Graph Theoretic Algorithm for Nonlinear Power Optimization Problems

Javad Lavaei

Department of Electrical Engineering
Columbia University

Joint work with:
Ramtin Madani, Ghazal Fazelnia and Abdulrahman Kalbat (Columbia University)
Somayeh Sojoudi (New York University)

Software System

- **Decision making towards real-time operation**

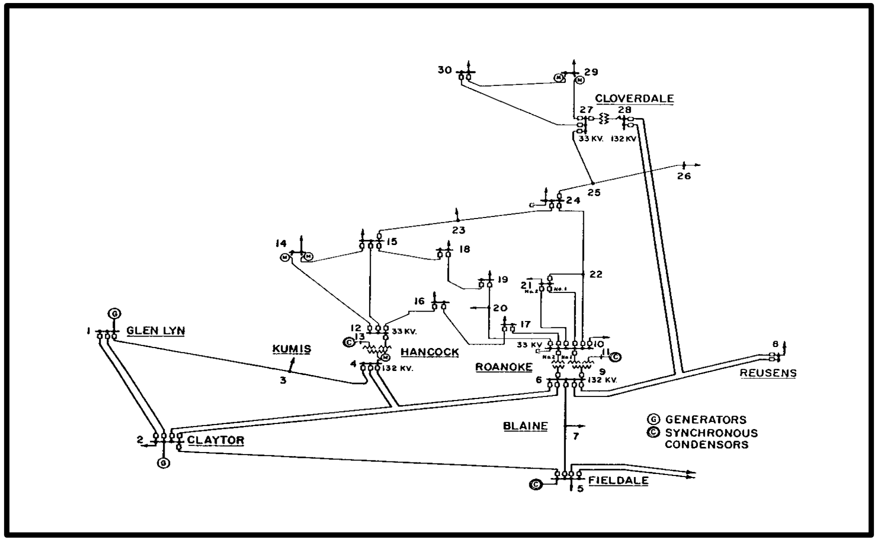
  - Slow time scale: Optimize the operating cost → Centralized or distributed optimization

  - Fast time scale: Regulate the signals → Distributed control

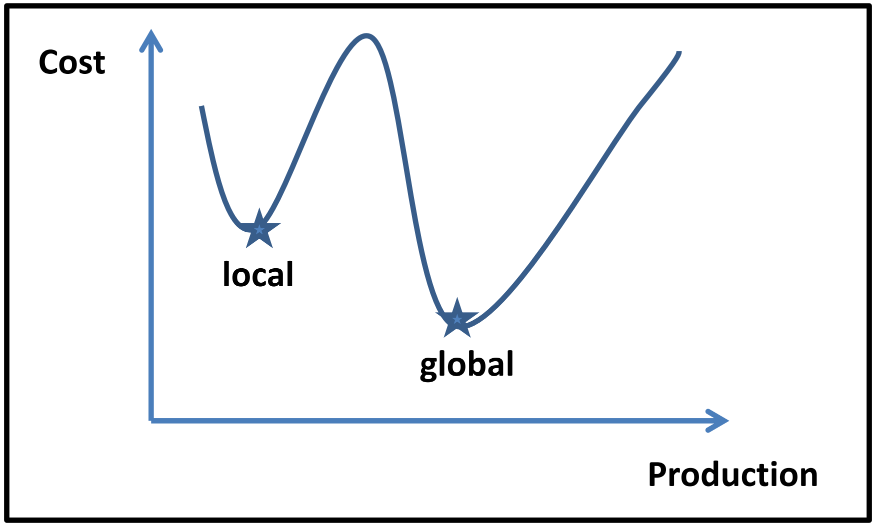
- **Approach:** SDP relaxation, matrix completion, tree decomposition, low-rank optimization

Optimization

- **Optimization:**
  - Optimal power flow (OPF)
  - Security-constrained OPF
  - State estimation
  - Network reconfiguration
  - Unit commitment
  - Dynamic energy management

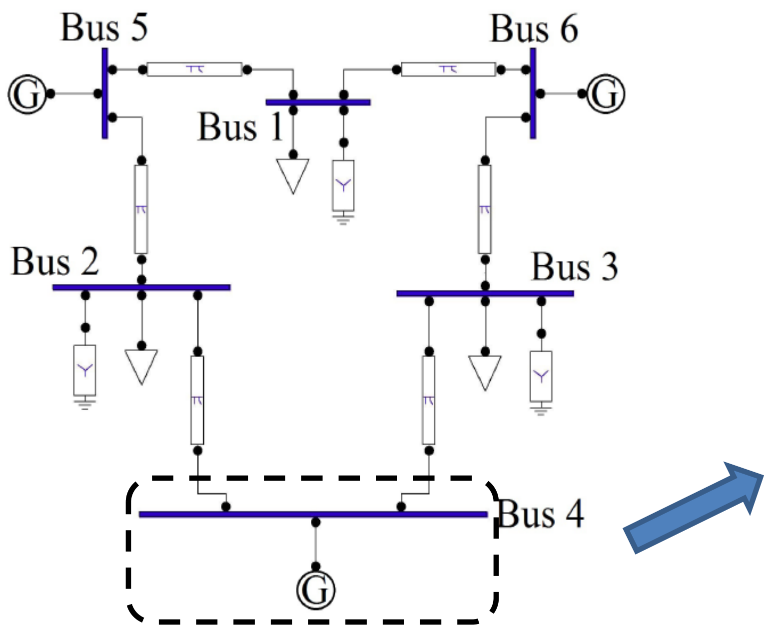


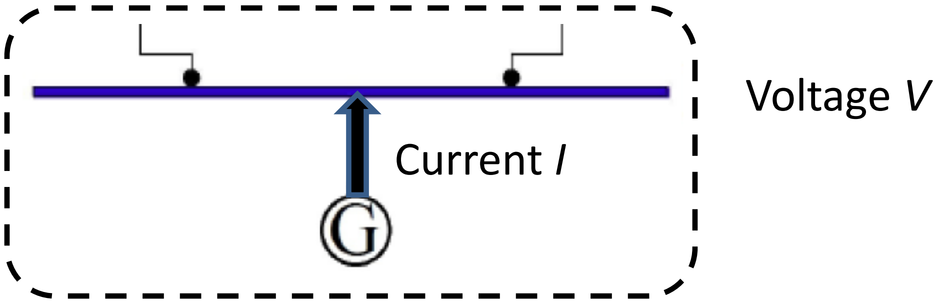
- **Issue of non-convexity:**
  - Discrete parameters
  - Nonlinearity in continuous variables



- Challenge: ~90% of decisions are made in day ahead and ~10% are updated iteratively during the day so a local solution remains throughout the day.

Resource Allocation: Optimal Power Flow (OPF)





Complex power = $VI^{*}=P + Q$ i

OPF: Given constant-power loads, find optimal *P*'s subject to:

- Demand constraints
- Constraints on *V*'s, *P*'s, and *Q*'s.

Broad Interest in Optimal Power Flow

- OPF-based problems solved on different time scales:
  - Electricity market
  - Real-time operation
  - Security assessment
  - Transmission planning

- Existing methods based on linearization or local search

- **Question:** How to find the best solution using a scalable robust algorithm?

- Huge literature since 1962 by power, OR and Econ people

Penalized Semidefinite Programming (SDP) Relaxation

Quadratic optimization in $\mathbf{v}$ → Linear optimization in $\mathbf{v}\mathbf{v}^*$ → Replace $\mathbf{v}\mathbf{v}^*$ with a matrix $\mathbf{W}$ subject to $\mathbf{W} \succeq 0$ and $\text{rank}\{\mathbf{W}\} = 1$ → Remove the rank constraint and possibly penalize its effect

$$\min_{\mathbf{v}\in\mathbb{R}^n} \quad \mathbf{v}^* M_0 \mathbf{v}$$
$$\text{s.t.} \quad \mathbf{v}^* M_i \mathbf{v} \leq 0, \quad i = 1, 2, \ldots, t$$

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

- **Exactness of SDP relaxation:**
  - Existence of a rank-1 solution
  - Implies finding a global solution

Optimal Power Flow

$$\min_{\mathbf{V},\mathbf{P}_G,\mathbf{Q}_G} \sum_{k\in\mathcal{G}} f_k(P_{G_k}) \quad (1a)$$

Subject to

$$P_k^{\min} \le P_{G_k} \le P_k^{\max} \quad (1b)$$

$$Q_k^{\min} \le Q_{G_k} \le Q_k^{\max} \quad (1c)$$

$$V_k^{\min} \le |V_k| \le V_k^{\max} \quad (1d)$$

$$\text{Re}\left\{V_l(V_l - V_m)^* y_{lm}^*\right\} \le P_{lm}^{\max} \quad (1e)$$

$$\text{trace}\{\mathbf{V}\mathbf{V}^*\mathbf{Y}^* e_k e_k^*\} = P_{G_k} - P_{D_k} + (Q_{G_k} - Q_{D_k})\mathrm{i} \quad (1f)$$

Cost

Operation

Flow

Balance

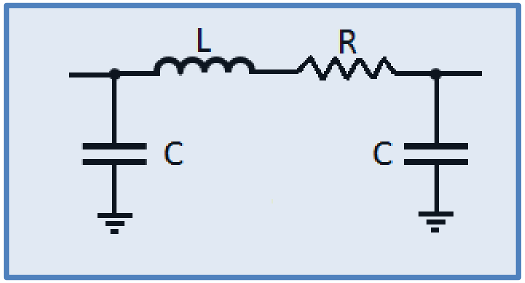
Trick: Replace $\mathbf{V}\mathbf{V}^*$ with a matrix $\mathbf{W} \succeq 0$ subject to $\text{rank}\{\mathbf{W}\} = 1$.

Old Project 1

Project 1: How to solve a given OPF in polynomial time? (joint work with Steven Low)

- A sufficient condition to globally solve OPF:
  - Numerous randomly generated systems
  - IEEE systems with 14, 30, 57, 118, 300 buses
  - European grid

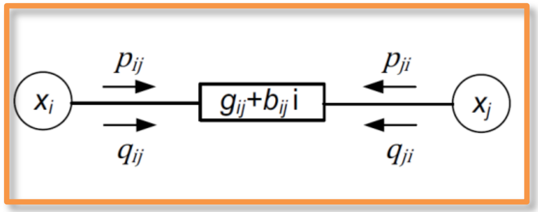
- **Various theories:** It holds widely in practice

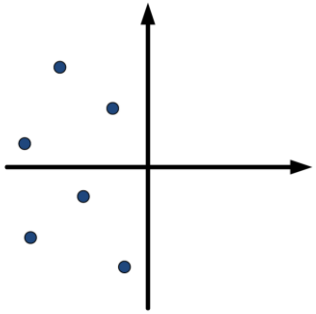


Old Project 2

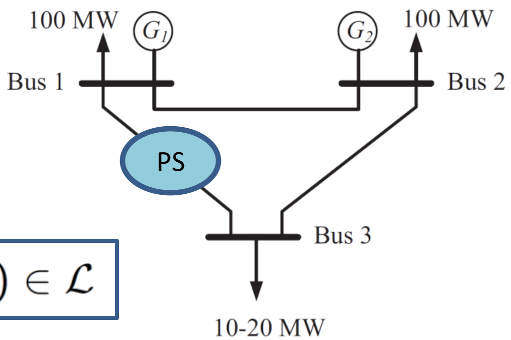
Project 2: Find network topologies over which optimization is easy? (joint work with Somayeh Sojoudi, David Tse and Baosen Zhang)

- Distribution networks are fine due to a sign definite property:





- Transmission networks may need phase shifters:



$$\text{SDP:}\quad \mathbf{W} \succeq 0 \;\Rightarrow\; \text{SOCP:}\quad \mathbf{W}\{i,j\} \succeq 0,\ \forall (i,j) \in \mathcal{L}$$

Old Project 3

Project 3: How to design a distributed algorithm for solving OPF? (joint work with Stephen Boyd, Eric Chu and Matt Kranning)

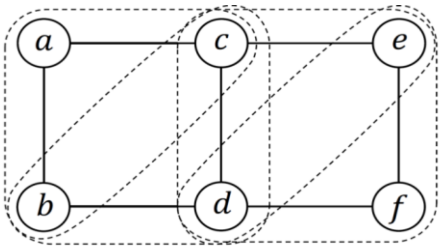
- ❑ A practical (infinitely) parallelizable algorithm using ADMM.
- ❑ It solves 10,000-bus OPF in 0.85 seconds on a single core machine.

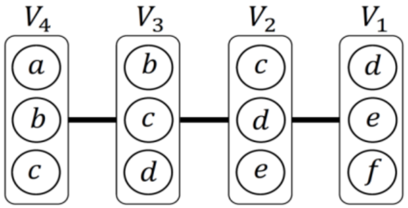
Graph-Theoretic SDP Relaxation

- **Issues 1:** What if we get a low-rank but not rank-1 solution?
- **Issue 2:** How to deal with a computationally-expensive SDP?
- **Approach:**
  1. Use a graph-theoretic approach to break down complexity
  2. This also tells what lines of the network cause non-convexity
  3. We first sparsify SDP and then penalize problematic lines
- **Example borrowed from Bukhsh et al.:**
  1. Modify IEEE 118-bus system with 3 local solutions: 129625.03, 177984.32 and 195695.54.
  2. Our method finds the best one.

Treewidth

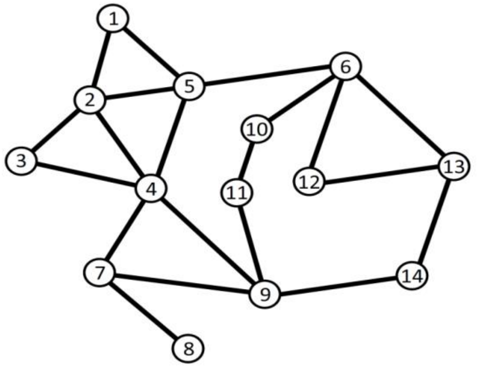
❑ **Tree decomposition:**

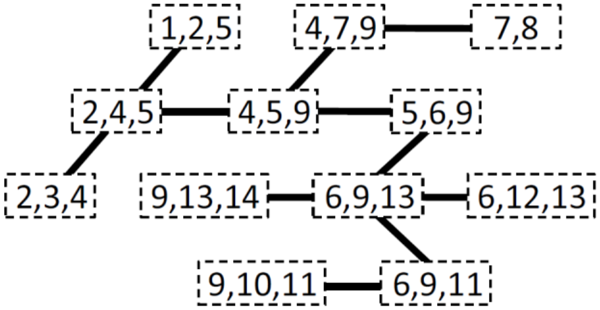




❑ **Treewidth of graph:** The smallest width of all tree decompositions

❑ Treewidth: 1 for a tree, 2 for IEEE 14-bus and 2-26 for IEEE and Polish systems





Power Networks

- ❑ **Break down the complexity through sparsification:**

W = positive semidefinite

⇕

submatrices of W induced by bags of tree = positive semidefinite

- ❑ Reduction of the number of parameters for a Polish system from ~9,000,00 to ~100K.

- ❑ **Result 1:** Rank of W at optimality ≤ Treewidth +1 (valid for SC-UC-OPF)
- ❑ **Result 2:** Rank of W at optimality ≤ maximum rank of bags (valid for SC-UC-OPF)
- ❑ **Result 3:** Lines of network in high-rank bags are the source of nonzero duality gap

Power Networks

- **SDP:** no penalty $\sum_{k\in\mathcal{G}} f_k(P_{G_k})$
- **Tier 1:** uniform penalty $\sum_{k\in\mathcal{G}} f_k(P_{G_k}) + \epsilon_b \sum_{k\in\mathcal{G}} Q_{G_k}$
- **Tier 2:** non-uniform penalty $\sum_{k\in\mathcal{G}} f_k(P_{G_k}) + \epsilon_b \sum_{k\in\mathcal{G}} Q_{G_k} + \epsilon_l \sum_{(l,m)\in\mathcal{L}_0} |S_{lm} + S_{ml}|$

Total reactive loss

Select loss

Polish 2383wp	Unpenalized objective	$\epsilon_b = 3500$	$\epsilon_b = 3500,\ \epsilon_l = 3000$
High rank bags	651	5	0
Problematic lines	751	9	0
Power balance violations	1338	88	0
Generator capacity violations	85	28	0
Line ratings violations	6	0	0
Cost	1861510.42	1874751.22	1874322.56

$$\text{Tolerance} = 10^{-6}$$

Power Networks

- Several bad examples have been contrived by Buksh et al.

Test cases	# local mins	# prob. bags	ϵ_b	ϵ_l	Lower bound	Upper bound	Opt. degree
WB2	2	0	0	0	877.78	877.78	%100
WB3	2	0	0	0	417.25	417.25	%100
WB5	2	3	0	500	946.53	946.58	%99.995
WB5 Mod	3	0	0	0	1482.22	1482.22	%100
LMBM3	5	0	0	0	5694.54	5694.54	%100
LMBM3_50	2	2	0	500	5789.91	5823.86	%99.807
case22loop	2	0	0	0	4538.80	4538.80	%100
case30loop	2	0	0	0	2863.06	2863.06	%100
case30loop Mod	3	0	0	0	2861.88	2861.88	%100
case39 Mod4	3	4	1	0	557.08	557.15	%99.999
case118 Mod1	3	36	10	0	129624.98	129625.19	%99.999
case118 Mod2	2	42	1	0	85987.27	85987.59	%100
case300 Mod2	2	107	0.5	50	474625.99	474643.46	%99.996

Power Networks

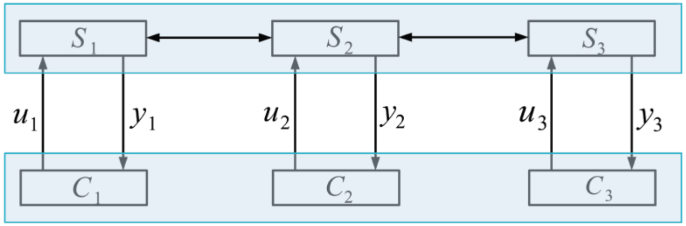
Test cases	α	TW	# prob. bags	ϵ_b	ϵ_l	Lower bound	Upper bound	Opt. degree	Com. time
Chow's 9 bus	0	2	2	10	0	5296.68	5296.68	%100	≤ 5
IEEE 14 bus	0	2	0	0	0	8081.53	8081.53	%100	≤ 5
IEEE 24 bus	0	4	0	0	0	63352.20	63352.20	%100	≤ 5
IEEE 30 bus	0	3	1	0.1	0	576.89	576.89	%100	≤ 5
NE 39 bus	0	3	1	10	0	41862.08	41864.40	%99.994	≤ 5
IEEE 57 bus	0	5	0	0	0	41737.78	41737.78	%100	≤ 5
IEEE 118 bus	0	4	61	10	0	129654.61	129660.81	%99.995	≤ 5
IEEE 300 bus	0	6	7	0.1	100	719711.63	719725.10	%99.998	13.9
Polish 2383wp	0	23	651	3500	3000	1861510.42	1874322.65	%99.316	529
Polish 2736sp	0	23	1	1500	0	1307882.29	1308270.20	%99.970	701
Polish 2737sop	0	23	3	1000	0	777626.26	777664.02	%99.995	675
Polish 2746wop	0	23	1	1000	0	1208273.91	1208453.93	%99.985	801
Polish 2746wp	0	24	1	1000	0	1631772.83	1632384.87	%99.962	699
Polish 3012wp	1	24	605	0	10000	2587740.98	2608918.45	%99.188	814
Polish 3120sp	-1.5	24	20	0	10000	2140765.92	2160800.42	%99.073	910

- We have written a solver in MATLAB to find a near-global solution.

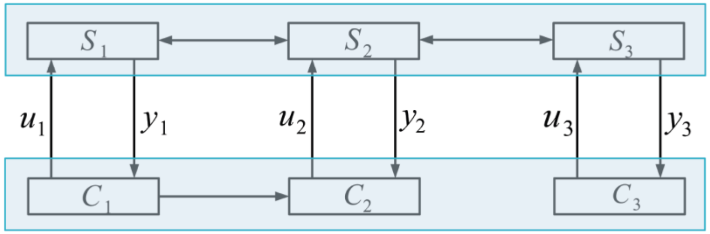
- Computation time for Polish System with ~3100 buses: ~2.4 min in MOSEK (low precision).

Distributed Control

- Computational challenges arising in the control of real-world systems:
  - Communication networks
  - Electrical power systems
  - Aerospace systems
  - Large-space flexible structures
  - Traffic systems
  - Wireless sensor networks
  - Various multi-agent systems



Decentralized control



Distributed control

Optimal Distributed control (ODC)

- **Optimal centralized control:** Easy (LQR, LQG, etc.)
- **Optimal distributed control (ODC):** NP-hard (Witsenhausen's example)

- **Stochastic ODC:** Find a structured control $u[\tau] = Ky[\tau]$ for the system:

$$\begin{cases} x[\tau+1] = Ax[\tau] + Bu[\tau] + Ed[\tau] \\ y[\tau] = Cx[\tau] + Fv[\tau] \end{cases}$$

disturbance

noise

to minimize the cost functional:

$$\lim_{\tau \to +\infty} \mathcal{E}\left(x[\tau]^T Q x[\tau] + u[\tau]^T R u[\tau]\right) + \alpha \, \mathrm{trace}\{KK^T\}$$

- Rank of an expanded SDP relaxation of SODC =1, 2 or 3.
- How to find a computationally-cheap relaxation?

First Stage of SDP Relaxation

Computationally-Cheap SDP Relaxation of SODC: This optimization problem is defined as the minimization of

$$\text{trace}\{P\Sigma_d + M\Sigma_v + \mu_2 R\mathbf{W}_{33} + \alpha\mathbf{W}_{33} + K^T RK\widehat{\Sigma}_v\} \quad (30)$$

subject to the constraints

$$\begin{bmatrix} G-\mu_1\mathbf{W}_{22} & G & (AG+BL)^T & L^T \\ G & \widehat{Q}^{-1} & 0 & 0 \\ AG+BL & 0 & G & 0 \\ L & 0 & 0 & R^{-1} \end{bmatrix} \succeq 0, \quad (31a)$$

$$\begin{bmatrix} P & I \\ I & G \end{bmatrix} \succeq 0, \quad (31b)$$

$$\begin{bmatrix} M & (BK)^T \\ BK & G \end{bmatrix} \succeq 0, \quad (31c)$$

$$\mathbf{W} := \left[\begin{array}{c|c|c} I_n & \Phi^{-1}G & \begin{bmatrix} K^T \\ 0 \end{bmatrix} \\ \hline G\Phi^{-T} & \mathbf{W}_{22} & L^T \\ \hline \begin{bmatrix} K & 0 \end{bmatrix} & L & \mathbf{W}_{33} \end{array}\right] \succeq 0, \quad (31d)$$

$$K \in \mathcal{K}, \quad (31e)$$

$$\mathbf{W}_{33} \in \mathcal{K}^2, \quad (31f)$$

with the parameter set $\{K, L, G, P, M, \mathbf{W}\}$.

Automatic penalty of the trace of W

Lyapunov

Inversion of variables

Exactness: Rank n

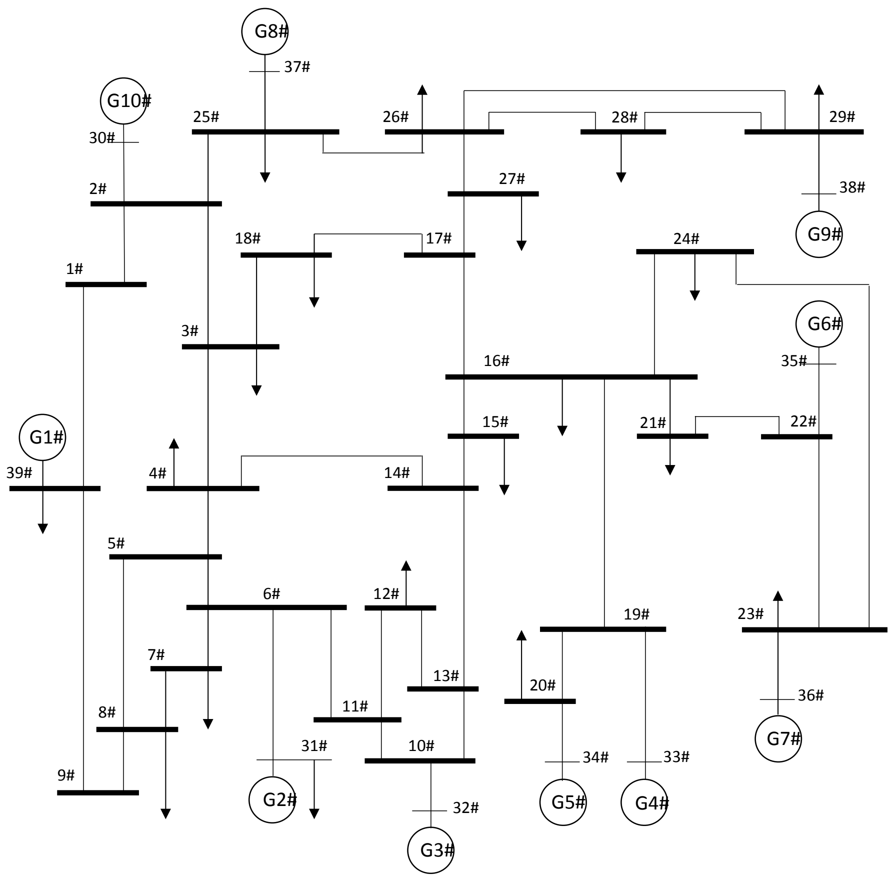
Direct and two-hop pattern

Second Stage of SDP Relaxation

- **Direct Recovery Method:** Recover the controller from the SDP solution.
- **Indirect Recovery Method:** Recover the Lyapunov matrix from the SDP solution and pass it to a second SDP problem to design a controller.
- **Note:** Trace of the to-be low-rank W is penalized in the objective by the noise covariance.
- **Implication:** The higher the noise level, the better the rank enforcement.
- **Theorem:** The relaxation is always exact in the centralized case (= Riccati equations)

IEEE 39 Bus (New England Power System)

- **Maximization of penetration of renewables:** Adjust the mechanical power of each generator based on the angle and frequency of neighboring generators to minimize variations.



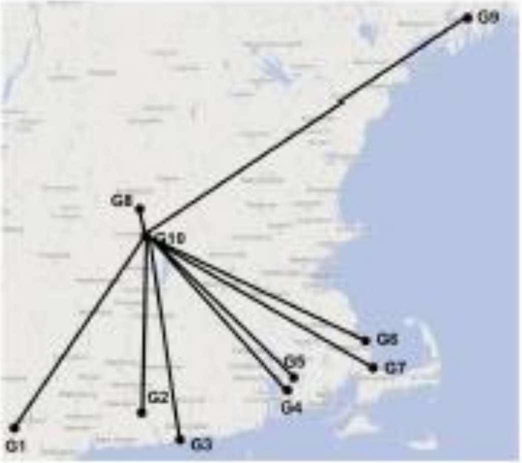
New England System

Four Communication Topologies

(a) Decentralized

(b) Localized

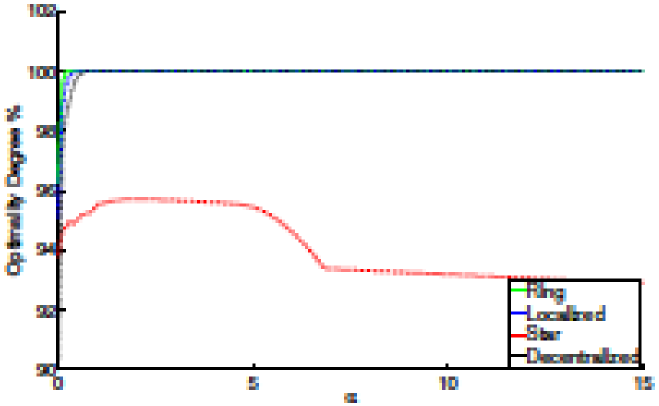
(c) Ring



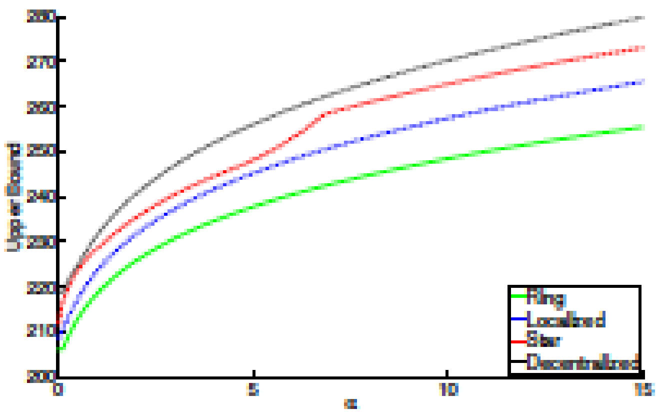
(d) Star Topology (G_{10} in center)

Near-Global Controllers

Various levels of controller gain

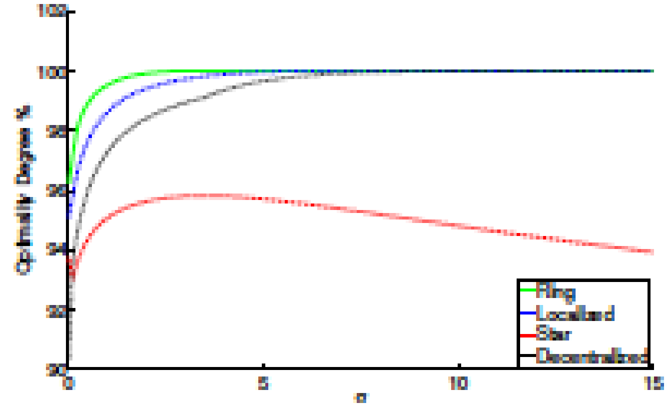


(a) Optimality degree for stochastic ODC

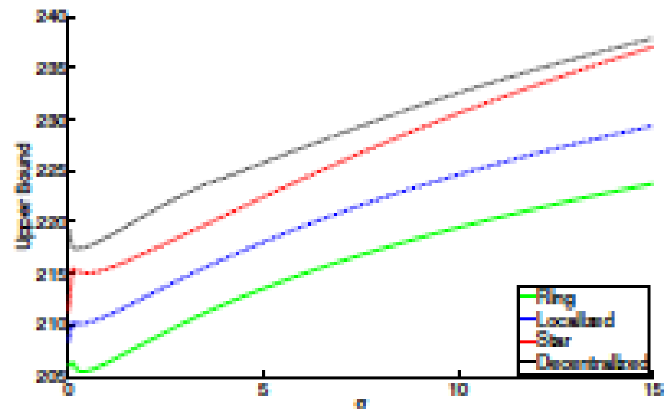


(b) Near-optimal cost for stochastic ODC

Various levels of noise



(a) Optimality degree for stochastic ODC



(b) Near-optimal cost for stochastic ODC

Low-Rank SDP Solution

$$\begin{aligned}
\min_{\mathbf{X}\in\mathbb{S}^n} \quad & \mathrm{trace}\{\mathbf{F}_0\mathbf{X}\} \\
\text{s.t.} \quad & \mathrm{trace}\{\mathbf{F}_k\mathbf{X}\} \leq 0 \quad \text{for} \quad k=1,\ldots,p \\
& X_{11} = 1 \\
& \mathbf{X} \succeq 0
\end{aligned}$$

Real/complex optimization

- Define G as the sparsity graph
- **Theorem:** There exists a solution with rank at most treewidth of *G* +1
- We propose infinitely many optimizations to find that solution.
- This provides a deterministic upper bound for low-rank matrix completion problem.

Polynomial Optimization

$$\text{polynomial optimization} \quad \Longleftrightarrow \quad \text{dense QCQP} \quad \Longleftrightarrow \quad \text{sparse QCQP}$$

- **Vertex Duplication Procedure:**

$$x_i \quad \Longleftrightarrow \quad (x_{i1}, x_{i2}) \quad \text{s.t.} \quad x_{i1} = x_{i2}$$

- **Edge Elimination Procedure:**

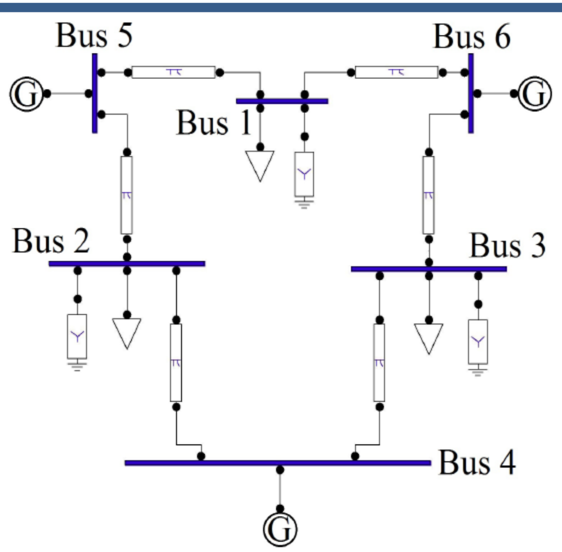
$$x_i x_j \quad \Longleftrightarrow \quad z_1^2 - z_2^2 \quad \text{s.t.} \quad z_1 = \frac{x_i + x_j}{2}, \; z_2 = \frac{x_i - x_j}{2}$$

- This gives rise to a sparse QCQP with a sparse graph.
- The treewidth can be reduced to 2.

Theorem: Every polynomial optimization has a QCQP formulation whose SDP relaxation has a solution with rank 1 or 2.

Conclusions

- ❑ **Focus:** Optimization and control
- ❑ **Goal:** Design efficient algorithms



- ❑ Two thrusts:
  - ❖ Global optimization
  - ❖ Distributed control

- ❑ We have developed two solvers (available on my website).