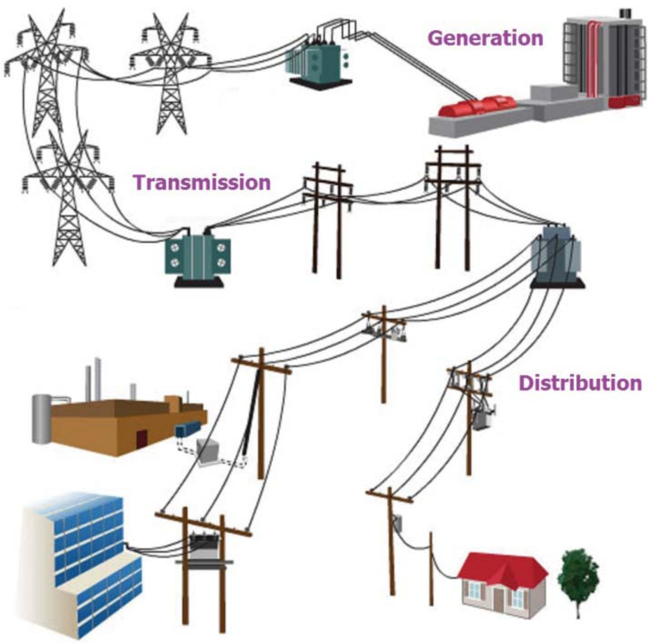
Computational Techniques for Nonlinear Optimization and Learning Problems

Javad Lavaei

Industrial Engineering and Operations Research
University of California, Berkeley

Outline

Background on Power Systems



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

Convexification via Conic Optimization for Sparse Problems

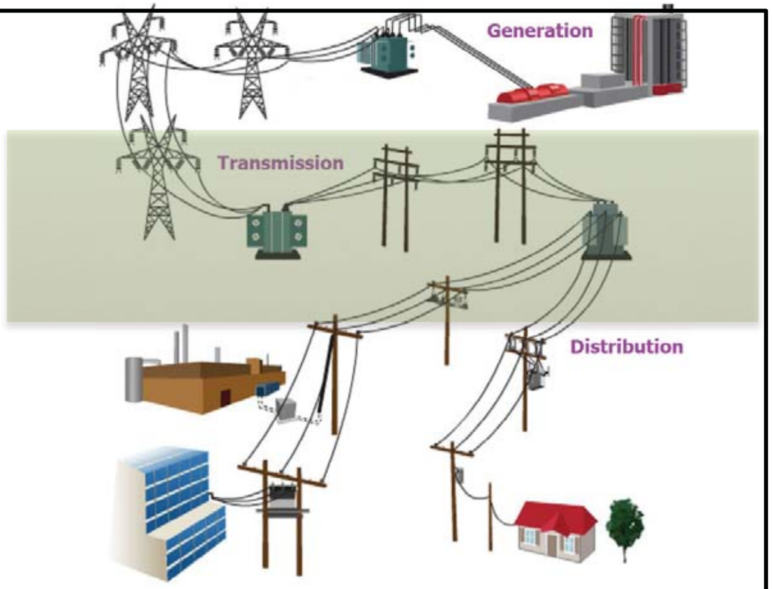
Nonlinear Data Analytics via Conic Optimization

Near-Linear time Algorithm for Conic Optimization

Online Non-convex Optimization

Power Systems

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

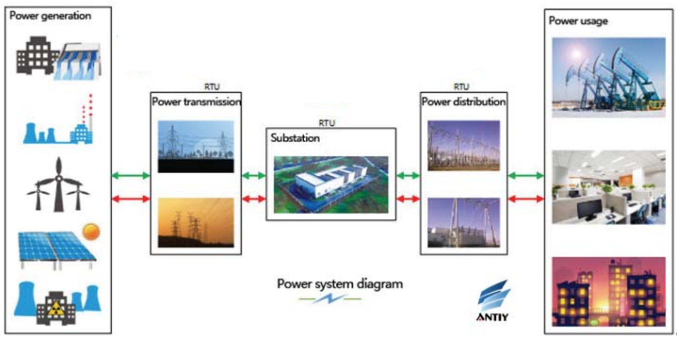


ISO, RTO, TSO

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

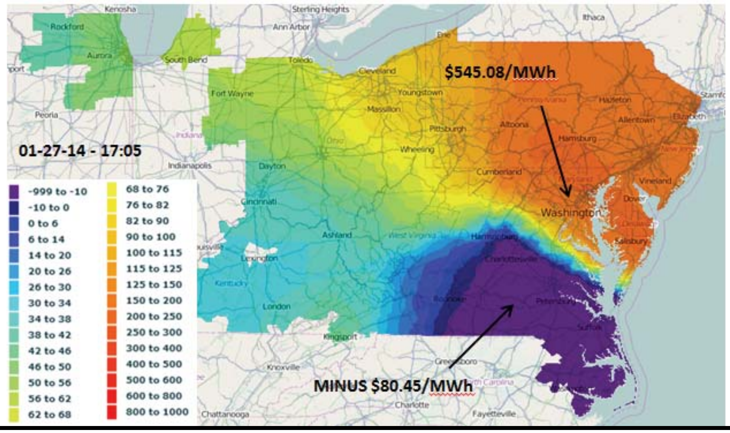
Power Operational Problems

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



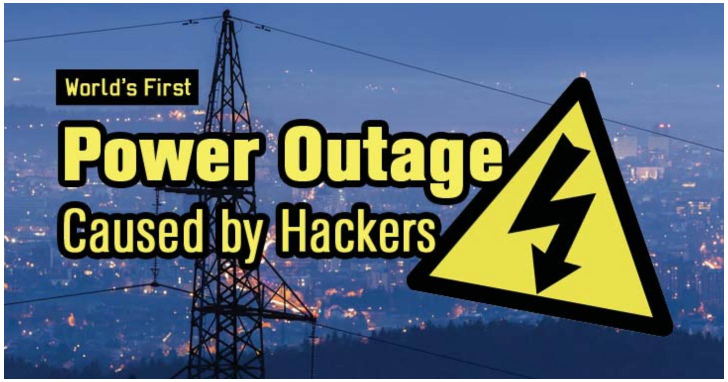
Generators Loads

Optimal power flow: Optimize the flows and parameters.



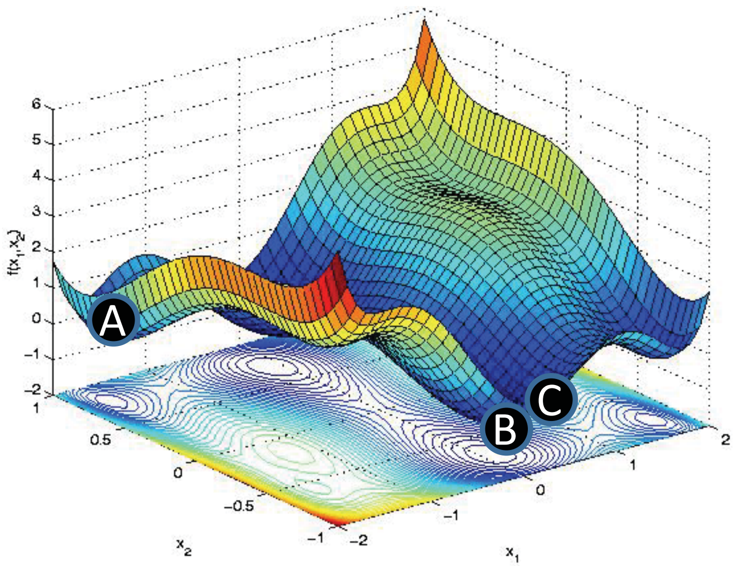
Security analysis: Guarantee robustness to faults.

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



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 goal

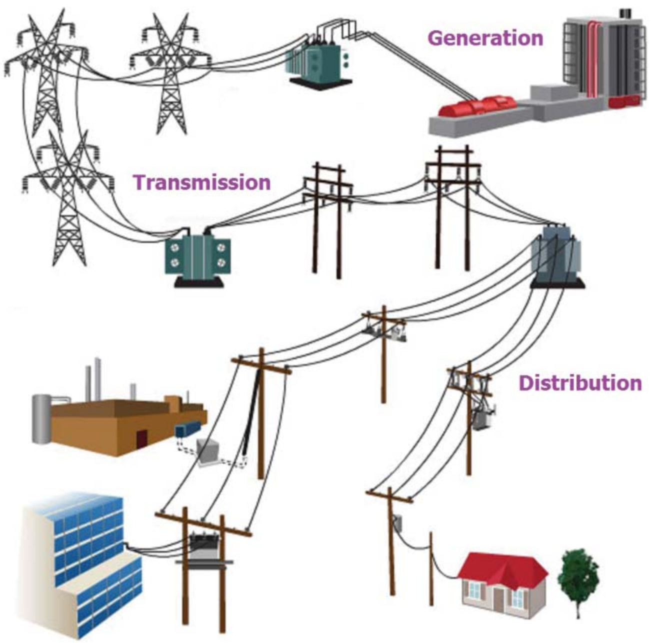
Find a global or near-global solution with a high optimality guarantee using convex optimization or local search algorithm

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

A number between 0 % and 100 %

Outline

Background on Power Systems



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

Convexification via Conic Optimization for Sparse Problems

Nonlinear Data Analytics via Conic Optimization

Near-Linear time Algorithm for Conic Optimization

Online Non-convex Optimization

Convexification

Arbitrary Real/Complex Polynomial Optimization

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

$$x \rightarrow \mathrm{W} = x x^H \rightarrow \mathbf{W} \succeq 0 \quad \text{and} \quad \cancel{\operatorname{rank}\{\mathbf{W}\} = 1}$$

SDP relaxation

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

Rank-1 SDP $\longrightarrow$ global min x

Penalized SDP

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

+ valid inequalities

Rank-1 SDP $\longrightarrow$ near-global min x

Convexification

Arbitrary Real/Complex Polynomial Optimization

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

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

SDP relaxation

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

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

$$W \succeq 0$$

Penalized SDP

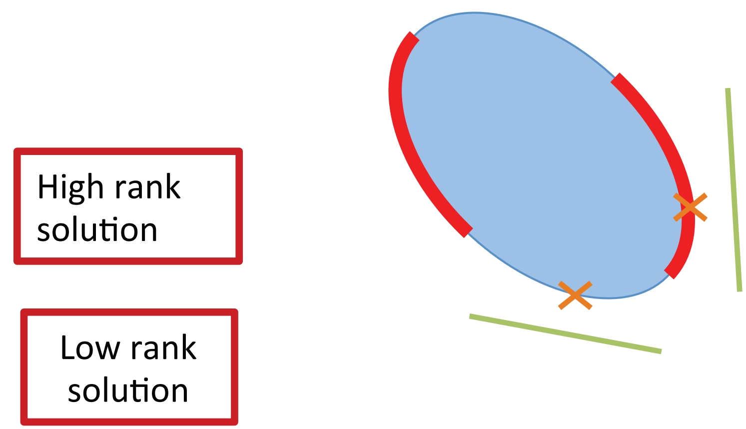
$$\min_{W} \; \operatorname{trace}\{M_0 W\} + \lambda\, g(W)$$

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

$$W \succeq 0$$

+ valid inequalities

Geometric interpretation:



Convexification

Arbitrary Real/Complex Polynomial Optimization

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

SDP relaxation

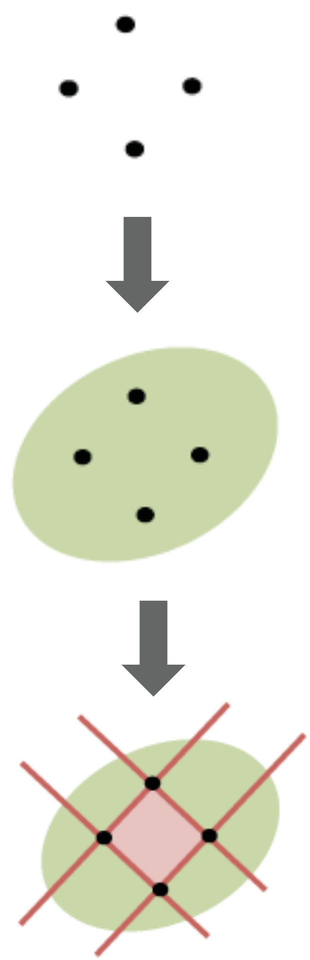
$$\min_{W\in\mathbb{H}^n} \; \mathrm{trace}\{M_0 W\}$$
$$\text{s.t.} \quad \mathrm{trace}\{M_i W\} \le a_i, \quad i = 1, 2, \ldots, m$$
$$W \succeq 0$$

Penalized SDP

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

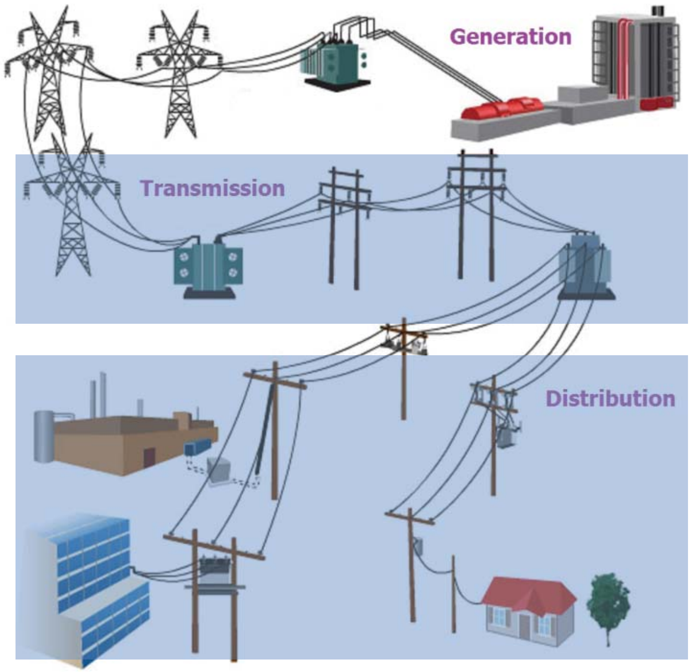
+ valid inequalities

Valid Inequalities:



Findings for Power Systems

- ❑ To find a global solution, we used **SDP**.
- ❑ **Theorem** (Lavaei & Low 2012): SDP worked for several real data sets.
- ❑ This was the first method that certified global minima for benchmark systems.



cyclic

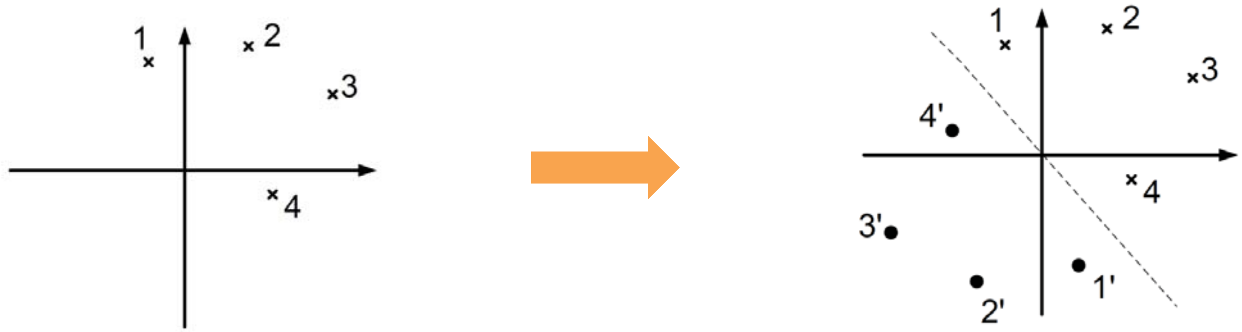
Theorem (Sojoudi & Lavaei 2012): SDP works under positive LMPs with phase-shifting transformers.

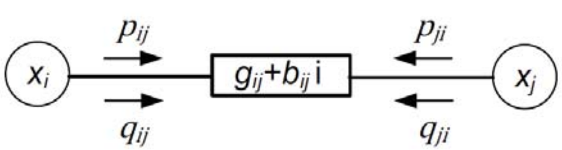
acyclic

Theorem (Sojoudi & Lavaei 2012): SDP works under positive locational marginal prices (LMPs).

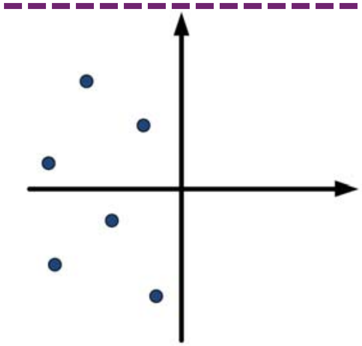
Structured Optimization

- Sign-definite real set : **"T "** is sign definite if **T** and **–T** are separable in **R**.
- Sign-definite complex set : **"T "** is sign definite if **T** and **–T** are separable in **R**^{2}





Coefficients of $x_i\, x_j$



Sign definite due to passivity for power systems

Theorem (Sojoudi & Lavaei 2014): SDP is exact for real/complex polynomial optimization with sign-define coefficients.

Findings for Power Systems

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

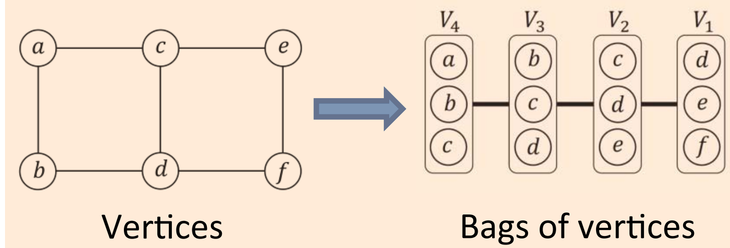
We generalized to and incorporated unit commitment, state estimation, security analysis, transmission switching, and distributed control

Our research revitalized the area:

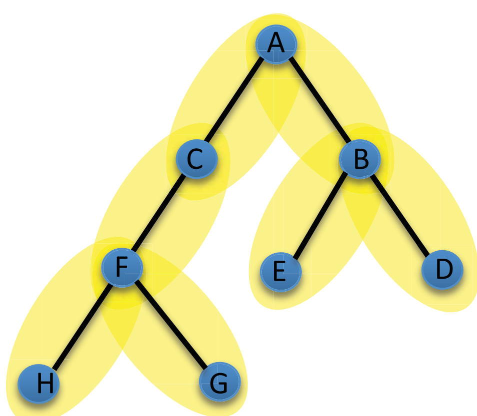
- Follow-up work in academia
- Interest from industry
- Many talks at FERC's summer workshops

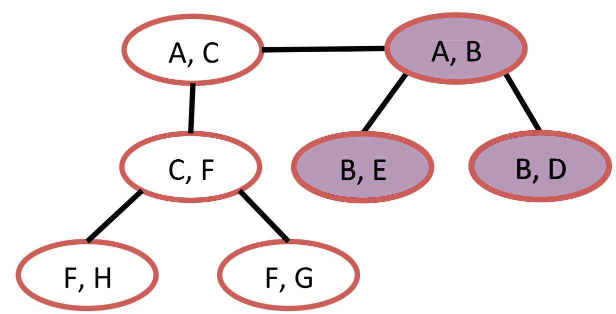
Graph Notions

- **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

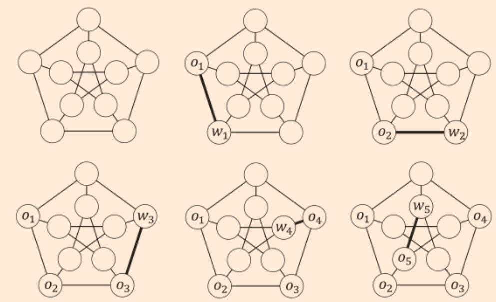




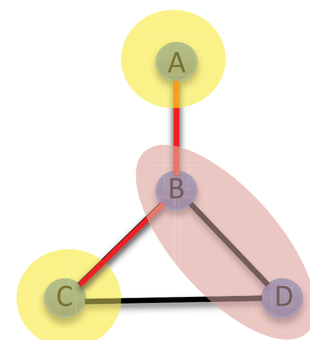
Treewidth = 1

Graph Notions

- **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 $\longrightarrow$ D $\longrightarrow$ B

OS-vertex sequence: C $\longrightarrow$ A

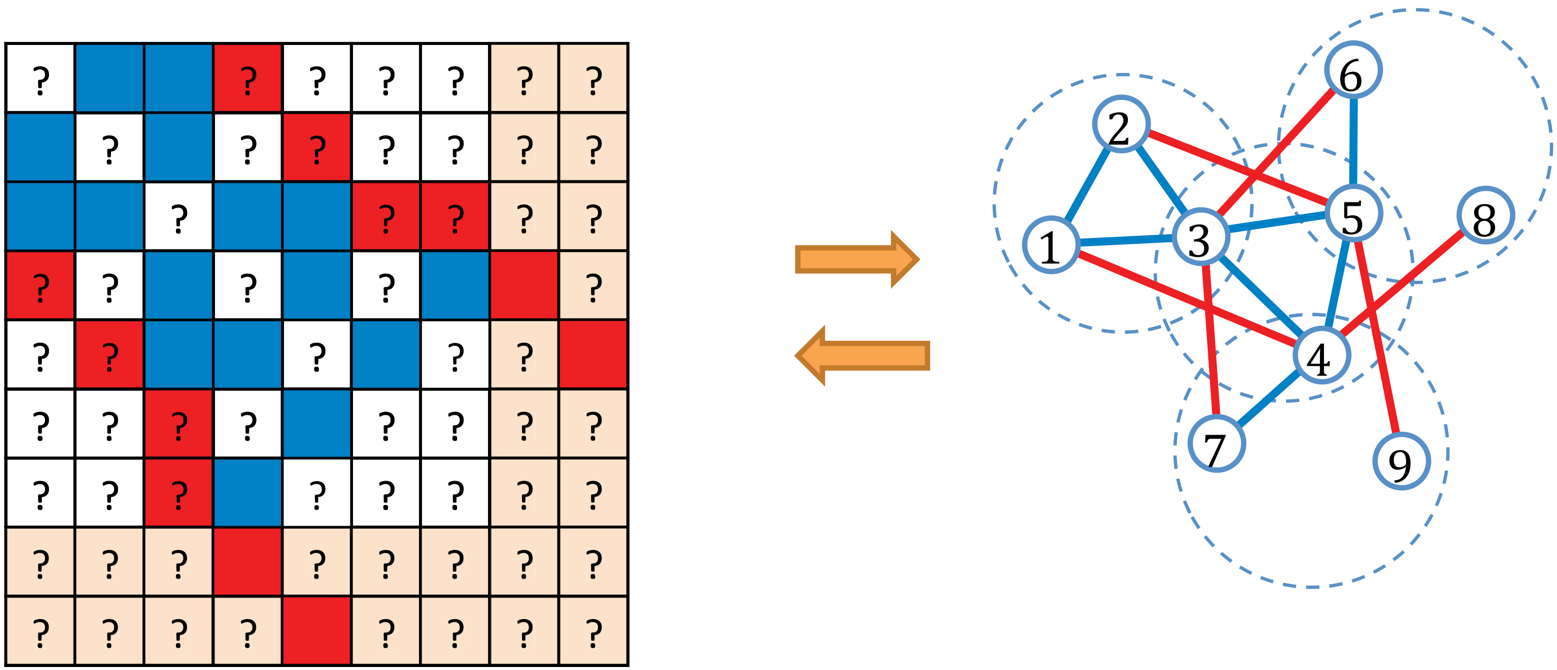
Low-Rank Solution

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

Example: Low-rank PSD completion

Sparsity graph:

Enriched supergraph:



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

Low-Rank Solution

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

SDP

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

Embedding + penalization of red entries

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

Sparsity Graph *G* (union of supports of M_i's)

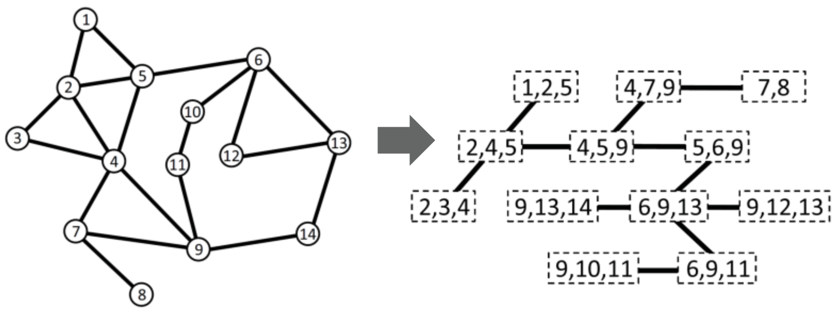
Theorem (Madani, Sojoudi & Lavaei 2017): SDP has a solution such that

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

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

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).

Penalty Design

How to promote low-rank soltuions?

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

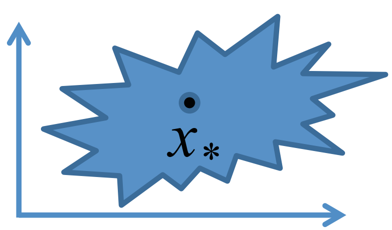
Proposed penalty:

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

Guess for solution of original QCQP: x_*

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

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



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

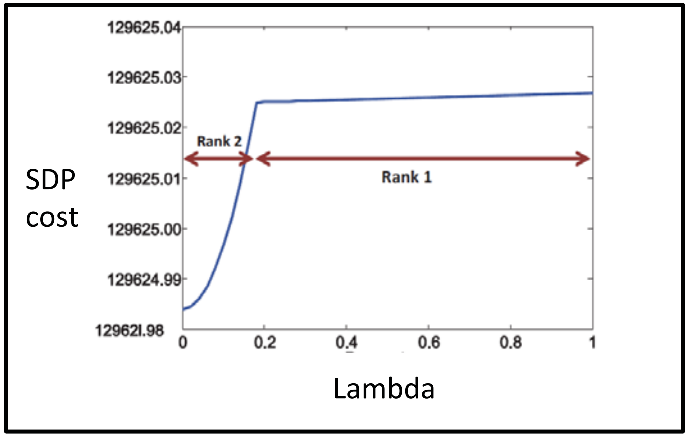
Sequential Penalized SDP:

- ❑ Start with a feasible point or an infeasible point not too far from the feasible set.
- ❑ Design M
- ❑ Solve penalized SDP
- ❑ Use the solution to design new M
- ❑ Solve penalized SDP
- ❑ ….

Near-Global Solutions

Strategy: Penalize reactive loss over problematic lines

- **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

- **Generalization to mixed-inter nonlinear program** (Fattahi, Lavaei & Atamturk 2016, 2017): Unit commitment and optimal transmission switching

Near-Global Solutions

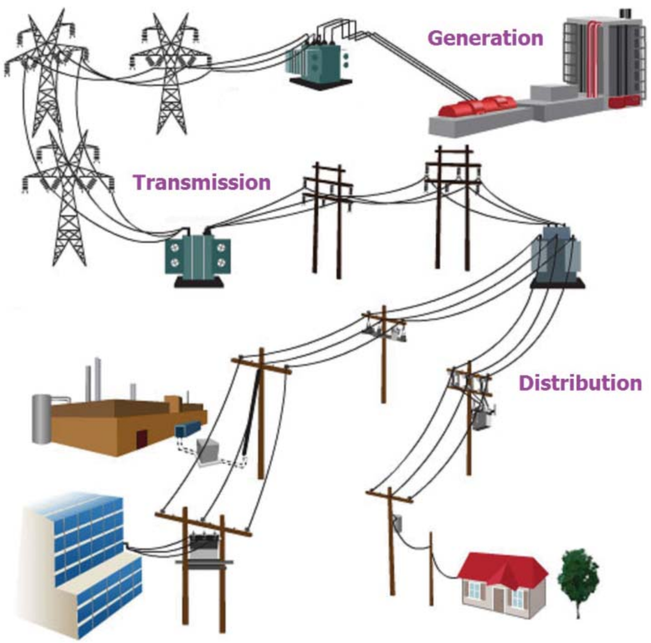
Library of quadratic programming (QPLIB):

Inst	Sequential SDP with flat start							BARON			COUENNE		
	η	i^{tight}	q_0^{tight}	i^{stop}	q_0^{stop}	GAP(%)	t(s)	LB	UB	GAP(%)	LB	UP	GAP(%)
0343	1e+2	1	-2.487	23	-6.382	0.06	1.43	-7668.005	-6.386	0.00	$-\infty$	-1.688	73.56
0911	5e+0	1	-5.653	16	-30.934	3.78	2.52	$-\infty$	0.000	100.00	$-\infty$	0.000	100.00
0975	5e+0	1	-11.619	12	-36.436	3.74	0.94	-171.113	-37.794	0.16	-76.477	-0.913	97.59
1055	5e+0	1	-12.157	14	-32.868	0.51	1.27	-199.457	-32.879	0.48	$-\infty$	0.000	100.00
1143	1e+1	1	-19.447	11	-55.628	2.83	1.07	-383.133	-57.247	0.00	$-\infty$	-2.874	94.98
1157	1e+0	1	-10.154	6	-10.947	0.01	0.32	-12.165	-10.948	0.00	$-\infty$	-7.925	27.61
1353	1e+0	1	-4.953	9	-7.714	0.00	0.49	-70.152	-7.714	0.00	-15.014	-0.298	96.13
1423	2e+0	1	-7.842	11	-14.689	1.86	0.91	-74.783	-14.967	0.00	$-\infty$	-5.574	62.76
1437	1e+0	1	-6.170	7	-7.789	0.01	0.39	-83.849	-7.789	0.00	$-\infty$	-1.895	75.67
1451	1e+1	1	-38.399	14	-87.405	0.20	3.31	$-\infty$	–	–	$-\infty$	–	–
1493	1e+1	1	-16.634	14	-42.141	2.36	0.70	-393.302	-43.160	0.00	$-\infty$	-12.785	70.38
1507	1e+0	1	-6.186	7	-8.301	0.01	0.42	-38.444	-8.301	0.00	$-\infty$	-3.325	59.95
1535	2e+0	1	-3.621	13	-11.133	3.91	4.20	$-\infty$	–	–	$-\infty$	–	–
1619	1e+0	1	-7.166	7	-9.217	0.01	0.89	$-\infty$	–	–	$-\infty$	–	–
1661	1e+0	1	-13.543	14	-15.953	0.01	0.84	-133.138	-15.955	0.00	$-\infty$	-6.964	56.35
1675	1e+1	1	-28.592	9	-75.526	0.19	0.49	-431.719	-75.669	0.00	$-\infty$	-6.525	91.38
1703	1e+1	1	-88.887	10	-132.637	0.12	1.84	$-\infty$	–	–	$-\infty$	–	–
1745	1e+1	1	-38.030	17	-72.366	0.02	3.62	$-\infty$	–	–	$-\infty$	–	–
1773	2e+0	1	-4.967	21	-14.638	0.02	1.21	-115.845	-14.642	0.00	$-\infty$	–	–
1886	1e+1	1	-24.032	11	-78.651	0.03	1.31	$-\infty$	0.000	100.00	$-\infty$	0.000	100.00
1913	5e+0	1	-27.656	10	-51.881	0.44	1.11	$-\infty$	0.000	100.00	$-\infty$	0.000	100.00
1922	1e+1	1	-10.284	19	-35.725	0.63	1.31	-51.020	-35.916	0.10	$-\infty$	-7.695	78.59
1931	5e+0	1	-38.753	9	-54.290	2.55	0.78	$-\infty$	0.000	100.00	$-\infty$	-1.468	97.37
1967	1e+1	1	-73.098	13	-106.895	0.64	3.45	-621.561	-106.91	0.62	$-\infty$	0.000	100.00

We have developed a strong theoretical result on why sequential penalized SDP works.

Outline

Background on Power Systems



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

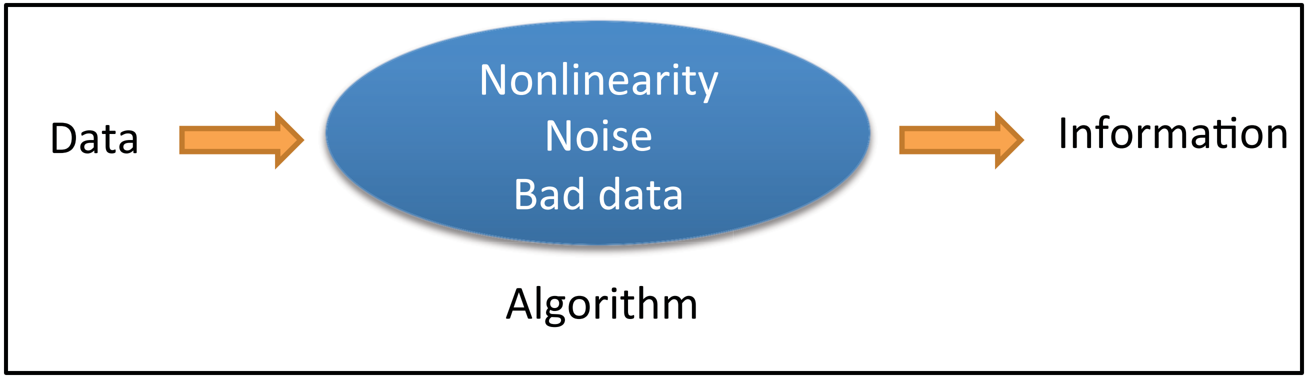
Convexification via Conic Optimization for Sparse Problems

Nonlinear Data Analytics via Conic Optimization

Near-Linear time Algorithm for Conic Optimization

Online Non-convex Optimization

Data Analytics



$$\underbrace{x_r}_{\text{measurement}} = \langle \underbrace{\mathbf{v}\mathbf{v}^*}_{\text{unknown state}}, \mathbf{M}_r \rangle + \underbrace{\omega_r}_{\text{noise}} + \underbrace{\eta_r}_{\text{bad data}}, \qquad r \in \mathcal{M}$$

- **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 (Molybog, Madani & Lavaei 2018):

Exact recovery if:

Condition 1: $\frac{\sqrt{cn+c'\log\frac{2}{\delta}}}{(1-2\varepsilon)\sqrt{|\mathcal{B}|}} < \Delta$

Condition 2: $\alpha = \frac{8\rho(\hat{\mathbf{x}},\mathbf{x})n\sqrt{1+\Delta}}{1-\Delta-4\rho(\hat{\mathbf{x}},\mathbf{x})n\sqrt{1-\Delta}} > 0$

Condition 3: $\sqrt{|\mathcal{G}|} > \alpha|\mathcal{B}|^{\frac{3}{2}} + \sqrt{n\frac{1+\Delta}{1-\Delta}}|\mathcal{B}|$

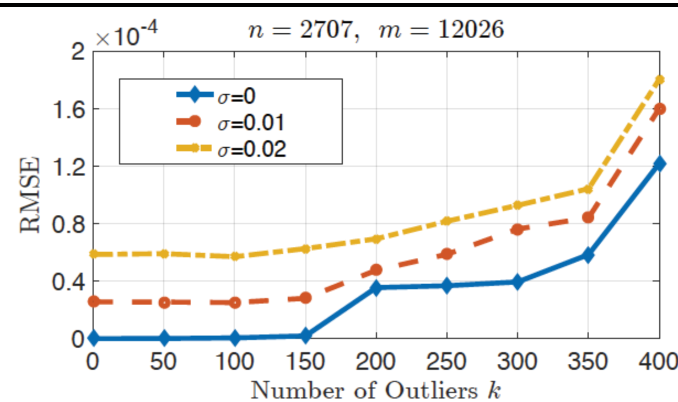
Condition 4: $\mu^* > \frac{2\rho(\hat{\mathbf{x}},\mathbf{x})\sqrt{n}}{\sqrt{(1-\Delta)|\mathcal{G}|}-\sqrt{n(1+\Delta)}|\mathcal{B}|}$

Condition 5: $\mu^* < \frac{\sqrt{(1-\Delta)|\mathcal{G}|}-4n\rho(\hat{\mathbf{x}},\mathbf{x})\sqrt{|\mathcal{G}|}}{2|\mathcal{B}|\sqrt{|\mathcal{G}|}\left(\sqrt{n(1+\Delta)}|\mathcal{B}|+\sqrt{(1-\Delta)}\right)}$

Limit for Gaussian systems:

bad measurements
=
O((#good measurements)$^{0.5}$)

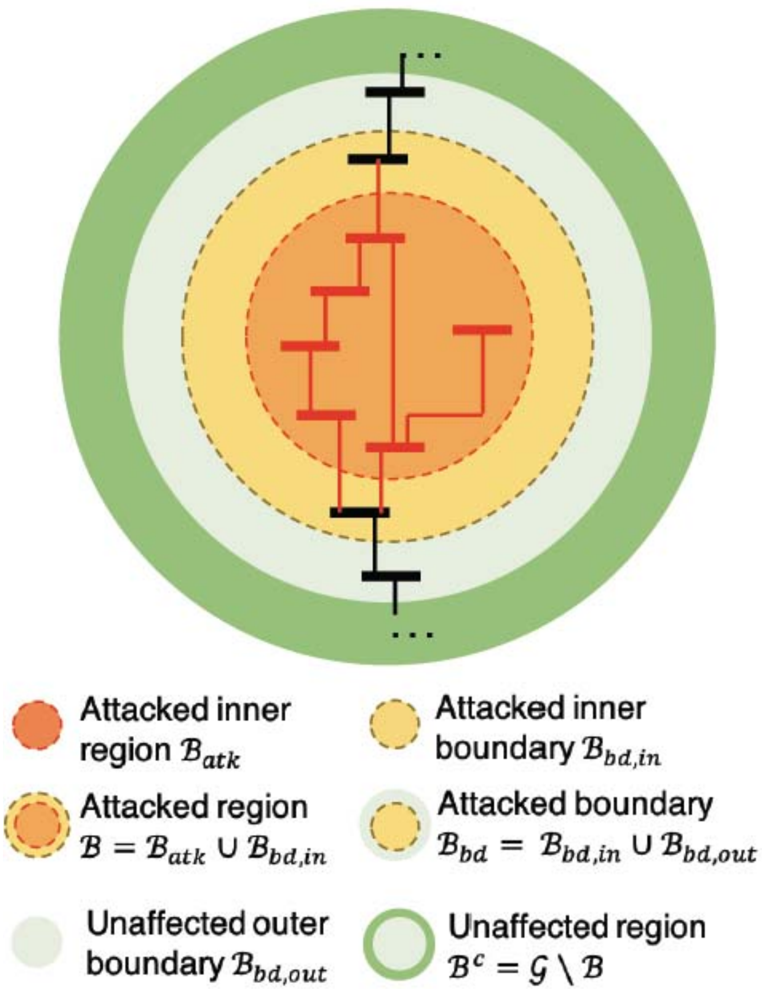
PEGASE 1354-bus system under three noise levels:



- **Data analytics subject to cyber attacks** (Jin, Lavaei & Johansson 2017)

Learning

- ❑ Can the proposed method detect attacks (identify subgraph)?
- ❑ Due to unobservability, traditional ML results do not apply here.

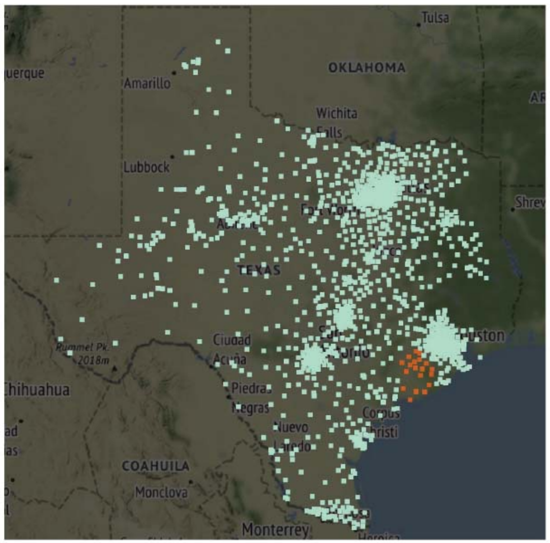


We developed notion of **graphical mutual incoherence (GMI)**.

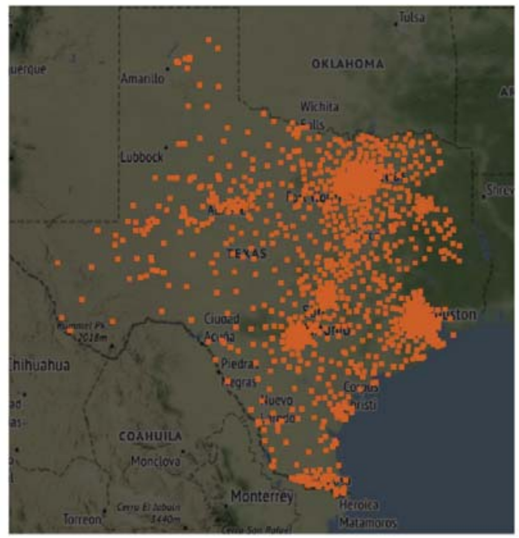
If all boundary lines satisfy GMI, then the affected subgraph can be detected via a two-stage learning mechanism.

Learning

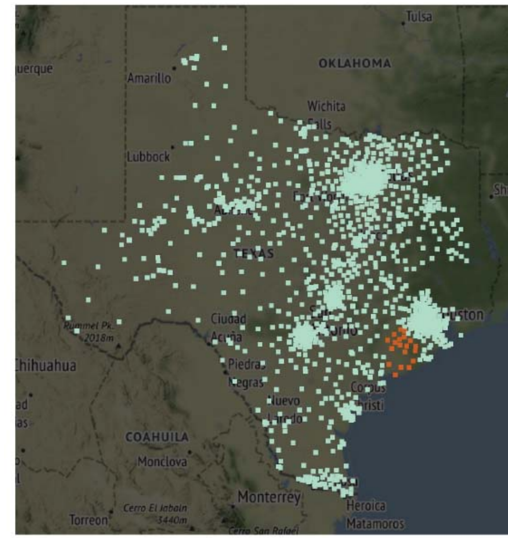
- ❑ Attack on a transmission company in Huston and effect on ERCOT:



Red: Attacked
Green: Protected



ISO method affected all nodes

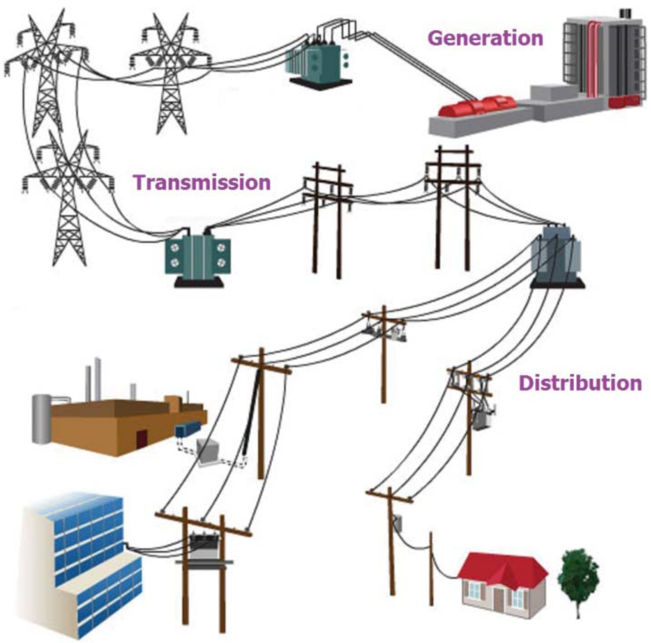


Graphical mutual incoherence detects it

- ❑ Other works: Design worst attack, learning of topological errors, etc.

Outline

Background on Power Systems



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

Convexification via Conic Optimization for Sparse Problems

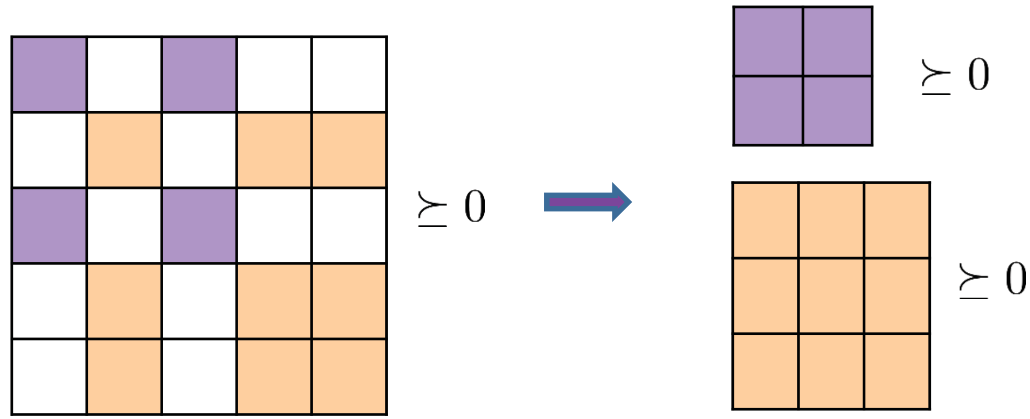
Nonlinear Data Analytics via Conic Optimization

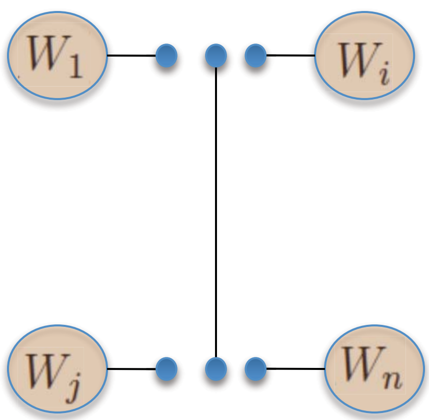
Near-Linear time Algorithm for Conic Optimization

Online Non-convex Optimization

Low-Complexity Distributed Computation

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





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

Sum of agents' objectives

Local constraints

Overlapping constraints

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 (Kalbat, Madani & Lavaei 2017):

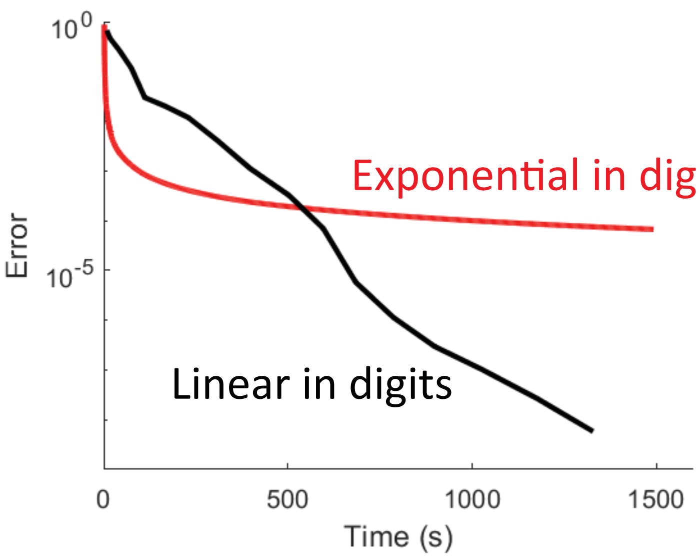
- Based on over-relaxed ADMM
- Has a guaranteed convergence
- Communications between agents
- Basic operations and eigenvalue decomposition.

- 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

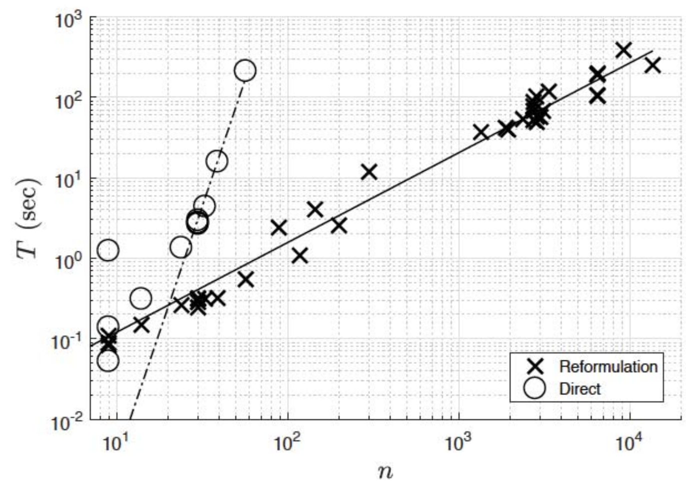
First-Order versus Interior-Point



FOM: O(n) time × O(1/ε) iters

IPM: O(n⁶) time × O(log(1/ε)) iters

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



Problem Description				Time (sec)		
#	n	m	ω	Prep	Sol	Total
35	6468	9000	30	6.62	99.85	106.47
36	6470	9005	30	6.62	183.17	189.78
37	6495	9019	31	6.78	97.02	103.80
38	6515	9037	31	6.66	191.60	198.26
39	9241	16049	35	10.56	375.76	386.32
40	13659	20467	35	17.48	233.32	250.81

Low-Complexity Second-order Methods

- Define *T* as the tree decomposition of the sparsity graph of SDP.
- Define *H* as the block intersection graph (supergraph of *T*).

Theorem (Zhang & Lavaei 2018)**:** SDP can be reformulated so that interior-point method solves it with the complexity

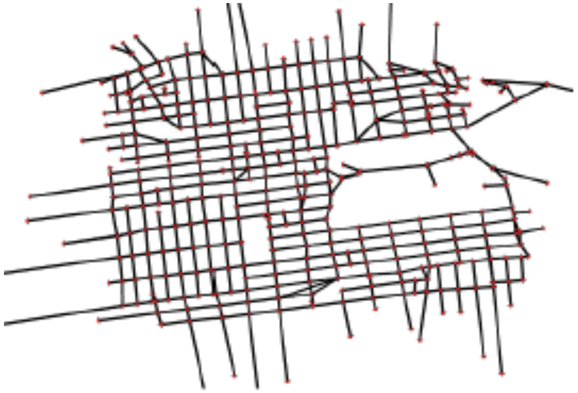
$O\left(\omega^7\tau^2 n^{1.5}\log\left(\frac{1}{\epsilon}\right)\right)$ time and $O(\omega^4\tau n)$ memory, where $\omega = \mathrm{tw}(\mathcal{T})+1$ and $\tau = \mathrm{tw}(\mathcal{H})+1$

- Power systems:
  - Bounded treewidth
  - Bounded degree
  - Bounded number of constraints per node
- Complexity of solving SDP relaxation of OPF:

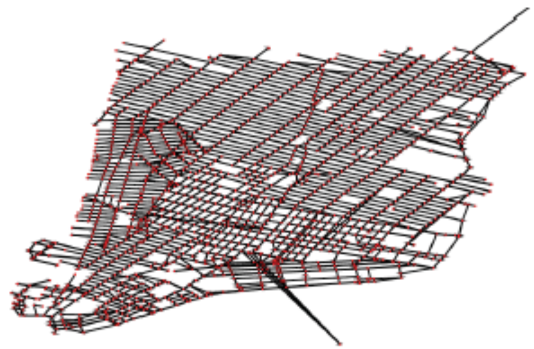
$$O(n^{1.5}\log\epsilon^{-1}) \text{ time and } O(n) \text{ memory}$$

Traffic Systems

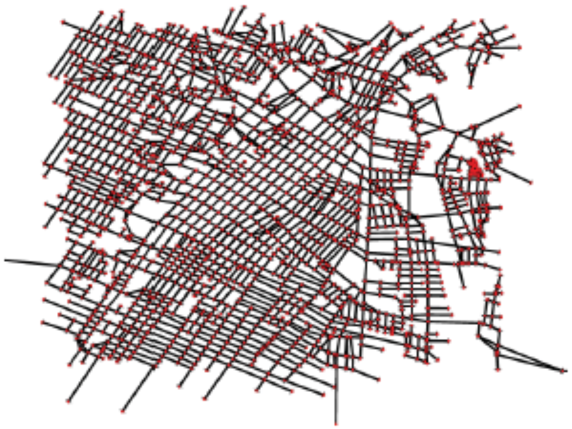
Theorem (Ouyang, Zhang, Lavaei & Varaiya 2018): SDP returns a suboptimal solution for traffic signal optimization with a constant approximation ratio in $O(n^{1.5})$ time and $O(n)$ memory.



Berkeley

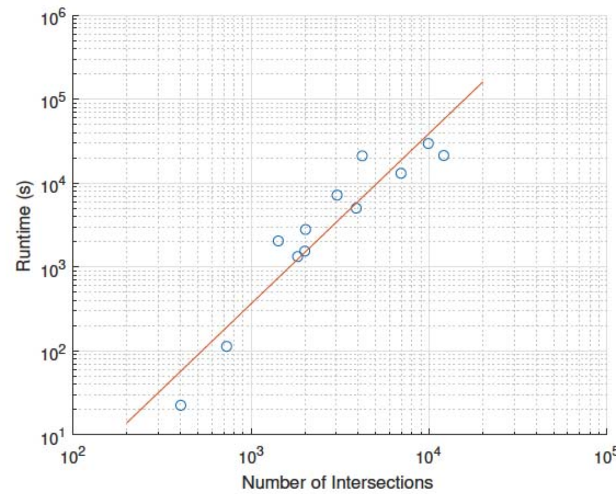


Manhattan



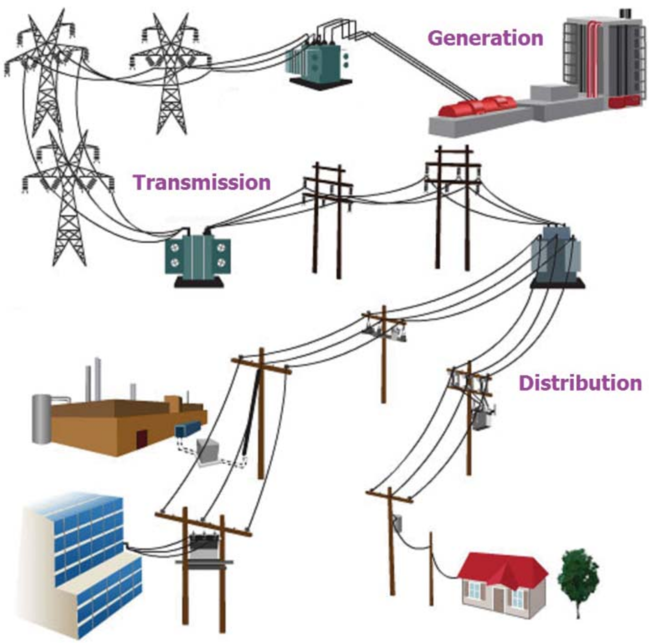
Los Angeles

Cases	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{L}\rvert$	ω	LB	UB	Runtime (s)
Berkeley-1	405	1122	14	79209	79447	22
Berkeley-2	2036	5789	35	477449	479736	2745
Berkeley-3	7000	19222	41	1588518	1597013	12838
Berkeley-4	12176	33725	42	2795242	2811025	20942
Manhattan-1	1432	2745	31	301366	302965	2005
Manhattan-2	2016	3854	31	417186	419601	1516
Manhattan-3	3923	7841	37	780878	787402	4932
Manhattan-4	9968	20945	39	2022526	2040154	29117
Los Angeles-1	733	2180	22	182811	183440	1305
Los Angeles-2	1838	5170	36	458209	460905	1305
Los Angeles-3	3062	8838	43	747804	752099	7045
Los Angeles-4	4239	12773	50	1139072	1147166	20733



Outline

Background on Power Systems



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

Convexification via Conic Optimization for Sparse Problems

Nonlinear Data Analytics via Conic Optimization

Near-Linear time Algorithm for Conic Optimization

Online Non-convex Optimization

Online Optimization

- Online Non-convex Optimization:

$$\inf_{x \in \mathbb{R}^n} f(x,t) \quad \text{s.t.} \quad h_i(x) = d_i(t), \qquad i = 1, \ldots, m$$

Time-varying objective

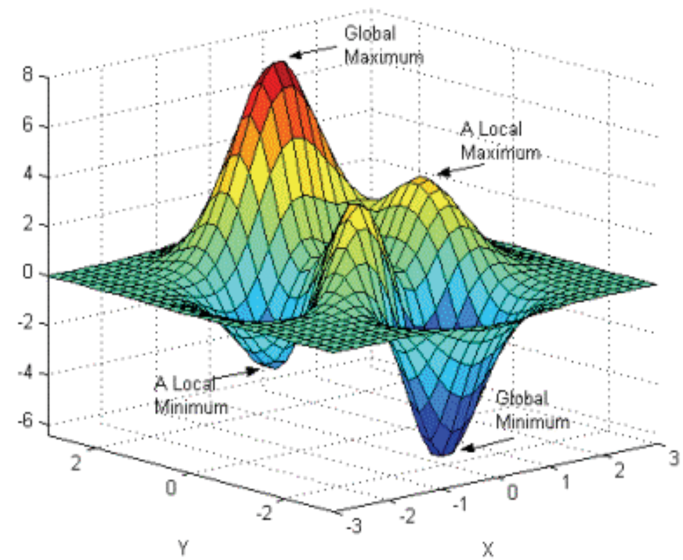
Time-varying data

- **Online:** Sequential decision-making at times 0,1,2,3,…:
  - **Initialization:** Use the previous (optimal) value of x as an initial point
  - **Temporal constraint:** Impose a constraint on the time-varying nature of x
  - **Local search:** Use cheap algorithms (SGD) to solve each snapshot of the problem

- **Offline:** Solve each problem without using prior knowledge and temporal constraints
  - **Local search:** Use cheap algorithms (SGD) with an arbitrary initial point

Online Optimization

- **Offline optimization:**
  - Different types of solutions due to nonlinearity
  - Spurious: A non-global local solution
  - Spurious solutions in ML: Matrix sensing, training of deep neural nets, etc.



Counterpart of spurious solutions for online optimization?

- We introduced the notion of **spurious trajectory**.
- It is based on deep notions in control theory (reachability/stability for nonlinear time-varying continuous-time ODEs)

Online Optimization

- Online Non-convex Optimization:

$$\inf_{x\in\mathbb{R}^n} f(x,t) \quad \text{s.t.} \quad h_i(x) = d_i(t), \qquad i = 1,\ldots,m$$

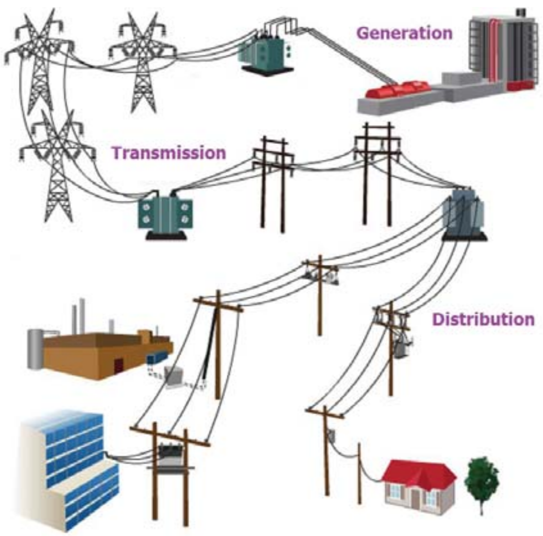
- Quality of solution at time *k+1*:
  - Optimality of *x(k+1)* → KKT point
  - Allowable change in state (*x(k+1)-x(k)*) → **α**
  - Data variation from time *k* to *k+1* → **β**

$$\begin{array}{lll} \inf_{x\in\mathbb{R}^n} & f(x,t_{k+1}) + \alpha \frac{\|x-x_k\|^2}{2(t_{k+1}-t_k)} \\ s.t. & h_i(x) = d_i(t_{k+1}), \quad i = 1,\ldots,m \end{array}$$

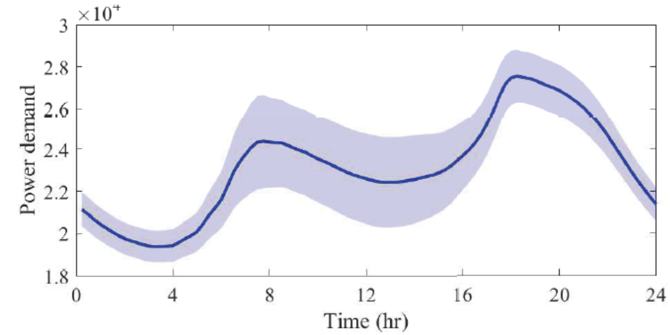
- Online trajectory: $x^*(0), x^*(1), x^*(2),\ldots$ where $x^*(0)$ is a local min.
- Offline trajectory: $y^*(0), y^*(1), y^*(2),\ldots$ where $y^*(k)$ is a global min of offline optimization
- **No spurious trajectory:** Online trajectory converges to offline trajectory (lies within basin of attraction) no matter how bad the initila point is.

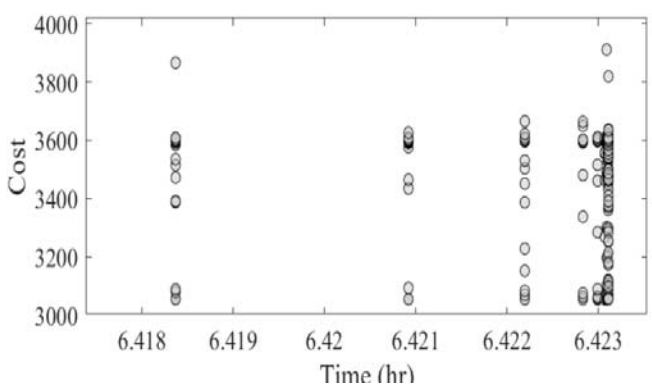
Online Optimization

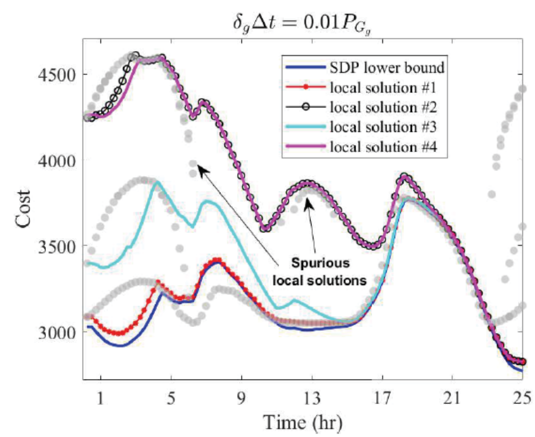
- Time-varying optimal power flow:



Average daily data for California in Jan. 2019







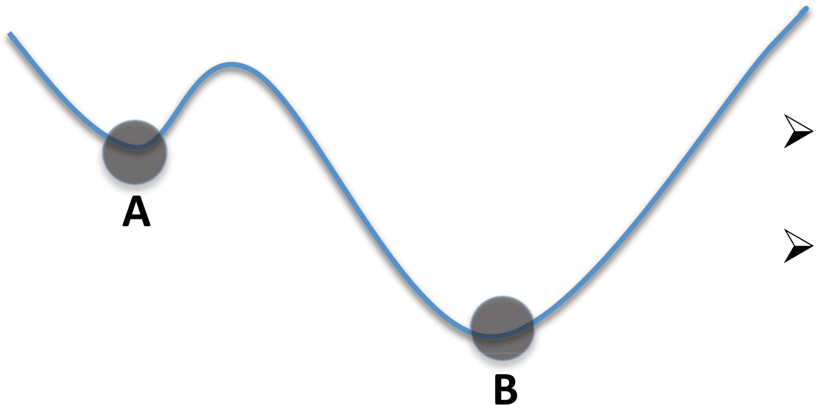
Start at a local min, but tomorrow the same time, the solution is global.

Many local minima at different times

Online trajectories converge

Online Optimization

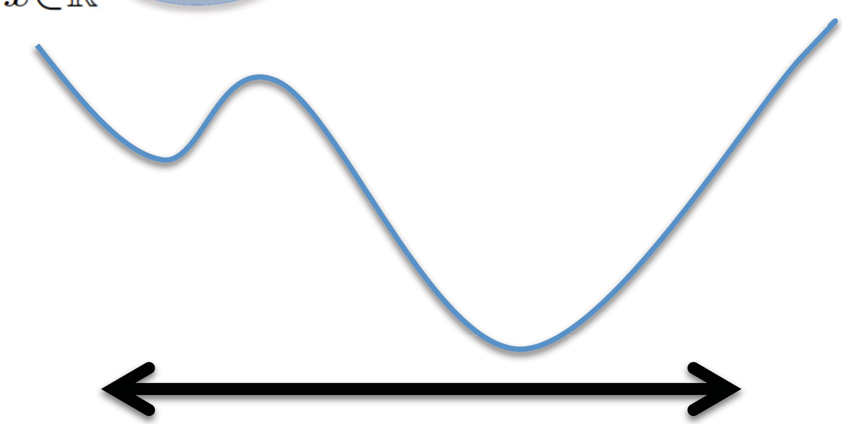
❑ **Escaping local minima for static optimization** (shallow spurious solutions):



- Local search initialized at A can't go to B.
- How to escape A?

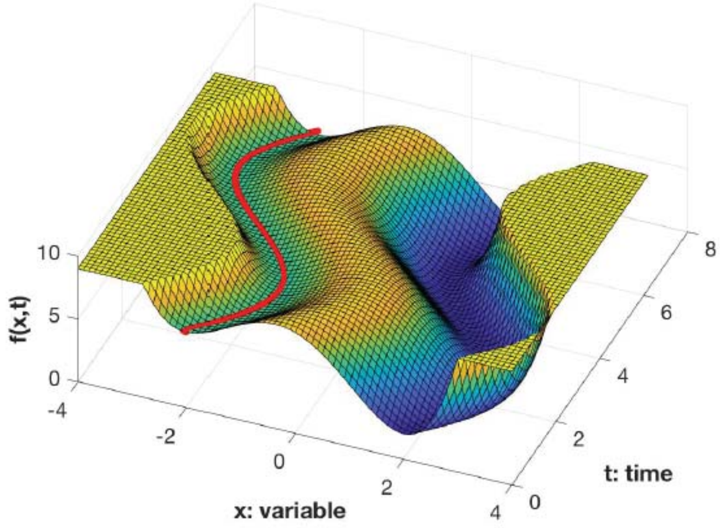
❑ **Conversion to online optimization:**

$$\inf_{x \in \mathbb{R}} f(x,t)$$

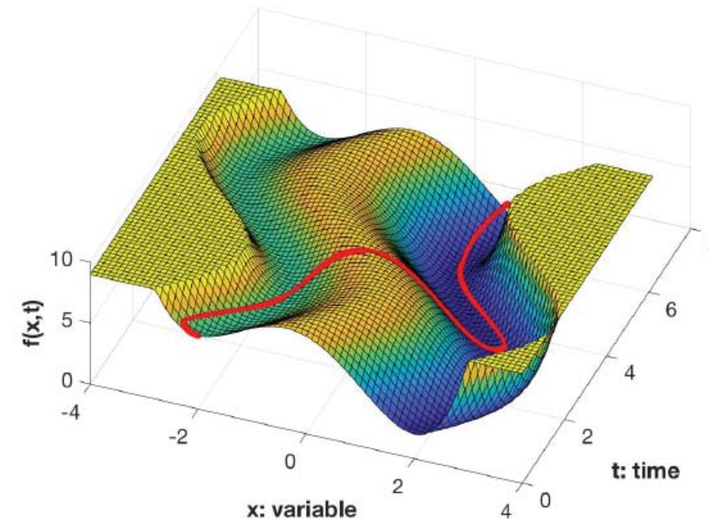


Oscillate at a constant speed

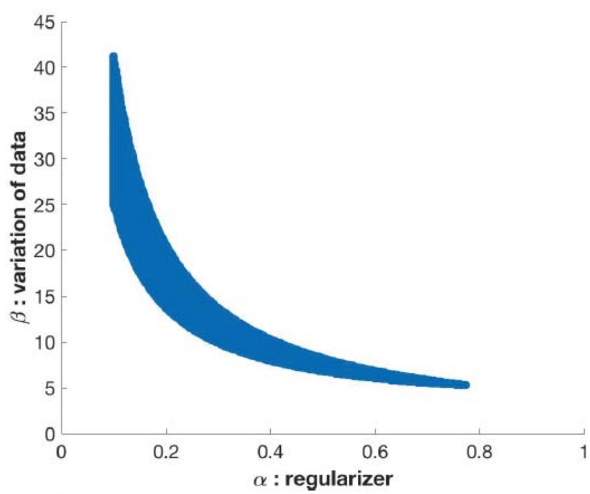
Online Optimization



(d) Spurious trajectory for $\alpha = 0.2$ and $\beta = 5$.



(c) Non-spurious trajectory for $\alpha = 0.4$ and $\beta = 10$.



(b) Sufficient condition in blue in function of α, β for absence of spurious trajectories.

Online Optimization

- Online Non-convex Optimization:

$$\inf_{x \in \mathbb{R}^n} f(x,t) \quad \text{s.t.} \quad h_i(x) = d_i(t), \quad i = 1, \ldots, m$$

Continuous limit

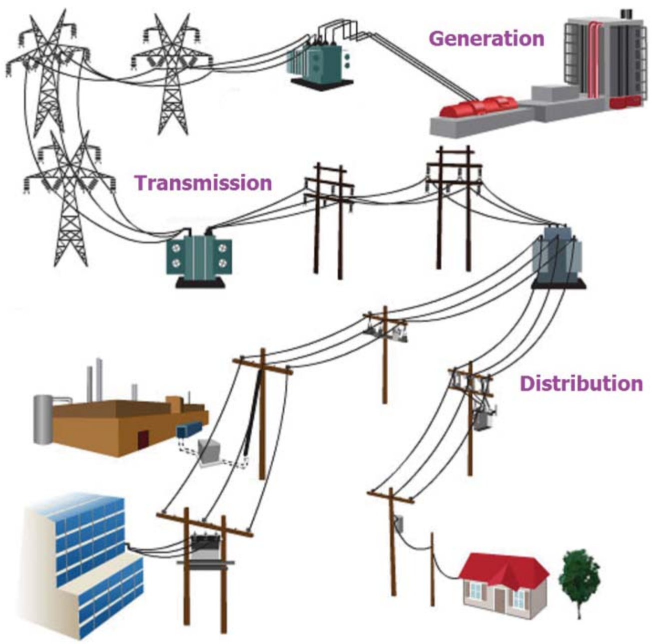
$$\dot{x} = -\frac{1}{2\alpha}\left(I - \mathcal{J}_h(x)^\top (\mathcal{J}_h(x)\mathcal{J}_h(x)^\top)^{-1}\mathcal{J}_h(x)\right)\nabla_x f(x,t) + \mathcal{J}_h(x)^\top (\mathcal{J}_h(x)\mathcal{J}_h(x)^\top)^{-1}\dot{b}$$

- This ODE is quite different from existing continuous limits of optimization problems (no dynamics for dual variables and carrying local optimality behavior pointwise)
- Checking non-existence of spurious trajectories is equivalent to analyzing reachable set of this ODE.
- Necessary condition: Linearize ODC around a locally optimal offline trajectory and certify instability:

$$\mathcal{J}_p(z) = -\frac{1}{\alpha}\underbrace{(\nabla^2 f(z) + \mu\nabla^2 h(z))}_{\text{SOC}}\underbrace{\left(I - \frac{\nabla h(z)\nabla h(z)^\top}{\|\nabla h(z)\|^2}\right)}_{\text{Projection on manifold}} + \underbrace{\frac{\nabla^2 h(z)}{\|\nabla h(z)\|^2}\left(I - \frac{2\nabla h(z)\nabla h(z)^\top}{\|\nabla h(z)\|^2}\right)\dot{d}}_{\text{Role of data}}$$

Conclusions

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



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

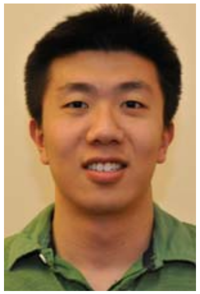
Convexification via Conic Optimization for Sparse Problems

Nonlinear Data Analytics via Conic Optimization

Near-Linear time Algorithm for Conic Optimization

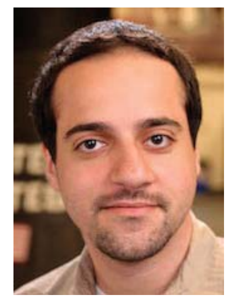
Online Non-convex Optimization

Joint work with







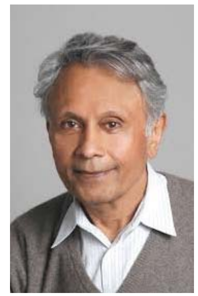


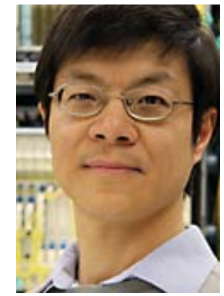




Joint work with

- Pravin Variya (UC Berkeley)
- Ross Baldick (UT Austin)
- Steven Low (Caltech)
- Alper Atamturk (UC Berkeley)
- Karl Henrik Johansson (KTH)
- Somayeh Sojoudi (UC Berkeley)





Funding Acknowledgements

DARPA

Office of Naval Research
ONR
Science & Technology

Air Force Office of Scientific Research
United States Air Force

U.S. ARMY

NSF

SIEBEL
ENERGY INSTITUTE

arpa·e
U.S. DEPARTMENT OF
ENERGY

Google™

TBSI
Tsinghua-Berkeley Shenzhen Institute