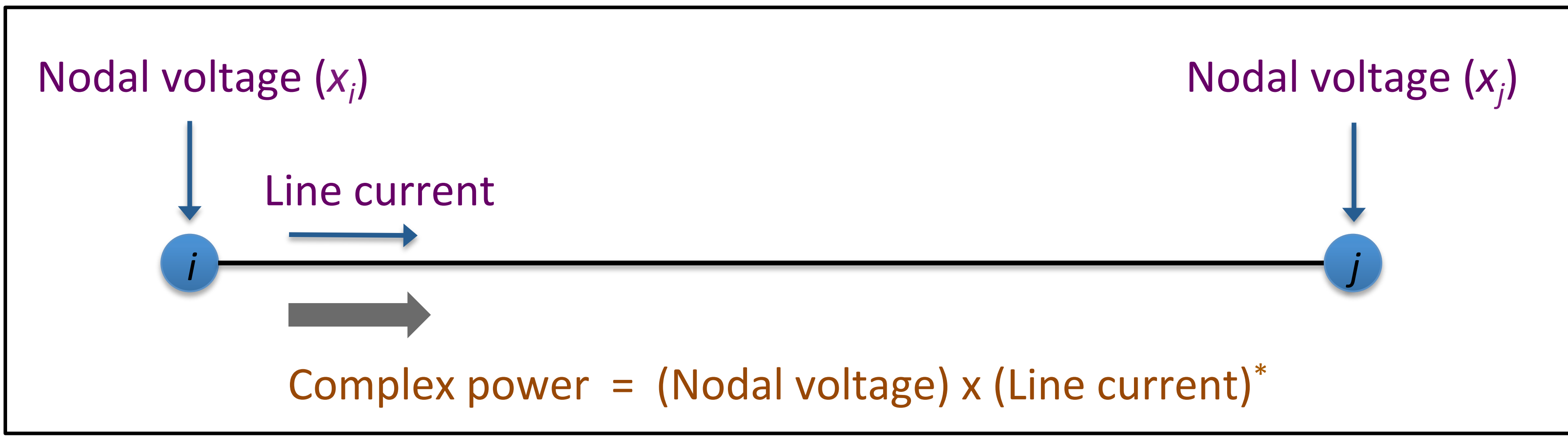
Security analysis: Guarantee robustness to faults.

State estimation: Find the state of system based on noisy measurements and bad data.



Modeling of Power Systems

- Consider two nodes *i* and *j* connected via a line:

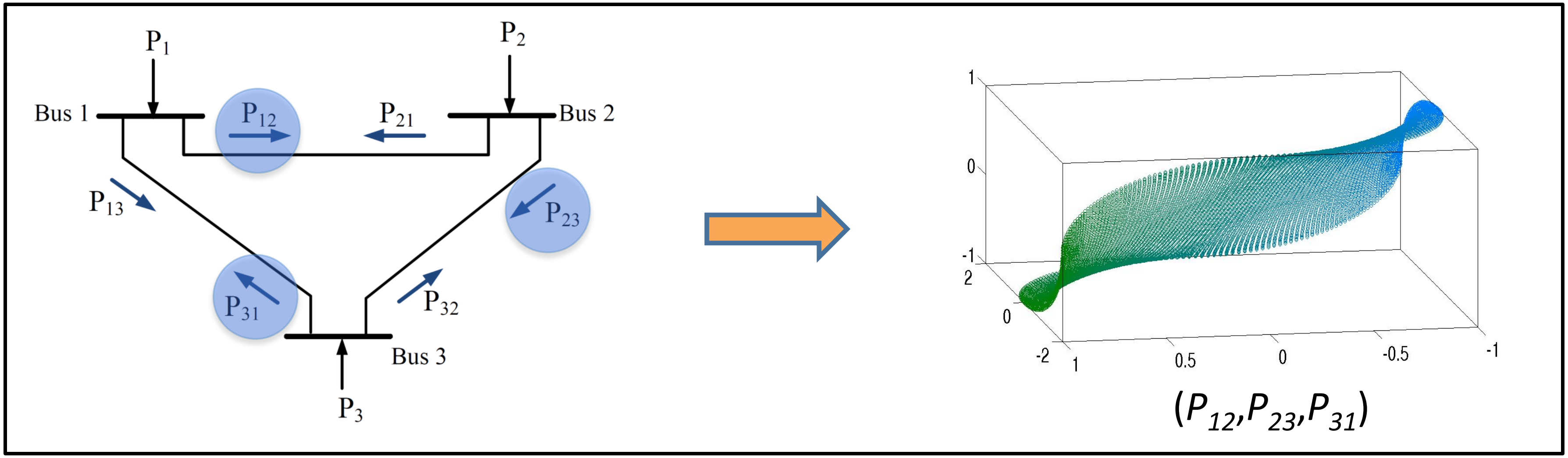


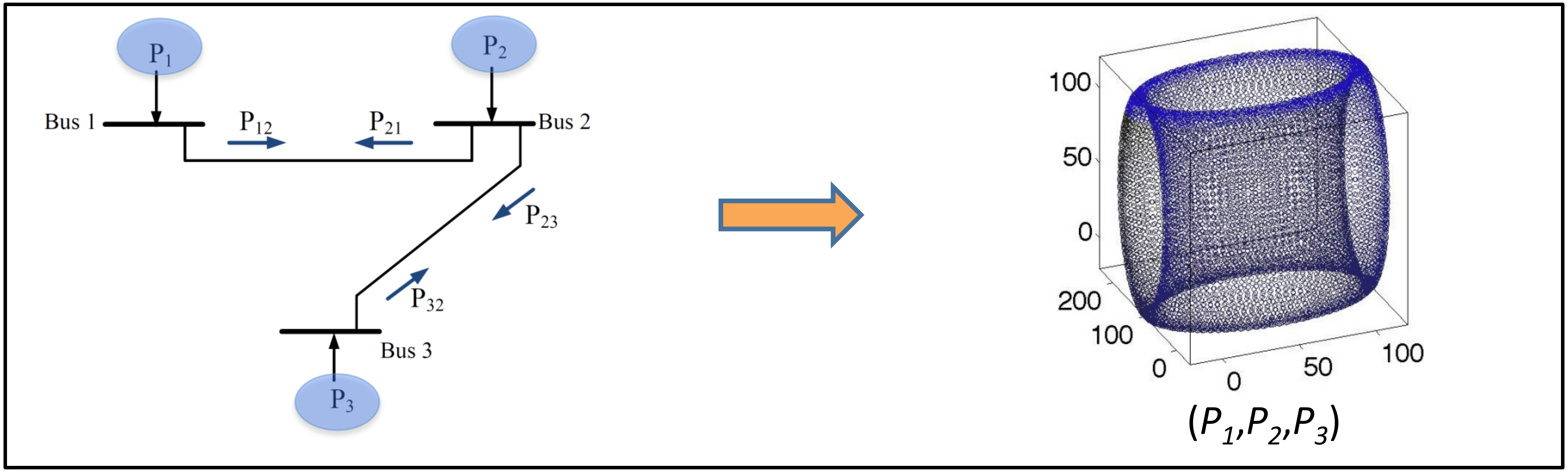
$$P_{ij} + Q_{ij}\sqrt{-1} = x_i(x_i - x_j)^* y_{ij}^*$$

Active power — Reactive power — Nonlinearity (due to laws of physics)

Nonlinearity in Power Systems

- Nonlinear laws of physics (assuming voltage magnitudes are fixed):





Power Operational Problems

Challenge 1: Nonlinearity due to laws of physics

- ❑ Complexity: Strongly NP-complete with long history since 1962.
- ❑ Common practice: Approximation
- ❑ FERC Study: Annual cost of approximation > $ 1 billion

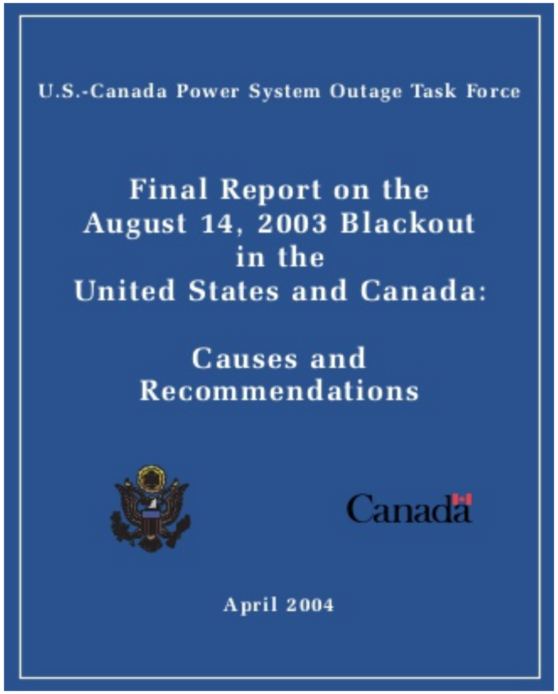
ARPA-E is setting up a $3.5M cash prize competition

Challenge 2: Nonlinearity due to discrete variables

Challenge 3: Astronomical number of security constraints

Challenge 4: Making decisions based on noisy and bad data

Blackout due to state estimation and security analysis


U.S.-Canada Power System Outage Task Force

Final Report on the August 14, 2003 Blackout in the United States and Canada:

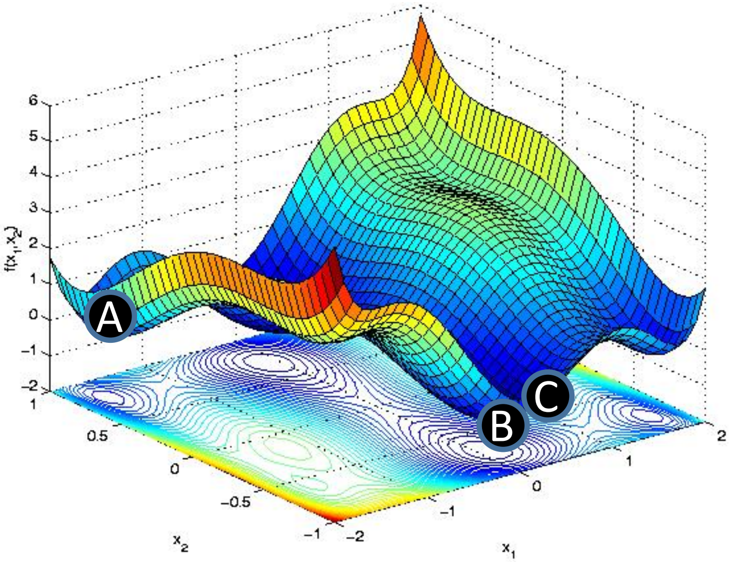
Causes and Recommendations

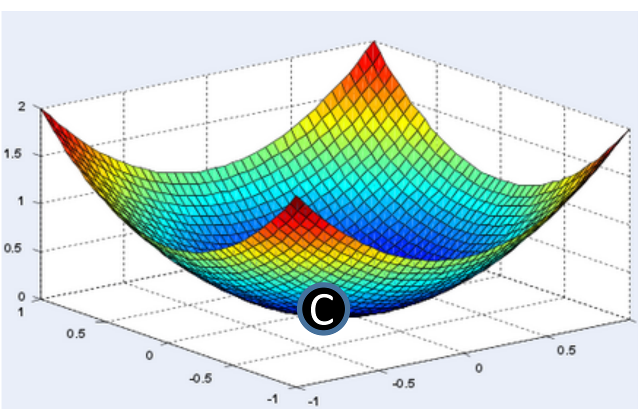
Canada

April 2004

Nonlinear Laws of physics

- Due to nonlinearity, there are different types of solutions:





Point A: Local solution	→	industry
Point B: Global solution		
Point C: Near-global solution	→	our method

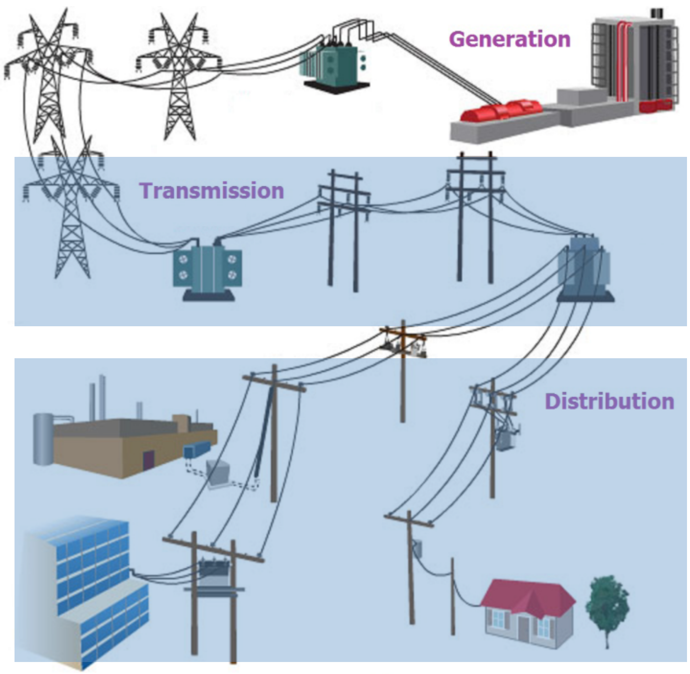
Find a near-global solution with a high optimality guarantee using convex optimization

$$\text{Optimality Guarantee} \geq \frac{\text{Global cost}}{\text{Near-global cost}} \times 100$$

A number between 0 % and 100 %

Findings for Power Systems

- To find a global solution, we proposed a method based on **SDP**.
- SDP worked for IEEE benchmark examples and several real data sets.
- For the first time, this method found and certified global minima for benchmark systems.



cyclic → **Theorem:** SDP works under positive LMPs with phase-shifting transformers.

acyclic → **Theorem:** SDP works under positive locational marginal prices (LMPs).

Physics of power networks (e.g., passivity) reduce computational complexity for power optimization problems (joint work with Steven Low and Somayeh Sojoudi)

Findings for Power Systems

- ❑ SDP may not be exact for ISOs' large-scale systems (some negative LMPs).
- ❑ To find a near-global solution, we proposed a method named **Penalized SDP**.

Case	Cost	Guarantee	Time (sec)
Polish 2383wp	1874322.65	99.316%	529
Polish 2736sp	1308270.20	99.970%	701
Polish 2737sop	777664.02	99.995%	675
Polish 2746wop	1208453.93	99.985%	801
Polish 2746wp	1632384.87	99.962%	699
Polish 3012wp	2608918.45	99.188%	814
Polish 3120sp	2160800.42	99.073%	910

We generalized to and incorporated unit commitment, state estimation, security analysis, transmission switching, and distributed control

Our research revitalized the area:

- ❖ Follow-up work in academia
- ❖ Interest from industry
- ❖ Many talks at FERC's summer workshops in 2012-17

This work significantly contributed to the initiation of the ARPA-E Grid Competition

Convexification

Arbitrary Real/Complex Polynomial Optimization

$$\min_{x \in \mathbf{C}^n} \; x^H M_0 x \;\longrightarrow\; \text{trace}\{M_0 x x^H\}$$

$$\text{s.t.} \quad x^H M_i x \le a_i, \quad i = 1, 2, \ldots, m$$

x → $W = xx^H$ → $W \succeq 0$ and ~~$\text{rank}\{W\} = 1$~~

SDP relaxation

$$\min_{W \in \mathbb{H}^n} \; \text{trace}\{M_0 W\}$$

$$\text{s.t.} \quad \text{trace}\{M_i W\} \le a_i, \quad i = 1, 2, \ldots, m$$

$$W \succeq 0$$

Rank-1 SDP → global min x

Penalized SDP

$$\min_{W} \; \text{trace}\{M_0 W\} + \lambda\, g(W)$$

$$\text{s.t.} \quad \text{trace}\{M_i W\} \le a_i, \quad i = 1, 2, \ldots, m$$

$$W \succeq 0$$

+ valid inequalities

Rank-1 SDP → near-global min x

Convexification

Arbitrary Real/Complex Polynomial Optimization

$$\min_{x \in \mathbf{C}^n} \; x^H M_0 x \longrightarrow \mathrm{trace}\{M_0 x x^H\}$$
$$\text{s.t.} \quad x^H M_i x \le a_i, \quad i = 1, 2, ..., m$$

SDP relaxation

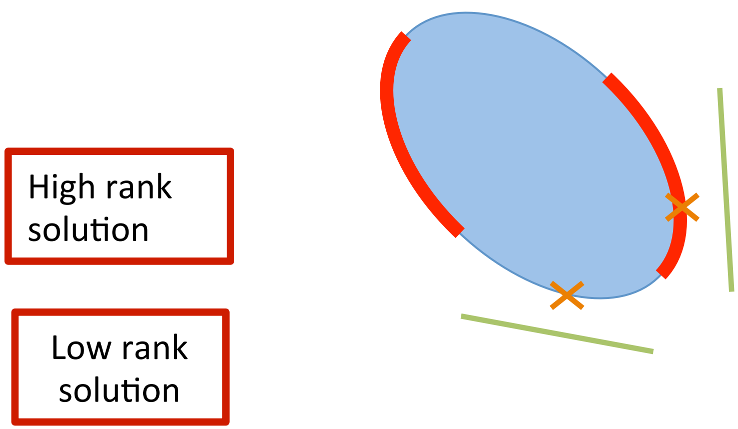
$$\min_{W \in \mathbb{H}^n} \; \mathrm{trace}\{M_0 W\}$$
$$\text{s.t.} \quad \mathrm{trace}\{M_i W\} \le a_i, \quad i = 1, 2, ..., m$$
$$W \succeq 0$$

Penalized SDP

$$\min_{W} \; \mathrm{trace}\{M_0 W\} + \lambda \, g(W)$$
$$\text{s.t.} \quad \mathrm{trace}\{M_i W\} \le a_i, \quad i = 1, 2, ..., m$$
$$W \succeq 0$$

+ valid inequalities

Geometric interpretation:



Convexification

Arbitrary Real/Complex Polynomial Optimization

$$\min_{x \in \mathbf{C}^n} \; x^H M_0 x \longrightarrow \text{trace}\{M_0 x x^H\}$$
$$\text{s.t.} \quad x^H M_i x \leq a_i, \quad i = 1, 2, ..., m$$

SDP relaxation

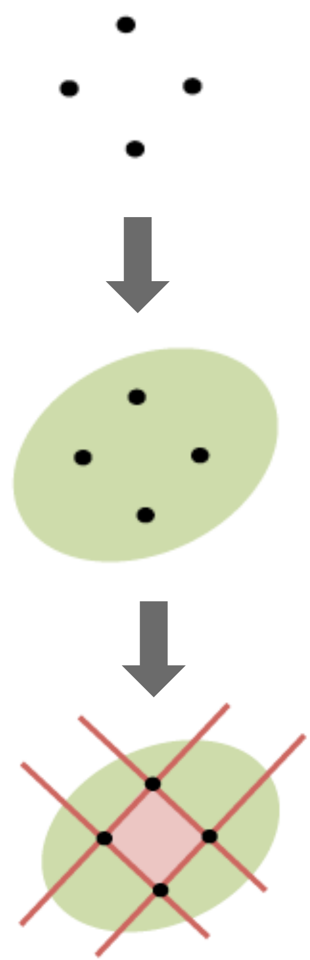
$$\min_{W \in \mathbb{H}^n} \; \text{trace}\{M_0 W\}$$
$$\text{s.t.} \quad \text{trace}\{M_i W\} \leq a_i, \quad i = 1, 2, ..., m$$
$$W \succeq 0$$

Penalized SDP

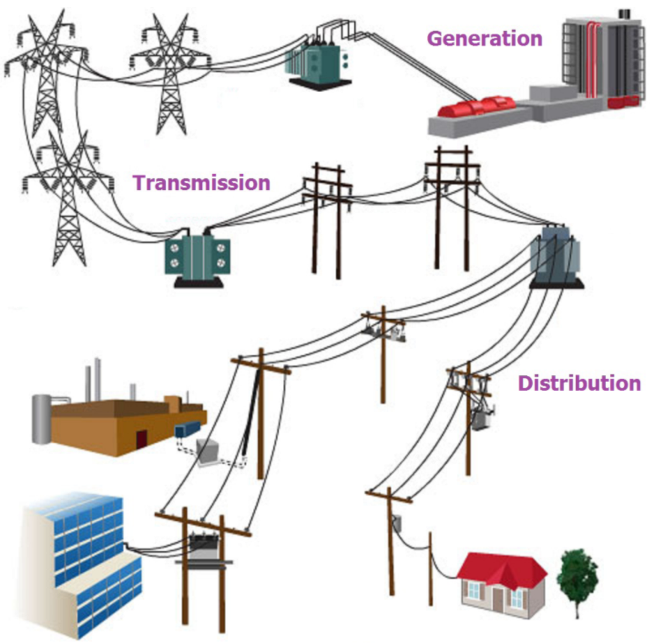
$$\min_{W} \; \text{trace}\{M_0 W\} + \lambda \, g(W)$$
$$\text{s.t.} \quad \text{trace}\{M_i W\} \leq a_i, \quad i = 1, 2, ..., m$$
$$W \succeq 0$$

+ valid inequalities

Valid Inequalities:



Outline



- ❑ Testing on several real-world systems with over 13,000 buses

Structured Optimization → Notion of generalized weighted graph

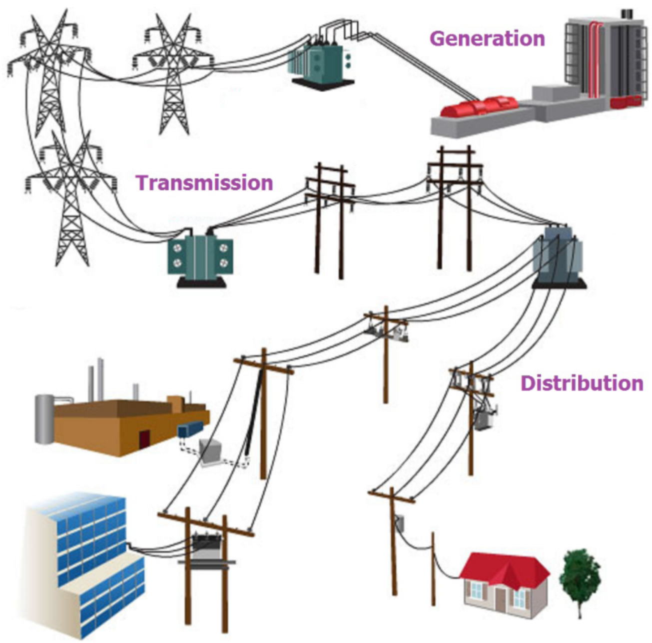
Sparse Optimization → Notions of OS and treewidth

Data Analytics → Statistical learning method

Algorithm for Sparse Optimization → Low-complexity methods

Control Theory → Optimal distributed control

Outline



- Testing on several real-world systems with over 13,000 buses

Structured Optimization →	Notion of generalized weighted graph
Sparse Optimization →	Notions of OS and treewidth
Data Analytics →	Statistical learning method
Algorithm for Sparse Optimization →	Low-complexity methods
Control Theory →	Optimal distributed control

Structured Optimization

- How does structure affect complexity?

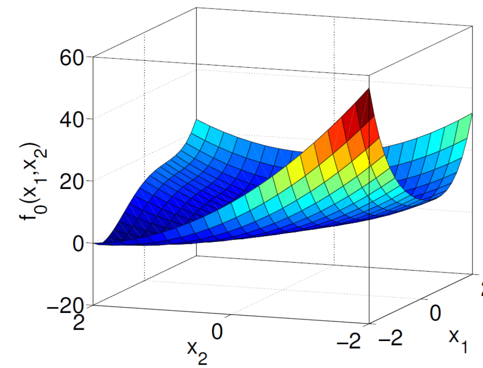
$$\min_{x_1,x_2} x_1^4 + a x_2^2 + b x_1^2 x_2 + c x_1 x_2$$

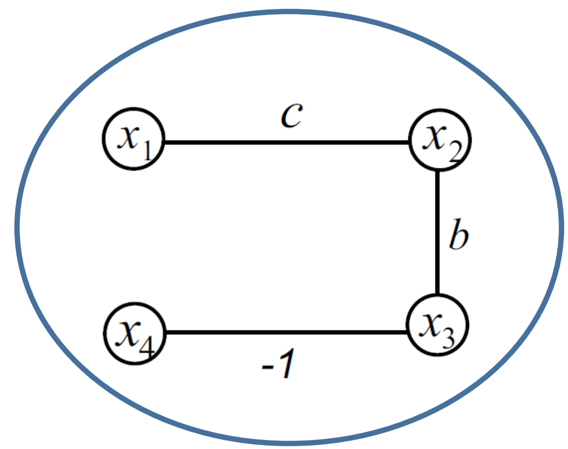
Trick: $x_1^4 = (x_1^2)^2 = x_3^2$

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{R}^4} \quad & x_3^2 + a x_2^2 + b x_3 x_2 + c x_1 x_2 \\ \text{s.t.} \quad & x_1^2 - x_3 x_4 = 0 \\ & x_4^2 - 1 = 0 \end{aligned}$$

SDP relaxation: $\mathbf{x}\mathbf{x}^* \longrightarrow W$, $x_i x_j \longrightarrow W_{ij}$

$$\begin{aligned} \min_{W\in\mathcal{S}^4} \quad & W_{33} + aW_{22} + bW_{32} + cW_{12} \\ \text{s.t.} \quad & W_{11} - W_{34} \leq 0 \\ & W_{44} - 1 = 0 \\ & W \succeq 0 \end{aligned}$$



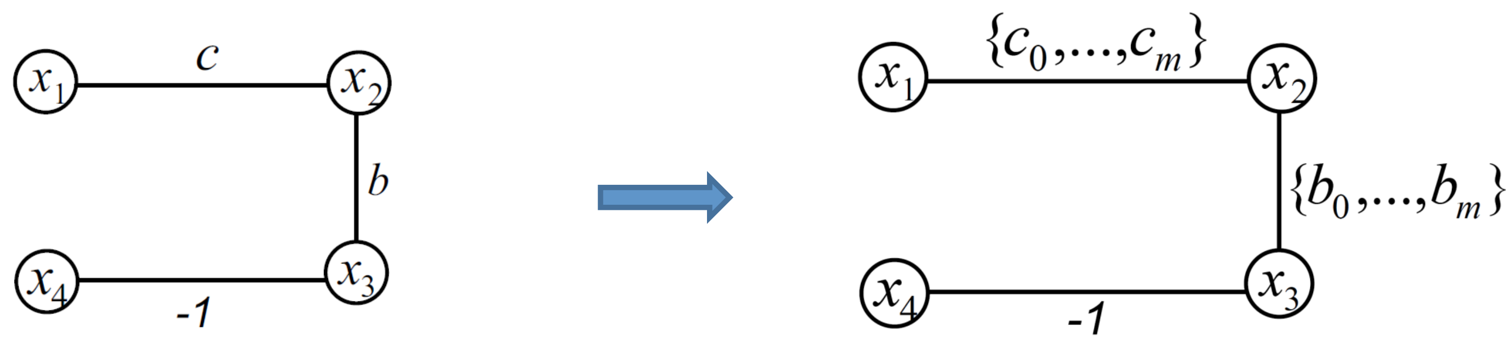


Due to structure, SDP is always exact.

Structured Optimization

$$\min_{x_1,x_2} \quad x_1^4 + a_0 x_2^2 + b_0 x_1^2 x_2 + c_0 x_1 x_2$$

$$\text{s.t.} \quad x_1^4 + a_j x_2^2 + b_j x_1^2 x_2 + c_j x_1 x_2 \le \alpha_j \quad j = 1, \ldots, m$$

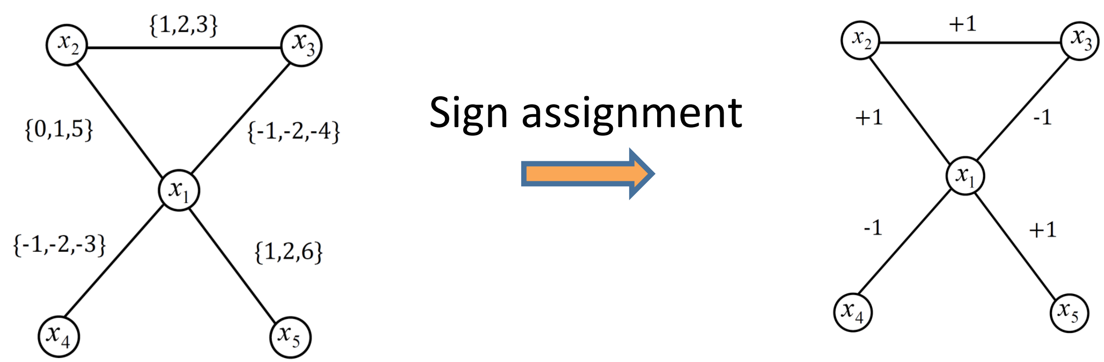


Generalized weighted graph:

- ❑ Vertices: variables
- ❑ Edges: couples
- ❑ Weight sets: coefficients

Structured Optimization

Real-valued Optimization:



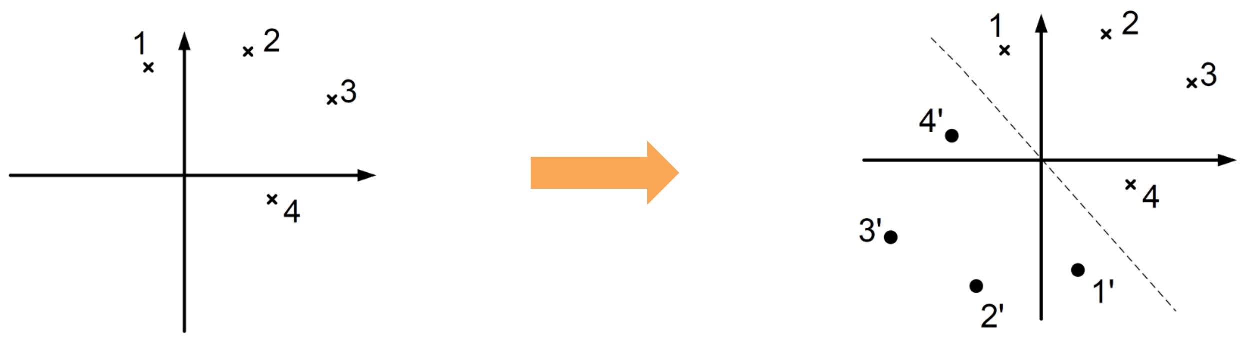
- ❑ **Local property:** the weight set of each edge is sign definite.
- ❑ **Global property:** the number of positive sets around each cycle is even.

- ❑ The proposed conditions include several existing ones ([Kim and Kojima, 2003], [Padberg, 1989], [Bose, Gayme, Chandy, and Low, 2012], etc.).

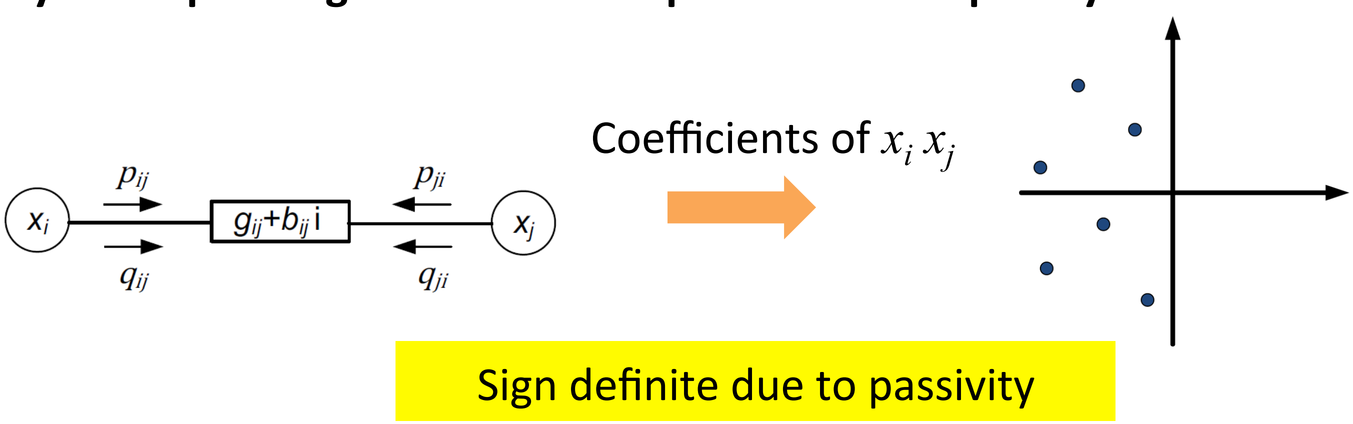
Structured Optimization

Complex-valued Optimization:

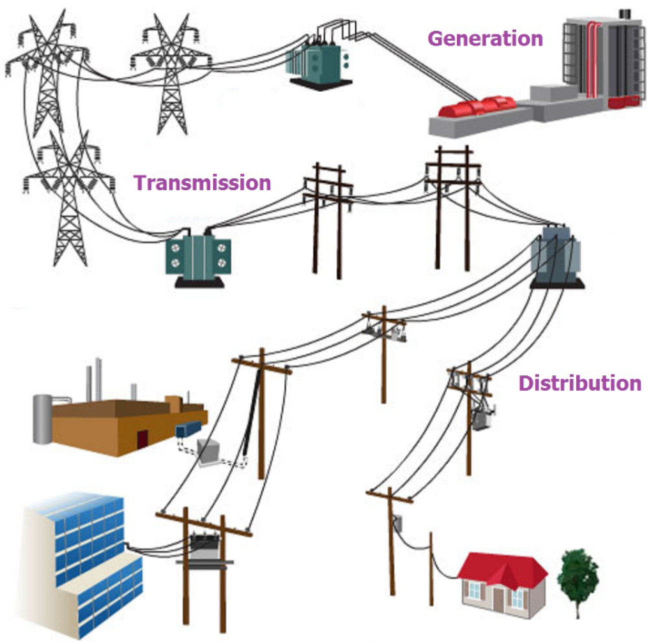
- Sign-definite set : **"T "** is sign definite if **T** and **–T** are separable in $\mathbf{R}^2$



Physics of power grids reduce computational complexity.



Outline

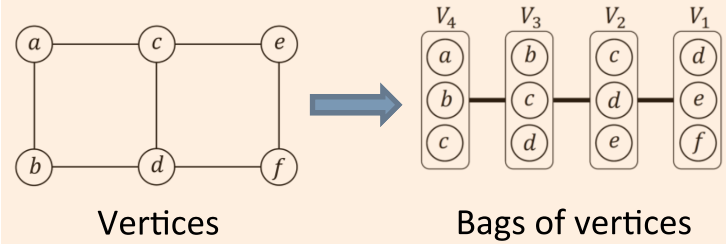


- ❑ Testing on several real-world systems with over 13,000 buses

Structured Optimization	→	Notion of generalized weighted graph
Sparse Optimization	→	Notions of OS and treewidth
Data Analytics	→	Statistical learning method
Algorithm for Sparse Optimization	→	Low-complexity methods
Control Theory	→	Optimal distributed control

Graph Notions

- **Tree decomposition:** Map the graph *G* into a tree *T:*
  - Each node of *T* is a bag of vertices of *G*
  - Each edge of *G* appears in one node of *T*
  - If a vertex shows up in multiple nodes of *T,* those nodes should form a subtree
- **Width of T:** Max cardinality minus 1

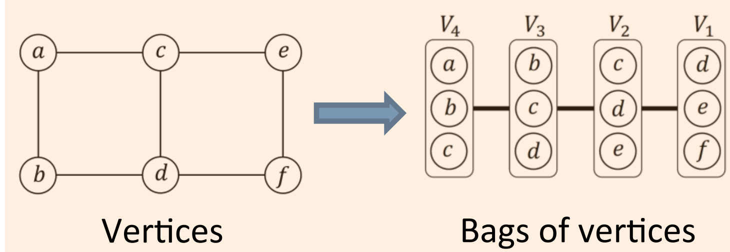


Treewidth of G: Minimum width

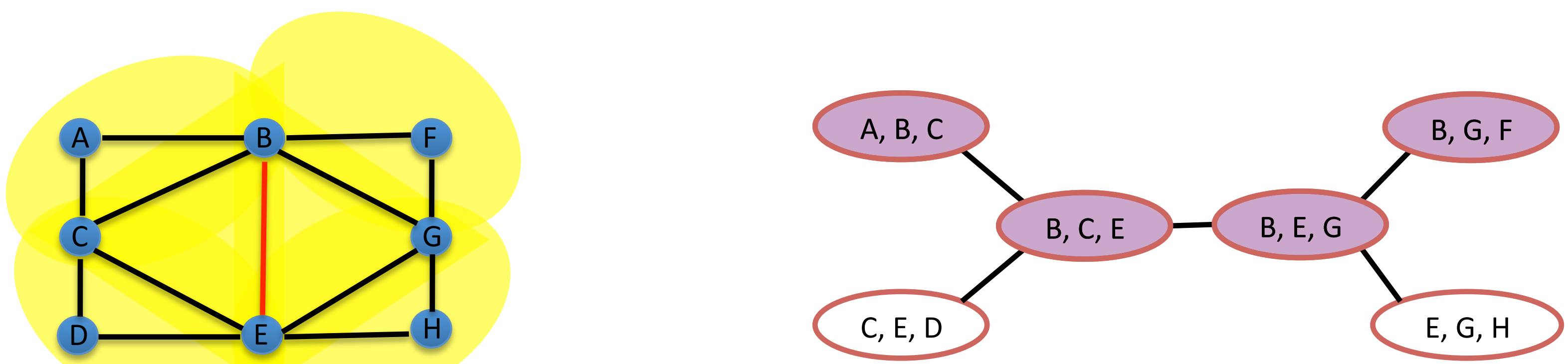
Treewidth = 1

Graph Notions

- ❑ **Tree decomposition:** Map the graph *G* into a tree *T:*
  - ❖ Each node of *T* is a bag of vertices of *G*
  - ❖ Each edge of *G* appears in one node of *T*
  - ❖ If a vertex shows up in multiple nodes of *T,* those nodes should form a subtree
- ❑ **Width of T:** Max cardinality minus 1



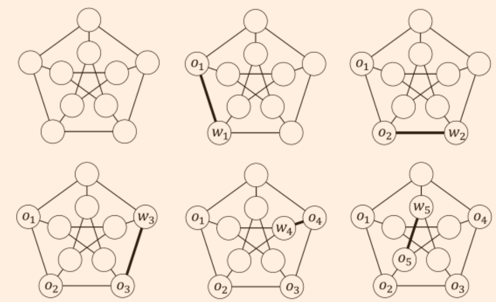
Treewidth of G: Minimum width



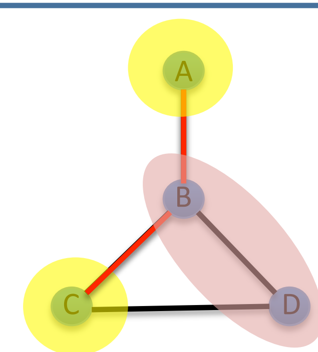
Treewidth = 2

Graph Notions

- **OS-vertex sequence:** [Hackney et al, 2009]
  - Partial ordering of vertices
  - Assume $O_1, O_2, \ldots, O_m$ is a sequence.
  - O_i has a neighbor w_i not connected to the connected component of O_i in the subgraph induced by $O_1, \ldots, O_i$



OS: Maximum cardinality among all OS sequences



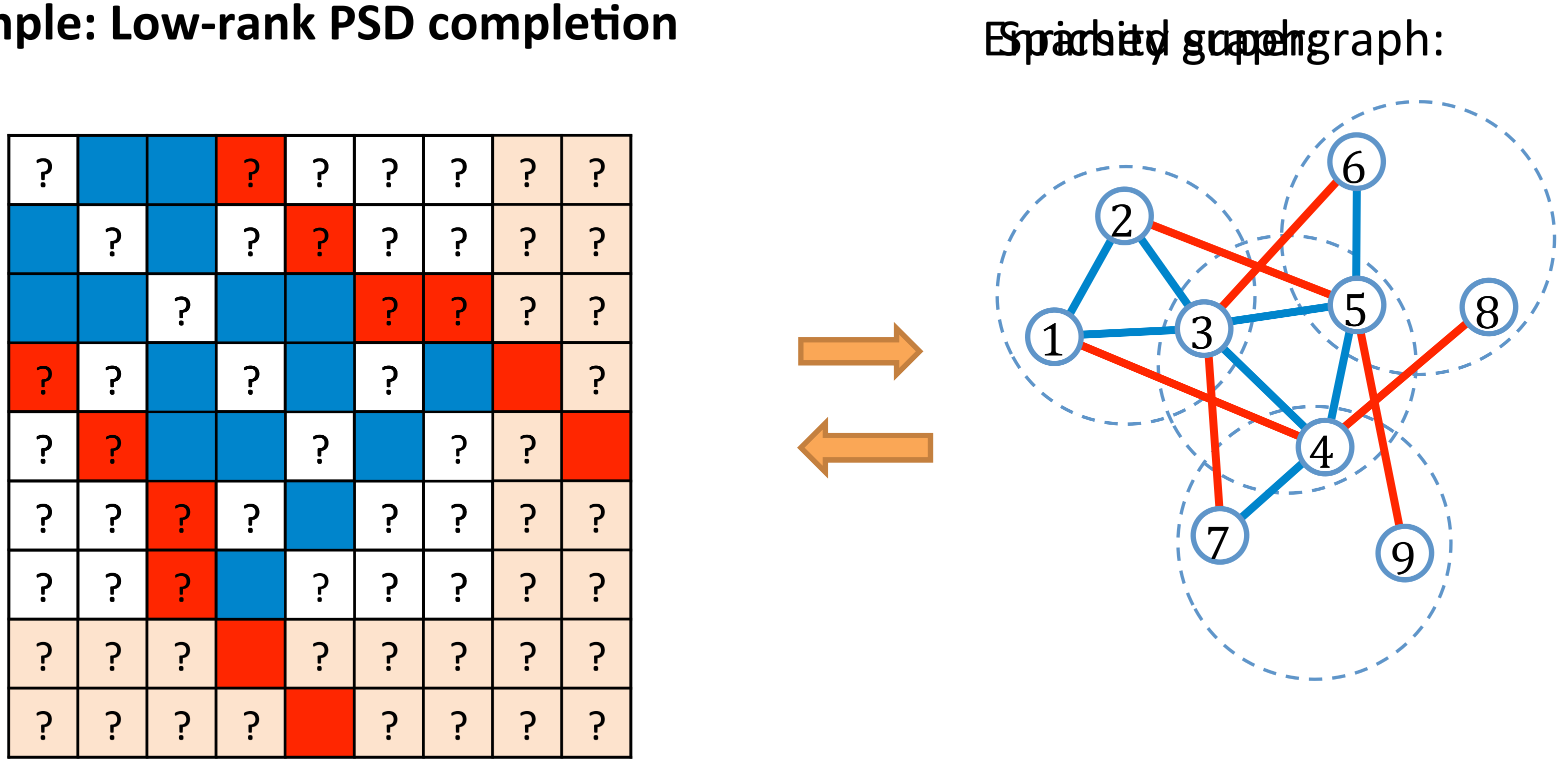
OS-vertex sequence: C ⟶ D ⟶ B

OS-vertex sequence: C ⟶ A

Low-Rank Solution

- Roughly speaking, <u>very</u> sparse graphs have high OS and low treewidth.
- We can use these notions to find low-rank solutions.

Example: Low-rank PSD completion



- Minimizing every nonzero weighted sum of the red entries gives a low-rank matrix.

Low-Rank Solution

$$\min_{x \in \mathbf{D}^n} \quad x^H M_0 x$$
$$\text{s.t.} \quad x^H M_i x \le a_i, \quad i = 1, 2, \ldots, m$$

SDP

$$\min_{W} \quad \text{trace}\{M_0 W\}$$
$$\text{s.t.} \quad \text{trace}\{M_i W\} \le a_i, \quad i = 1, 2, \ldots, m$$
$$W \succeq 0$$

→ **Sparsity Graph *G***: Generalized weighted graph

→ Often infinitely many low-rank and high-rank solutions

Embedding + penalization of red entries

$$\min_{\overline{W}} \quad \text{trace}\{\overline{M}_0 \overline{W}\} + \sum_{(j,k) \in \mathcal{G}'} \varepsilon_{j,k} \overline{W}_{jk}$$
$$\text{s.t.} \quad \text{trace}\{\overline{M}_i \overline{W}\} \le \overline{a}_i, \quad i = 1, 2, \ldots, \overline{m}$$
$$\overline{W} \succeq 0$$

Theorem: SDP has a solution such that

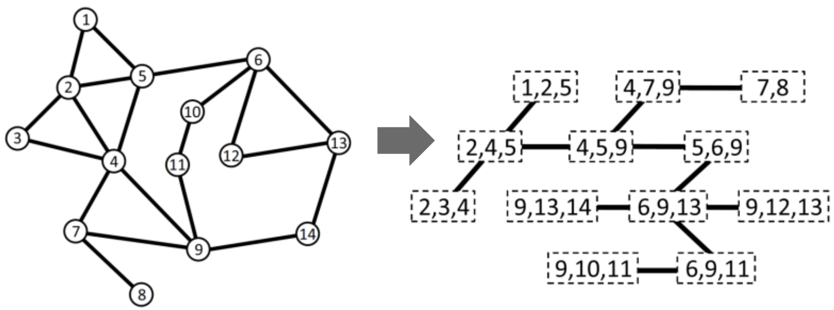
$$\text{Rank}\{W^{\text{opt}}\} \le |\mathcal{G}'| - \min_{\mathcal{G}_s} \left\{ \text{OS}(\mathcal{G}_s) \;\middle|\; (\mathcal{G}' - \mathcal{G}) \subseteq \mathcal{G}_s \subseteq \mathcal{G}' \right\}$$

= TW(*G*) +1 for *G' as enriched supergraph*

- This result includes the recent work *Laurent and Varvitsiotis*, 2012.

Treewidth in Power Systems

Tree decomposition for IEEE 14-bus system:



Case studies:

System $\mathcal{G}$	tw{$\mathcal{G}$}	System $\mathcal{G}$	Bound on tw{$\mathcal{G}$}
IEEE 14-bus	2	Polish 2383wp	23
IEEE 30-bus	3	Polish 2736sp	23
New England 39-bus	3	Polish 2746wop	23
IEEE 57-bus	5	Polish 3012wp	24
IEEE 118-bus	4	Polish 3120sp	24
IEEE 300-bus	6	Polish 3375wp	25

Treewidth of EU Grid < 35

Treewidth of NY < 40

Theorem: The rank of SDP solution is upper bounded by **Treewidth + 1.**

Complexity of solving optimization over a grid depends on its treewidth (related work by Bienstock & Munoz 2015).

Non-convexity Localization

$$\min_{x \in \mathbf{D}^n} \quad x^H M_0 x$$
$$\text{s.t.} \quad x^H M_i x \le a_i, \quad i = 1, 2, ..., m$$

Sparse → SDP works if *G* has no edges:

$$x_i^2 \quad \Longrightarrow \quad y_i \qquad \text{(LP)}$$

$$\min_{W} \quad \text{trace}\{M_0 W\}$$
$$\text{s.t.} \quad \text{trace}\{M_i W\} \le a_i, \quad i = 1, 2, ..., m$$
$$W \succeq 0$$

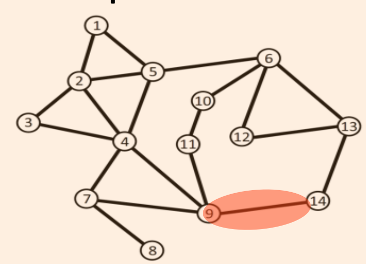
Low-rank →

- Assume SDP fails.
- **Can we identify what edges caused the failure?**
- Localized non-convexity v.s. uniform non-convexity?

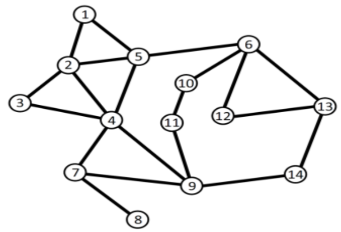
$$\min_{W} \quad \text{trace}\{M_0 W\} + \lambda \, g(W)$$
$$\text{s.t.} \quad \text{trace}\{M_i W\} \le a_i, \quad i = 1, 2, ..., m$$
$$W \succeq 0$$

Rank-1 →

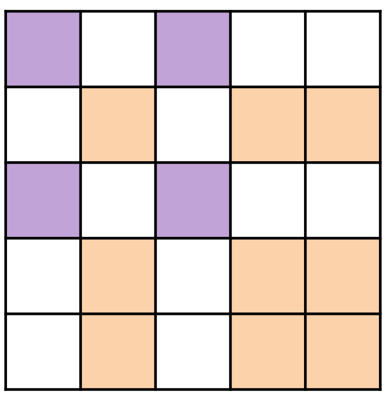
Approach for localized case:
Penalty over problematic edges



Problematic Edges



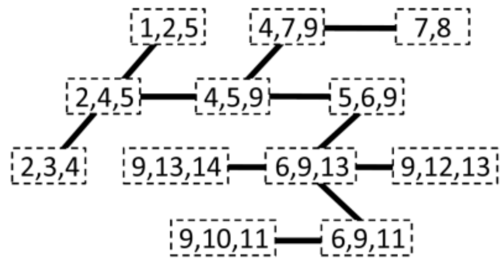
W



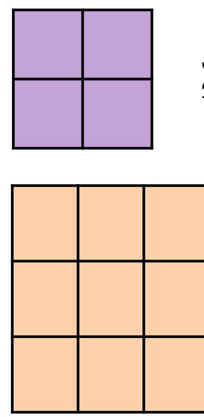
$\succeq 0$

Rank of W

=



Submatrices of W



$\succeq 0$

$\succeq 0$

Max rank of submatrices

Problematic edges: Identified based on high-rank submatrices

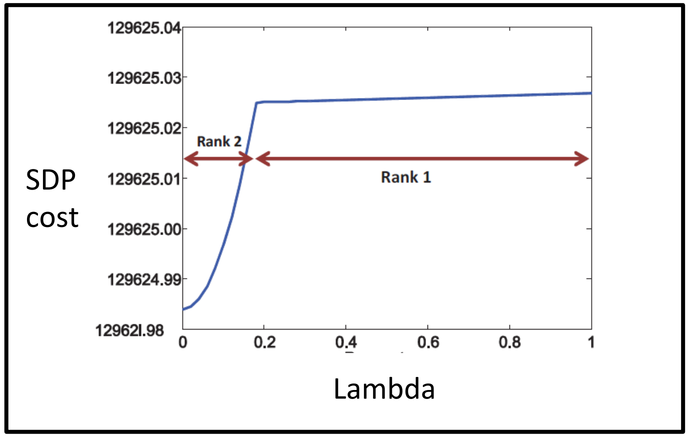
IEEE 300-bus: 2

Polish 2383-bus : 11

Near-Global Solutions

Strategy: Penalize reactive loss over problematic lines (proposed a systematic method)

- **Modified IEEE 118-bus:**
  - 3 local solutions
  - Costs: 129625, 177984, 195695



Case	TW	Cost	Guarantee	Time (sec)
Chow's 9 bus	2	5296.68	100%	≤ 5
IEEE 14 bus	2	8081.53	100%	≤ 5
IEEE 24 bus	4	63352.20	100%	≤ 5
IEEE 30 bus	3	576.89	100%	≤ 5
NE 39 bus	3	41864.40	99.994%	≤ 5
IEEE 57 bus	5	41737.78	100%	≤ 5
IEEE 118 bus	4	129660.81	99.995%	≤ 5
IEEE 300 bus	6	719725.10	99.998%	13.9
Polish 2383wp	23	1874322.65	99.316%	529
Polish 2736sp	23	1308270.20	99.970%	701
Polish 2737sop	23	777664.02	99.995%	675
Polish 2746wop	23	1208453.93	99.985%	801
Polish 2746wp	24	1632384.87	99.962 %	699
Polish 3012wp	24	2608918.45	99.188%	814
Polish 3120sp	24	2160800.42	99.073 %	910

Case	Minima	Cost	Guarantee
WB2	2	877.78	100%
WB3	2	417.25	100%
WB5	2	946.58	99.995%
WB5 Mod	3	1482.22	100%
LMBM3	5	5694.54	100%
LMBM3_50	2	5823.86	99.807%
case22loop	2	4538.80	100 %
case30loop	2	2863.06	100%
case30loop Mod	3	2861.88	100%
case39 Mod4	3	557.15	99.999%
case118 Mod1	3	129625.19	99.999%
case118 Mod2	2	85987.59	100 %
case300 Mod2	2	474643.46	99.996%

7000 simulations

Penalty Design

Why was penalty chosen as loss?

$$\min_{W} \quad \text{trace}\{M_0 W\} + \lambda\, g(W)$$
$$\text{s.t.} \quad \text{trace}\{M_i W\} \leq a_i, \quad i = 1, 2, \ldots, m$$
$$W \succeq 0$$

First try: $g(W) = \|W\|_*$

- Compressed sensing and phase retrieval
- Need *n log n* measurements for a much simpler problem [Candes and Recht].

Proposed penalty:

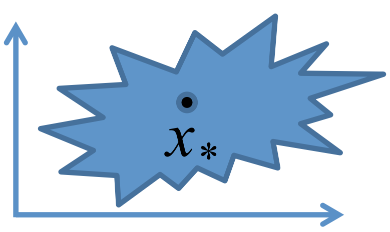
$$g(W) = \text{trace}\{MW\}$$

Algorithm design: How to find the best M?

Guess for solution of original QCQP: x_*

- $M \succeq 0$
- $M x_* = 0$
- Zero is a simple eig of M.

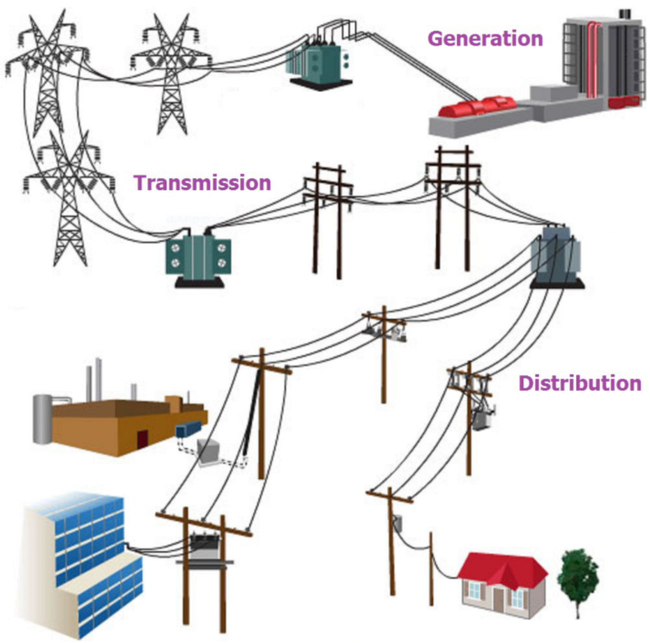
Recovery region for x **:**



$$R_M = \{x \mid g(x, M) \succeq 0\}$$

Theorem: SDP is exact if the solution is in a vicinity of x_*.

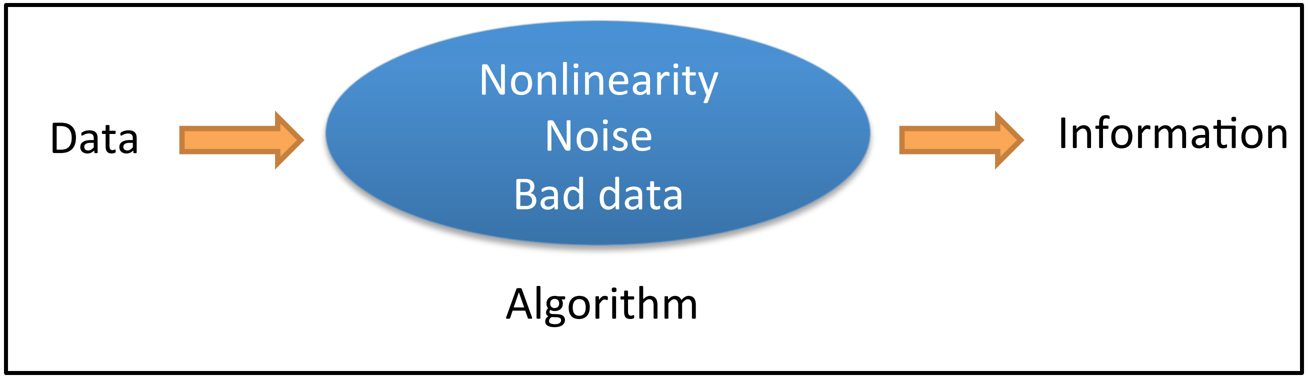
Outline



- ❑ Testing on several real-world systems with over 13,000 buses

Structured Optimization	→	Notion of generalized weighted graph
Sparse Optimization	→	Notions of OS and treewidth
Data Analytics	→	Statistical learning method
Algorithm for Sparse Optimization	→	Low-complexity methods
Control Theory	→	Optimal distributed control

Data Analytics



$$x_r = \langle \mathbf{v}\mathbf{v}^*, \mathbf{M}_r \rangle + \omega_r + \eta_r, \qquad r \in \mathcal{M}$$

measurement | unknown state | noise | bad data

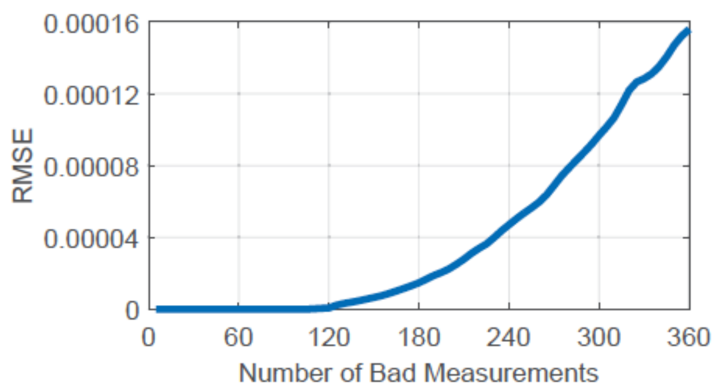
❑ **Penalized SDP:** Two-term objective to handle non-convexity and noise estimation

$$\begin{aligned} \underset{\substack{\mathbf{W} \in \mathbb{H}^n \\ \boldsymbol{\nu} \in \mathbb{R}^m}}{\text{minimize}} \quad & \langle \mathbf{W}, \mathbf{M} \rangle + \mu \times \|\boldsymbol{\nu}\|_1 \\ \text{subject to} \quad & \langle \mathbf{W}, \mathbf{M}_r \rangle + \nu_r = x_r, \qquad r \in \mathcal{M}, \\ & \mathbf{W} \succeq 0. \end{aligned}$$

State Estimation

Theorem: For carefully designed **M** and μ, if the number of bad data measurements is not too high, we have

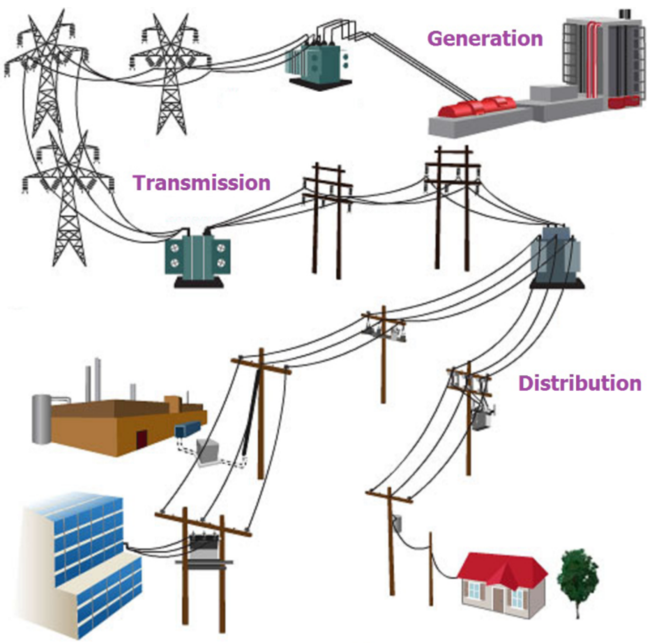
$$\|\mathbf{W}^{\text{opt}} - \alpha \mathbf{v}\mathbf{v}^*\|_F \leq \sqrt{\tau \times \text{trace}\{\mathbf{W}^{\text{opt}}\} \times \|\omega\|_1}$$



PEGASE 1354-bus system
(bad data with no noise)

- The above framework allows studying and mitigating the worst attacks possible on power grids.

Outline

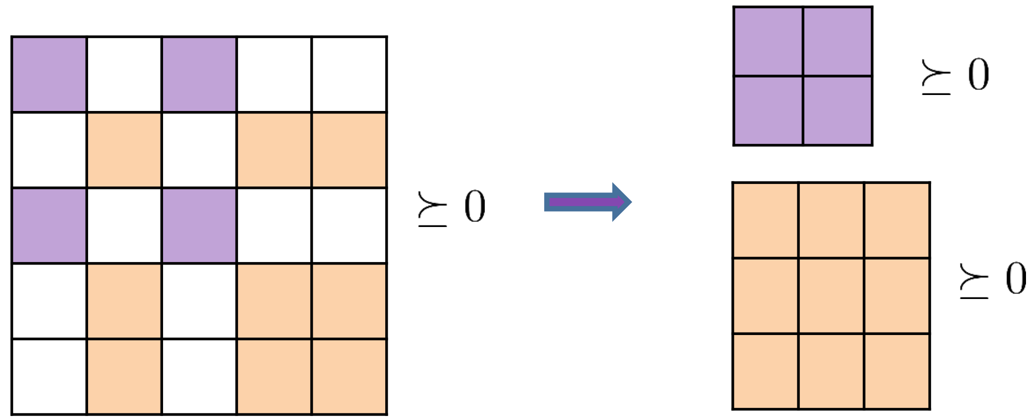


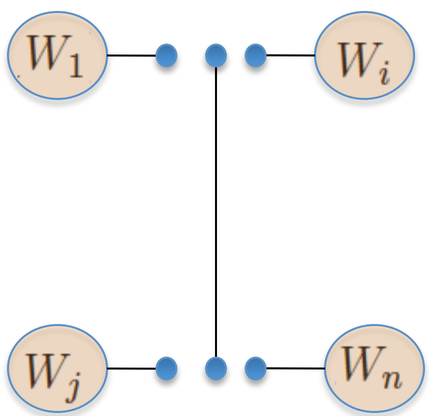
- Testing on several real-world systems with over 13,000 buses

Structured Optimization	→	Notion of generalized weighted graph
Sparse Optimization	→	Notions of OS and treewidth
Data Analytics	→	Statistical learning method
Algorithm for Sparse Optimization	→	Low-complexity methods
Control Theory	→	Optimal distributed control

Low-Complexity Distributed Computation

Goal: Design a low-complex algorithm for sparse conic optimization (LP/QP/QCQP/SOCP/SDP)





$$
\begin{aligned}
\text{minimize} \quad & \sum_{i\in\mathcal{V}} \mathbf{tr}(\mathbf{A}_i\mathbf{W}_i) && && \text{Sum of agents' objectives}\\
\text{subject to:} \quad & \mathbf{tr}(\mathbf{B}_j^i\mathbf{W}_i) = b_j^i && \forall\, j=1,\ldots,p_i \ \text{ and } \ i\in\mathcal{V} \\
& \mathbf{tr}(\mathbf{C}_j^i\mathbf{W}_i) \le c_j^i && \forall\, j=1,\ldots,q_i \ \text{ and } \ i\in\mathcal{V} && \text{Local constraints}\\
& \mathbf{W}_i \succeq 0 && \forall\, i\in\mathcal{V} \\
& \sum_{j\in N[i]} \mathbf{tr}(\mathbf{D}_k^{i,j}\mathbf{W}_j) = d_k^{(i)} && \forall\, k=1,\ldots,r_i \ \text{ and } \ i\in\mathcal{V} \\
& \sum_{j\in N[i]} \mathbf{tr}(\mathbf{E}_k^{i,j}\mathbf{W}_j) \le e_k^{(i)} && \forall\, k=1,\ldots,s_i \ \text{ and } \ i\in\mathcal{V} && \text{Overlapping constraints}\\
& \mathbf{W}_i(I_{i,j}, I_{i,j}) = \mathbf{W}_j(I_{j,i}, I_{j,i}) && \forall\, (i,j)\in\mathcal{E}^+
\end{aligned}
$$

Low-Complexity Distributed Computation

- **Distributed Algorithms for Big Data:** ADMM-based dual decomposed SDP (related work: [Parikh and Boyd, 2014], [Wen, Goldfarb and Yin, 2010], [Andersen, Vandenberghe and Dahl, 2010]).

Algorithm for Conic Optimization:

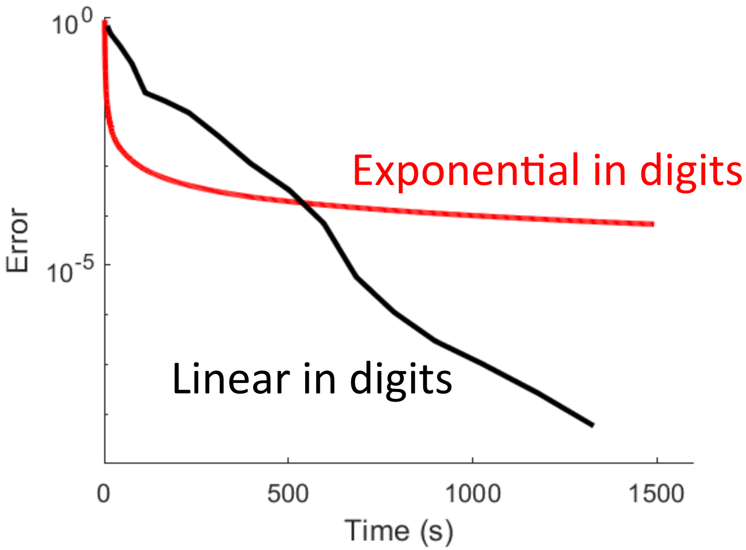
- Based on over-relaxed ADMM
- Has a guaranteed convergence
- Communications between agents
- Basic operations and eigenvalue decomposition.

- Code written in C++
- Tested on Amazon EC2 (36 cores, 60 GB RAM)
- 8000 agents with 40x40 local matrices
- Full-scale SDP: **57.6 billion variables**
- Decomposed-SDP: **12.8 million variables**
- MOSEK, SeDumi, SDPT3: **take months to solve**

	$p_i = 5, q_i = 0$	$p_i = 0, q_i = 5$	$p_i = 5, q_i = 5$
P_{obj}	3.939822e+06	6.475070e+06	9.458764e+06
D_{obj}	3.939368e+06	6.475035e+06	9.458743e+06
iter	325	1264	2810
t_{CPU} (min)	2.218	7.973	19.539
t_{iter} (sec per iter)	0.410	0.378	0.417
Optimality	99.98%	99.9994%	99.9997%

Low-Complexity Second-order Methods

First-Order versus Interior-Point



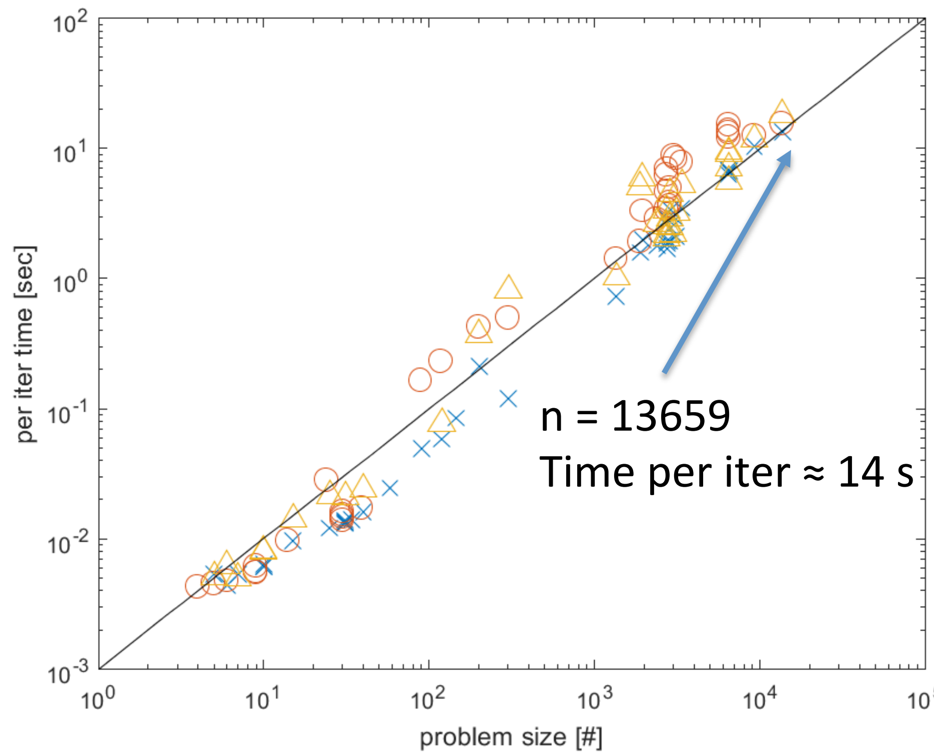
FOM: O(n) time × O(1/ε) iters

IPM: $O(n^6)$ time × O(log(1/ε)) iters

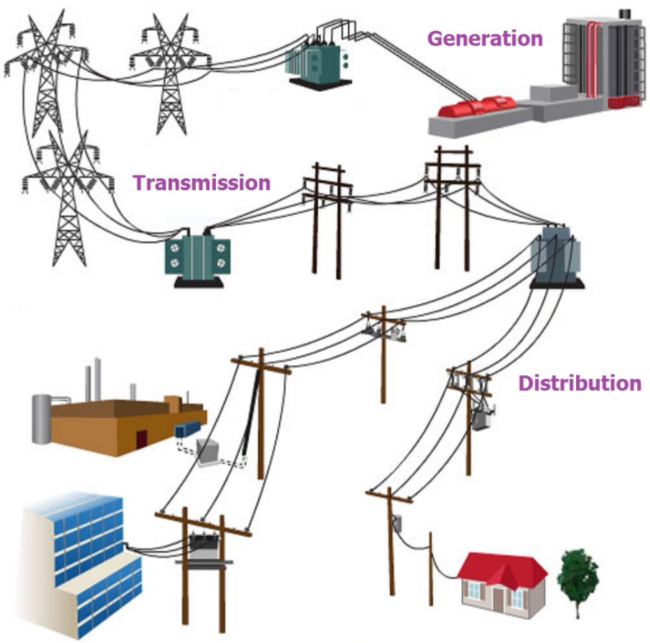
Sparsity-Exploiting IPM:
O(n) time × O(log(1/ε)) iters

(We also developed a rank-exploiting IPM.)

- ❑ Consider 40 power systems.
- ❑ Solve three SDP problems for each system:
  - ➢ MAX 3-CUT (o)
  - ➢ Lovasz Theta (x)
  - ➢ MAX BISECTION (Δ)



Outline

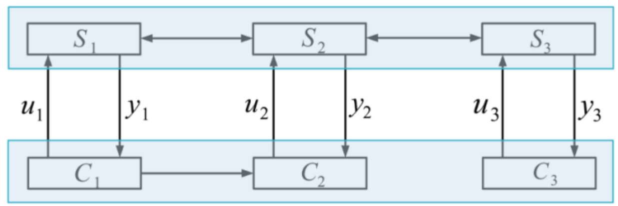


- Testing on several real-world systems with over 13,000 buses

Structured Optimization	→	Notion of generalized weighted graph
Sparse Optimization	→	Notions of OS and treewidth
Data Analytics	→	Statistical learning method
Algorithm for Sparse Optimization	→	Low-complexity methods
Control Theory	→	Optimal distributed control

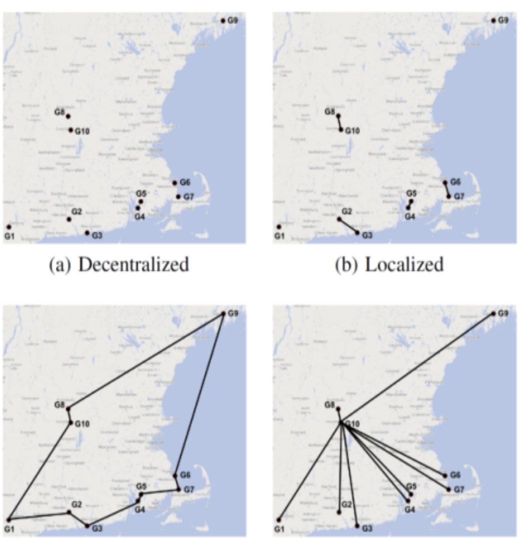
Distributed Control

Optimization for differential equations: Optimal control, dynamic programming, system ID, robust control, etc.



Distributed control

(NP-hard: Witsenhausen's example)

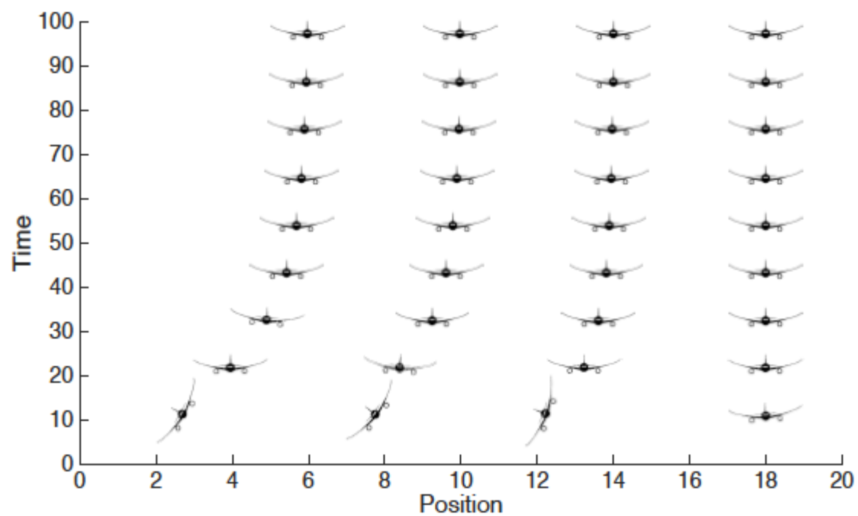


(a) Decentralized

(b) Localized

(c) Ring

(d) Star Topology (G_{10} in center)



Distributed Control

Result 1: Design based on penalized SDP (rank at most 3).

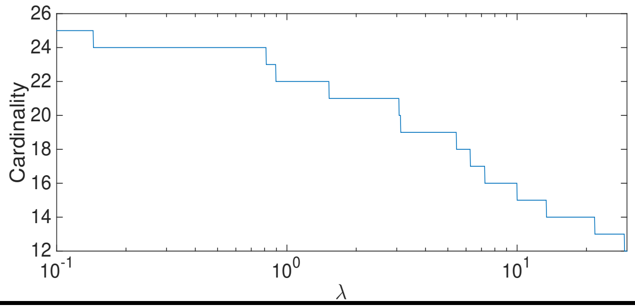
Result 2: Design based on a closed-form formula:

$$\left(1+\mu_s\sqrt{C_1^s(K_d)}\right)^2 J(K_c) \geq J(K_d) \geq J(K_c)$$

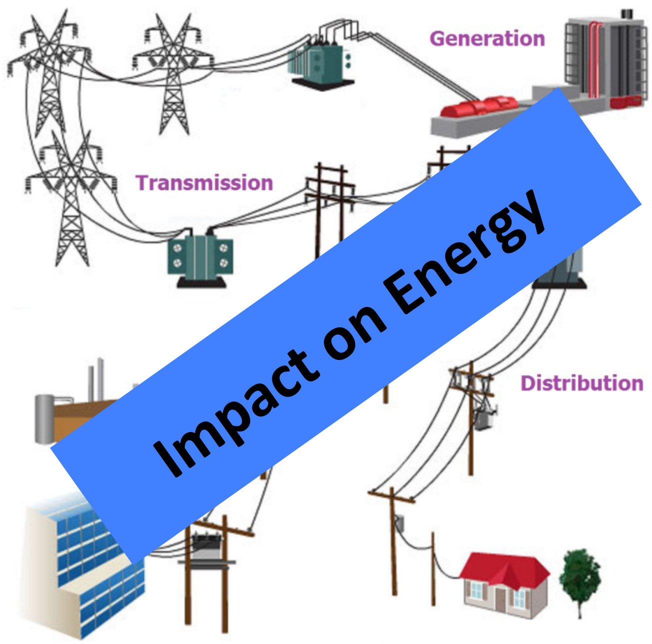
$$\mu_s = \max\left\{\frac{\kappa(V)\|B\|_2}{(1-\rho(A+BK_d))\sqrt{\mathcal{E}\{\|x_c[\infty]\|_2^2\}}},\ \frac{1-\rho(A+BK_d)+\kappa(V)\|K_d\|_2\|B\|_2}{(1-\rho(A+BK_d))\sqrt{\mathcal{E}\{\|u_c[\infty]\|_2^2\}}}\right\}$$

Result 3: Design a controller with *k* communication links:

$$\min_{K\in\mathcal{S}} \quad J(K)+\lambda||K||_1$$



Conclusions



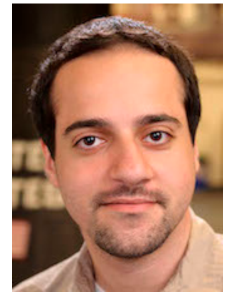
Optimization is hard in the worst case, but real-world problems may not be too hard.

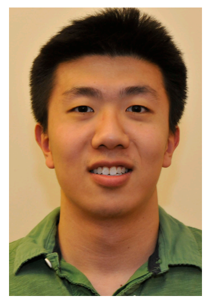
- Structured Optimization → Notion of generalized weighted graph
- Sparse Optimization → Notions of OS and treewidth
- Data Analytics → Statistical learning method
- Algorithm for Sparse Optimization → Low-complexity methods
- Control Theory → Optimal distributed control

❑ Testing on several real-world systems with over 13,000 buses

Former and Current Group Members

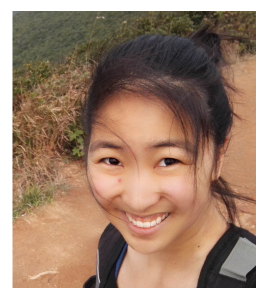
- Ramtin Madani
- Abdulrahman Kalbat
- Salar Fattahi
- Richard Zhang
- Yu Zhang
- Ming Jin
- Ghazal Fazelnia
- Morteza Ashraphjuo
- SangWoo Park
- Victoria Chang
- Han Feng







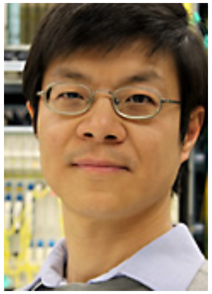



Incomplete List of Collaborators

Caltech and UT Austin:

- John Doyle
- Richard Murray
- Steven Low
- Ross Baldick






Stanford, Concordia, Washington:

- Stephen Boyd
- David Tse
- Amir Aghdam
- Baosen Zhang




UC Berkeley:

- Somayeh Sojoudi
- Shmuel Oren
- Alper Atamturk
- Murat Arcak




Funding Acknowledgements

DARPA

Office of Naval Research
ONR
Science & Technology

Air Force Office of Scientific Research
United States Air Force

NSF

Google

Siebel Energy Institute

TBSI
Tsinghua-Berkeley Shenzhen Institute