Graph-Theoretic Algorithm for Arbitrary Polynomial Optimization Problems

Javad Lavaei

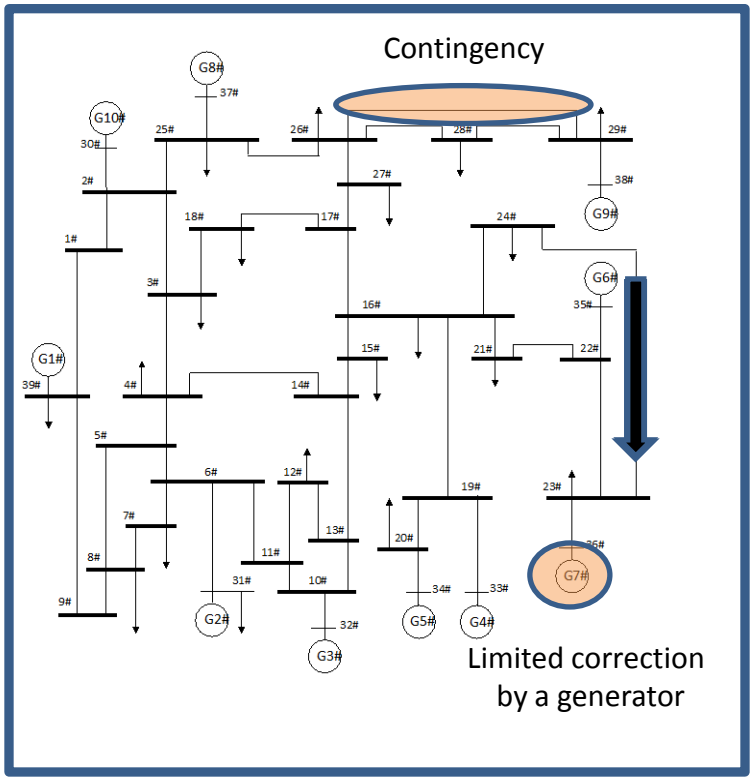
Department of Electrical Engineering
Columbia University

Joint work with:
Ramtin Madani, Ghazal Fazelnia and Abdulrahman Kalbat (Columbia University)
Somayeh Sojoudi (New York University)
Ross Baldick (University of Texas-Austin)

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
  - Robustness: Make the system robust and resilient → Constraint identification

Constraint Identification: Contingency Analysis



- **Contingency Analysis:**
  - Assume a line is disconnected.
  - Many generators cannot change productions quickly.
  - The flows over other lines would increase.
  - This triggers a cascading failure.

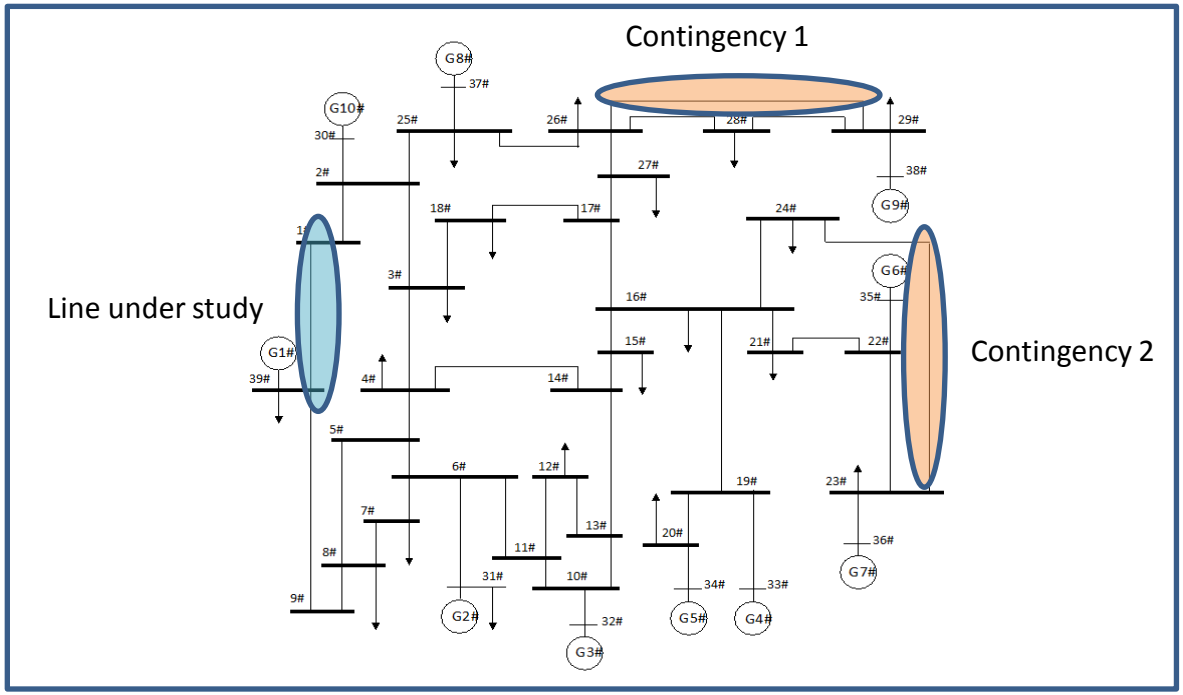
- **N-1 contingency:** Design the operating point such that the flow limits are satisfied under every single fault.

- **Challenge:** Number of constraints on the order of l^2.

- **Example:** Number of contingency constraints for Polish system with ~3300 buses = ~17 millions

Contingency Analysis

- **Bad news:** There are l^2 contingency constraints.
- **Good news:** Just n of them would be binding.
- **Strategy:** Identify constraints that are implied by others and then remove them.

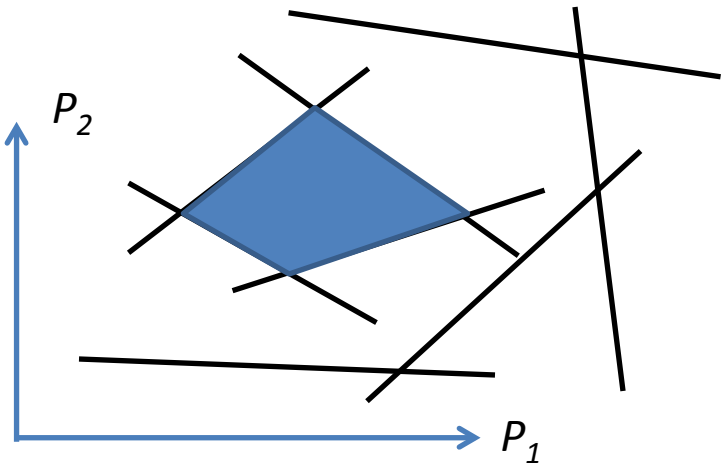


1- Constraint for line *(i,j)* under normal conditions
2- Constraint for line *(i,j)* under contingency 1

3- Constraint for line *(i,j)* under contingency 2

Contingency Analysis

- **Parameters:** $P_1, P_2, \ldots, P_n$
- **Constraints:** l^2 linear constraints in terms of $P_1, P_2, \ldots, P_n$.
- **Feasible injection region:**
- Some constraints do not define a face.
- **Observation:** A majority of constraints are not important because:



Normal Rating < Long-term emergency < Short-term emergency

- **Main idea behind our geometric approach:**
  - Mapping 1: injections ----> flows with no fault
  - Mapping 2: injections ----> flows with one specific fault
  - Transformation from mapping 1 to mapping 2 is not too far from identity.

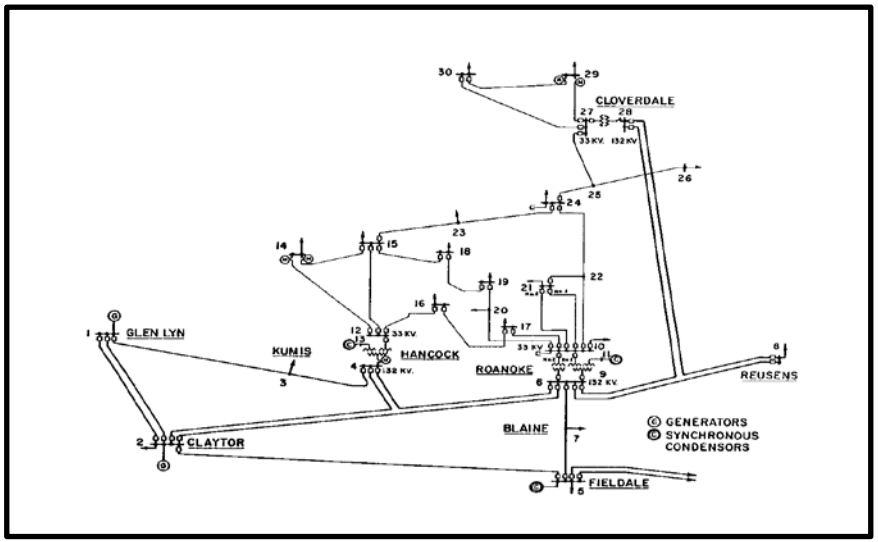
Contingency Analysis

- Many constraints are always unimportant, independent of the on/off status of generators, production levels, and load profiles.

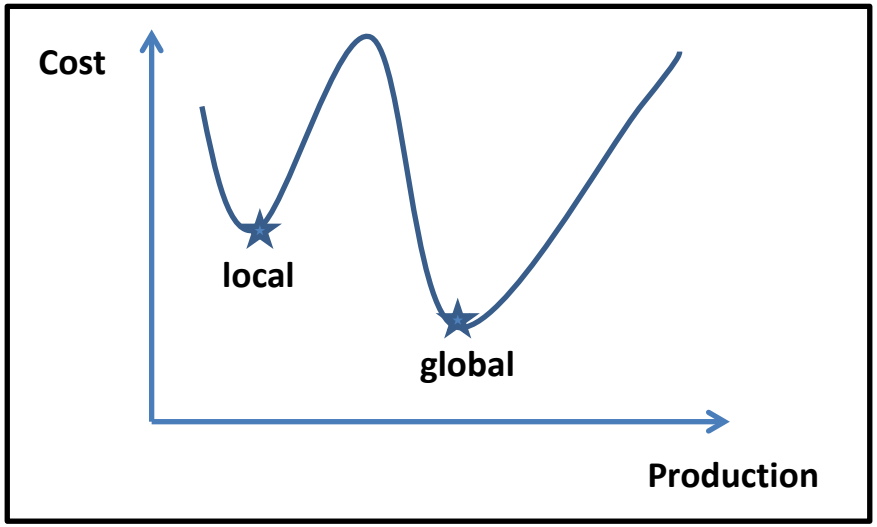
System	Total constraints	Important constraints	Ratio
Case 14	400	95	0.2375
Case 30	1681	271	0.1612
Case 39	2116	253	0.1195
Case 57	6400	642	0.1003
Case 118	34596	934	0.0269
Case 300	168921	2492	0.0148

Optimization

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

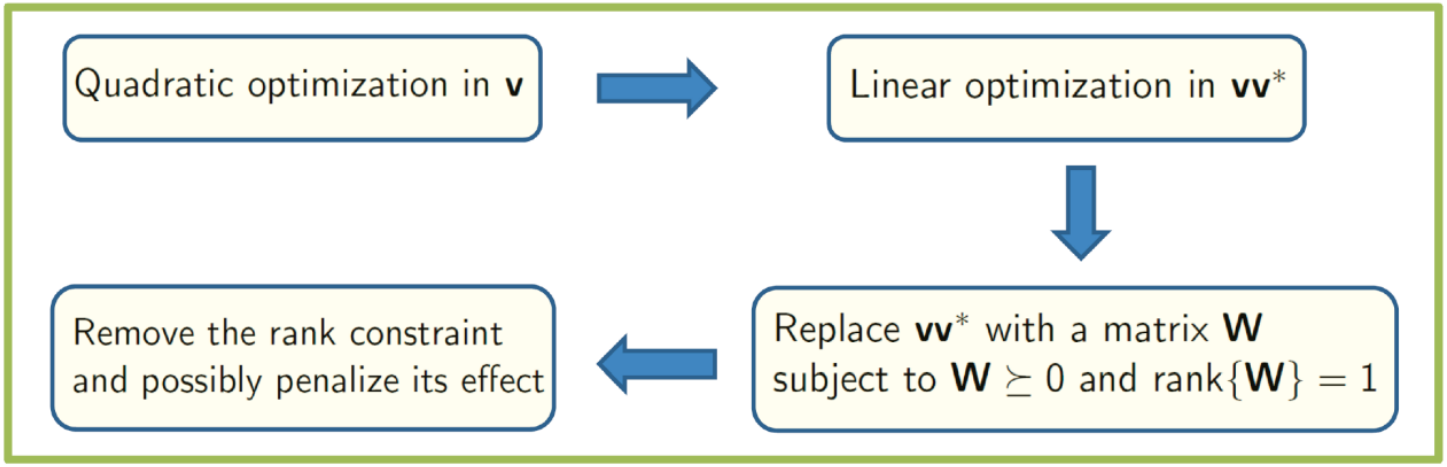


- **Issue of non-convexity:**
  - Discrete parameters
  - Nonlinearity in continuous variables



- 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.

Penalized Semidefinite Programming (SDP) Relaxation



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

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

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

Optimal Power Flow

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

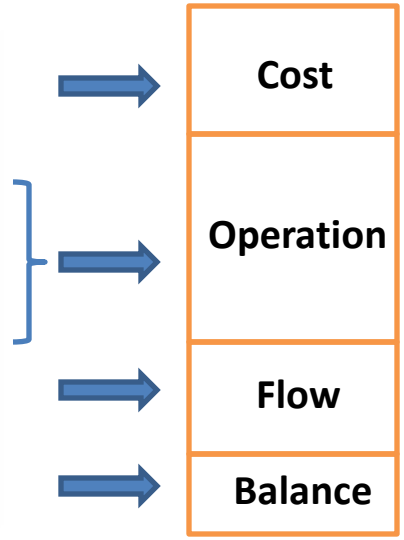
$$\text{Subject to} \quad P_k^{\min} \le P_{G_k} \le P_k^{\max} \tag{1b}$$

$$Q_k^{\min} \le Q_{G_k} \le Q_k^{\max} \tag{1c}$$

$$V_k^{\min} \le |V_k| \le V_k^{\max} \tag{1d}$$

$$\mathrm{Re}\{V_l(V_l - V_m)^* y_{lm}^*\} \le P_{lm}^{\max} \tag{1e}$$

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



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

- **Observation:** SDP relaxation works for almost all benchmark examples and several data sets for US and Europe.

- **Theory:** Lots of theories to support this observation.

Penalized SDP Relaxation

- **What if we get a low-rank but not rank-1 solution?**

- Penalized SDP relaxation:

$$\sum_{k\in\mathcal{G}} f_k(P_{G_k}) \quad \Longrightarrow \quad \sum_{k\in\mathcal{G}} f_k(P_{G_k}) + \varepsilon \sum_{k\in\mathcal{G}} Q_{G_k}$$

- Penalized SDP relaxation aims to find a near-optimal solution.

- It worked for IEEE systems with over 7000 different cost functions.

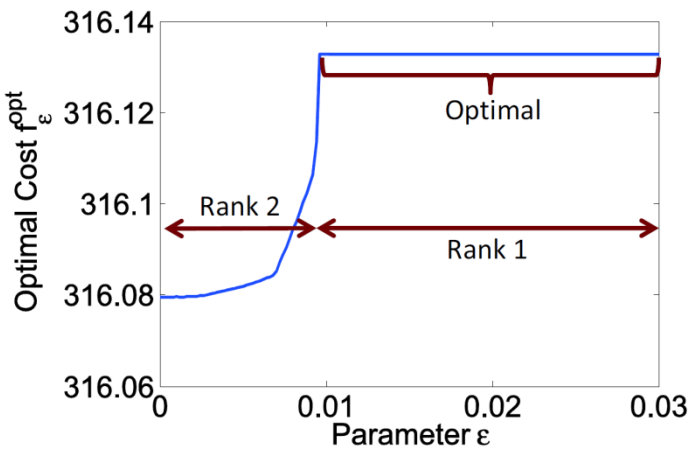
- Near-optimal solution coincided with the IPM's solution in 100%, 96.6% and 95.8% of cases for IEEE 14, 30 and 57-bus systems.

Penalized SDP Relaxation

- Let λ_1 and λ_2 denote the two largest eigenvalues of *W*.

IEEE-14		
ϵ	0	0.012
λ_1	15.1617	15.1340
λ_2	0.0138	0
Cost	\$316.08	\$316.13

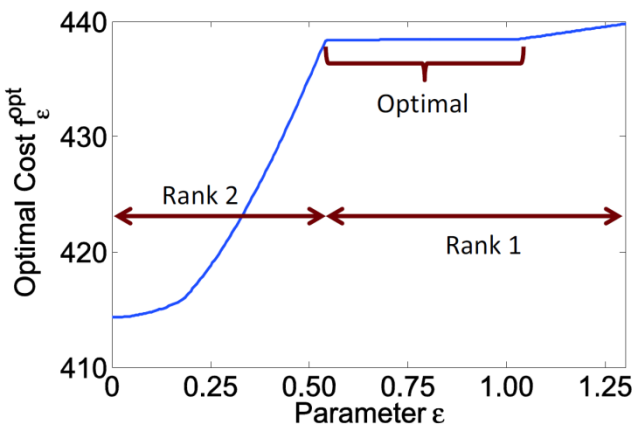
k	c_k	P_{G_k}	Q_{G_k}	P_{G_k}	Q_{G_k}
1	3	25.36	0	25.38	0.85
2	1	140	25.44	140	22.25
3	4	0	28.77	0	27.11
6	1	100	-6	100	-6
8	4	0	9.16	0	6.42



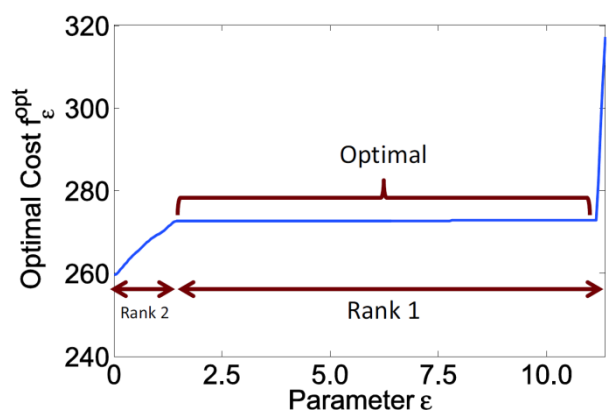
- Correction of active powers is negligible but reactive powers change noticeably.
- There is a wide range of values for ε giving rise to a nearly-global local solution.

Penalized SDP Relaxation

IEEE-30					
ϵ	0			0.55	
λ_1	30.6789			30.8677	
λ_2	0.4986			0	
Cost	\$414.34			\$438.40	
k	c_k	P_{G_k}	Q_{G_k}	P_{G_k}	Q_{G_k}
1	1	80	11.11	80	-4.60
2	10	0	39.16	0	-2.10
13	1	40	44.70	40	44.70
33	10	23.98	35.26	27.32	33.36
23	100	0	33.39	0	15.62
27	1	54.55	25.65	45.22	21.33

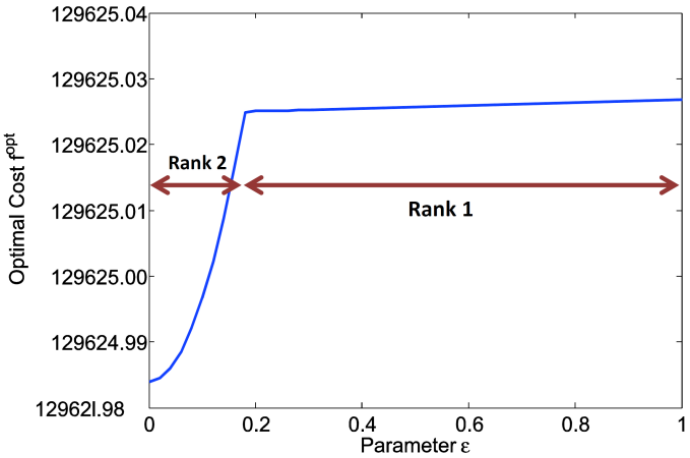


IEEE-57					
ϵ	0			1.5	
λ_1	57.1776			56.8887	
λ_2	0.0767			0	
Cost	\$259.70			\$272.73	
k	c_k	P_{G_k}	Q_{G_k}	P_{G_k}	Q_{G_k}
1	0.1	575.88	78.60	575.88	111.87
2	0.1	100	50	100	50
3	100	0	60	0	44.29
6	0.1	100	25	100	25
8	10	13.11	117.90	14.41	159.64
9	0.1	100	9	100	9
12	0.1	410	96.91	410	-6.29



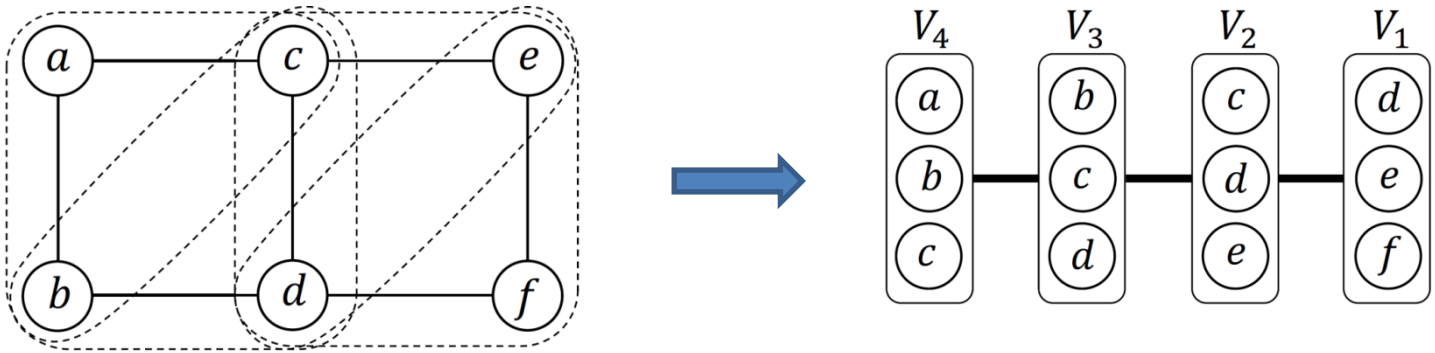
Example with Multiple Solutions

- **Example borrowed from Bukhsh et al.:**
  - Modify IEEE 118-bus system has 3 local solutions with the optimal costs 129625.03, 177984.32 and 195695.54.
  - Our method finds the best one.



Treewidth

- **Tree decomposition:**



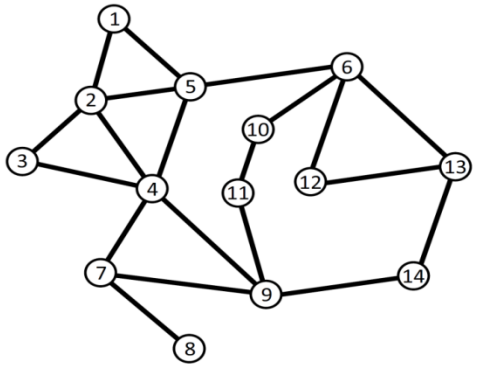
- We map a given graph *G* into a tree *T* such that:
  - Each node of *T* is a collection of vertices of *G*
  - Each edge of *G* appears in one node of *T*
  - If a vertex shows up in multiple nodes of *T*, those nodes should form a subtree

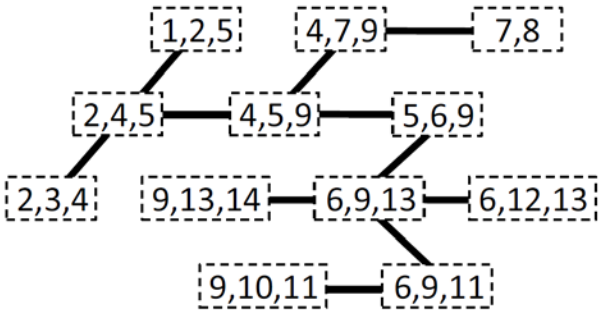
- **Width of a tree decomposition:** The cardinality of largest node minus one

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

Power Networks

- Treewidth of a tree: 1
- How about the treewidth of IEEE 14-bus system with multiple cycles? 2





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	29
New England 39-bus	3	Polish 2746wop	29
IEEE 57-bus	5	Polish 3012wp	25
IEEE 118-bus	4	Polish 3120sp	26
IEEE 300-bus	6	Polish 3375wp	28

Power Networks

$$\begin{aligned} \min \quad & \operatorname{trace}\{\mathbf{F}_0\mathbf{X}\} \\ \text{s.t.} \quad & \operatorname{trace}\{\mathbf{F}_k\mathbf{X}\} \leq 0 \quad \text{for} \quad k = 1, \ldots, p \\ & X_{11} = 1 \\ & \mathbf{X} \succeq 0 \end{aligned}$$

Real/complex optimization

- ❑ Define G as the sparsity graph
- ❑ **Theorem:** There exists a solution with rank at most treewidth of *G* +1
- ❑ We proposed infinitely many optimizations to find that solution.

❑ **Technique:**

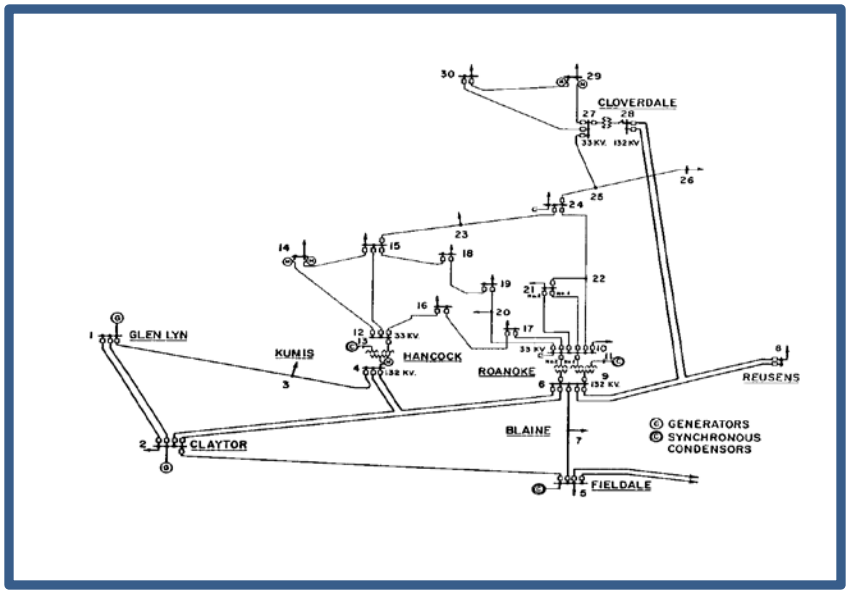
- ❖ Choose a set of edges *E* from the sparsity graph through our notion of enriched graph.
- ❖ For every *(i,j)* in *E*, add a penalty term $t_{ij}X_{ij}$ to the objective, where t_{ij} is arbitrary (nonzero).
- ❖ All solutions of the penalized problem have rank at most treewidth of *G* +1.

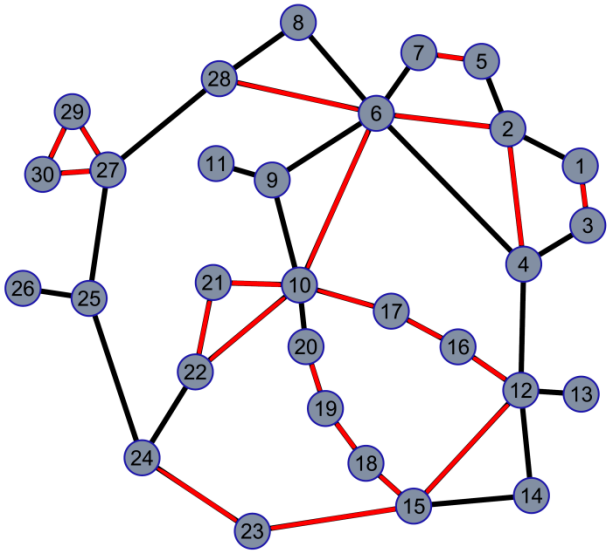
Examples

- **Example:** Consider the security-constrained unit-commitment OPF problem.
- Use SDP relaxation for this mixed-integer nonlinear program.
- What is the rank of $\boldsymbol{X}^{opt}$?
  1. **IEEE 300-bus system:** rank ≤ 6
  2. **Polish 3120-bus system:** Rank ≤ 27
- Use penalization to make them rank-1.

- We have written a solver in Java and MATLAB (using CVX and MOSEK) to find a near-global solution.
- Computation time for penalized SDP relaxation of ACOPF for Polish System with ~3300 buses (over 9,000,000 parameters): ~2.4 minutes.

Security-Constrained OPF for IEEE 30-Bus System





- A system with 41 lines and 21 contingencies (red lines)
- Optimal cost of OPF: 576.89
- Optimal cost of convex relaxation of SCOPF: 579.09
- Near-optimal cost of SCOPF for the penalty factor 0.1: 579.10

Polynomial Optimization

polynomial optimization $\iff$ dense QCQP $\iff$ sparse QCQP

- **Vertex Duplication Procedure:**

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

- **Edge Elimination Procedure:**

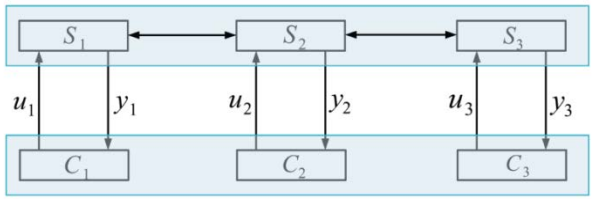
$$x_i x_j \iff z_1^2 - z_2^2 \quad \text{s.t.} \quad z_1 = \frac{x_i + x_j}{2}, \; z_2 = \frac{x_i - x_j}{2}$$

- This gives rise to a sparse QCQP with a sparse graph.

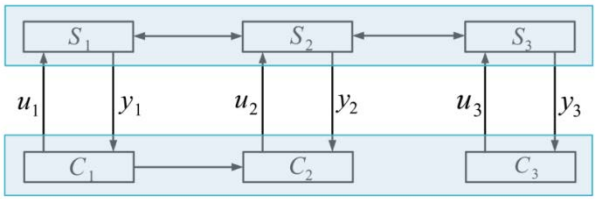
- The treewidth can be reduced to 2.

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

Distributed Control



Decentralized control



Distributed control

- Consider the time-varying system: $\begin{cases} x[\tau+1] = A[\tau]x[\tau] + B[\tau]u[\tau] \\ y[\tau] = C[\tau]x[\tau] \end{cases}$

- The goal is to design a structured controller $u[\tau] = Ky[\tau]$ to minimize

$$\sum_{\tau=0}^{p} \left(x[\tau]^T Q[\tau] x[\tau] + u[\tau]^T R[\tau] u[\tau] \right) + \mu \ \mathrm{trace}\{KK^T\}$$

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

Two Quadratic Formulations in Static Case

- **Formulation in time domain:**
  - Stack the free parameters of K in a vector h.
  - Define **v** as:

$$\mathbf{v} = \begin{bmatrix} 1 & h^* & x[0]^* & x[1]^* & \cdots & x[p]^* & y[0]^* & \cdots y[p]^* & u[0]^* & \cdots & u[p]^* \end{bmatrix}^*$$

- **Formulation in Lypunov domain:**
  - Consider the BMI constraint: $\begin{bmatrix} P & P(A+BKC)^* \\ (A+BKC)P & P \end{bmatrix} \succ 0$
  - Define **v** as:

$$v = \begin{bmatrix} 1 & h^* & P_{11} & P_{12} & \cdots & P_{1n} & \cdots & P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix}^*$$

Recovery of Near-Optimal Controller

- The proposed SDP relaxation was obtained from a big vector.

- **Computationally cheap SDP relaxation:** Define a relaxation in terms of a small super-vector.

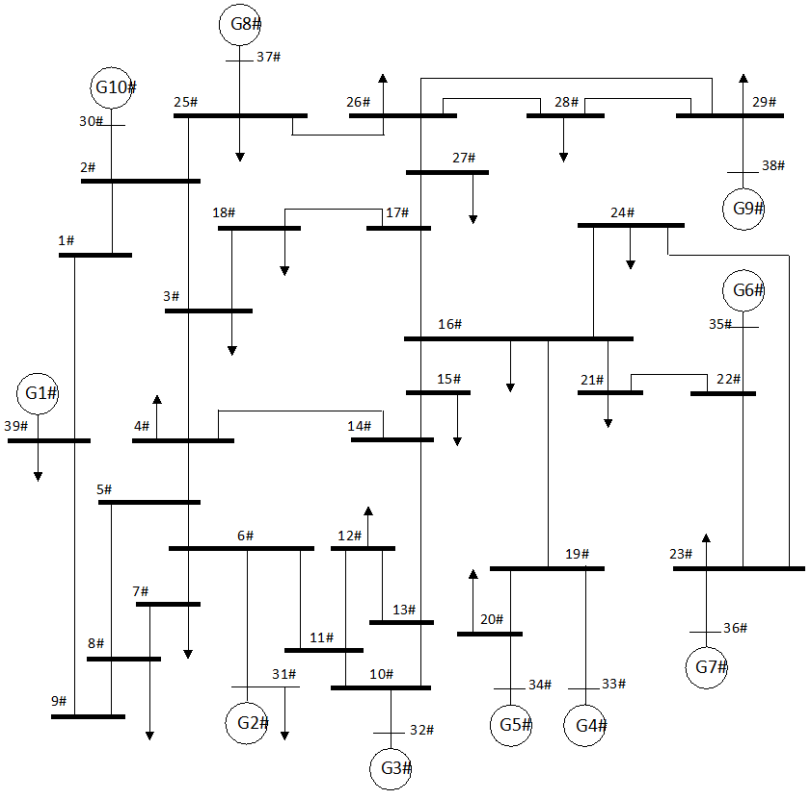
- **Strategy in time domain:**
  - Solve a computationally cheap SDP relaxation.
  - Recover the state parameters from the SDP matrix.
  - Solve a second optimization to recover the controller.

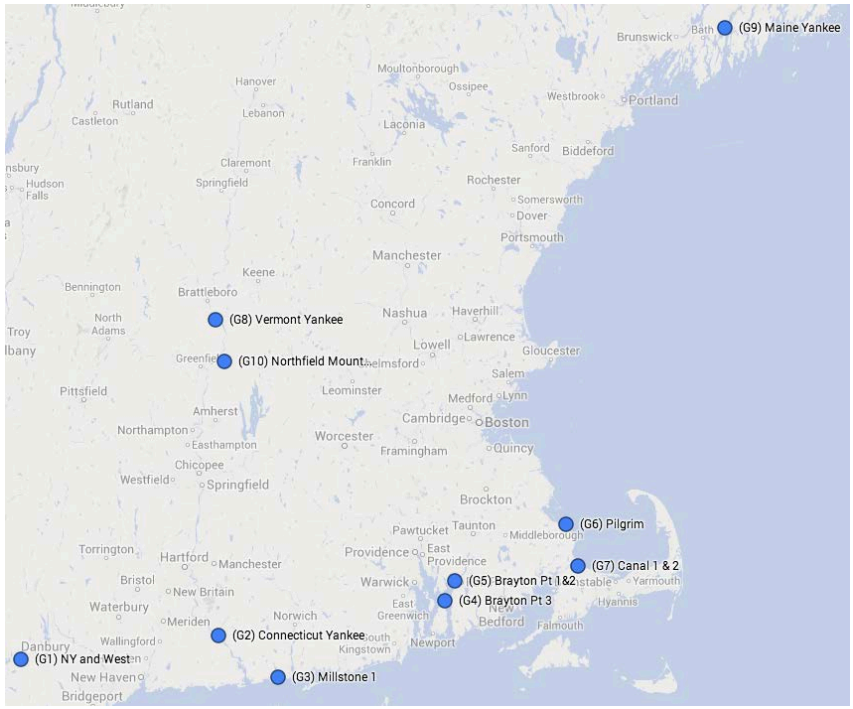
1	h^T	x^T	u^T	y^T
h	hh^T	hx^T	hu^T	hy^T
x	xh^T	xx^T	xu^T	xy^T
u	uh^T	ux^T	uu^T	uy^T
y	yh^T	yx^T	yu^T	yy^T

Red stripes: wrong information

- **Strategy in Lyapunov domain:**
  - Solve a computationally cheap "penalized" SDP relaxation.
  - Recover the Lyapunov matrix from the SDP matrix.
  - Solve a second optimization to recover the controller.

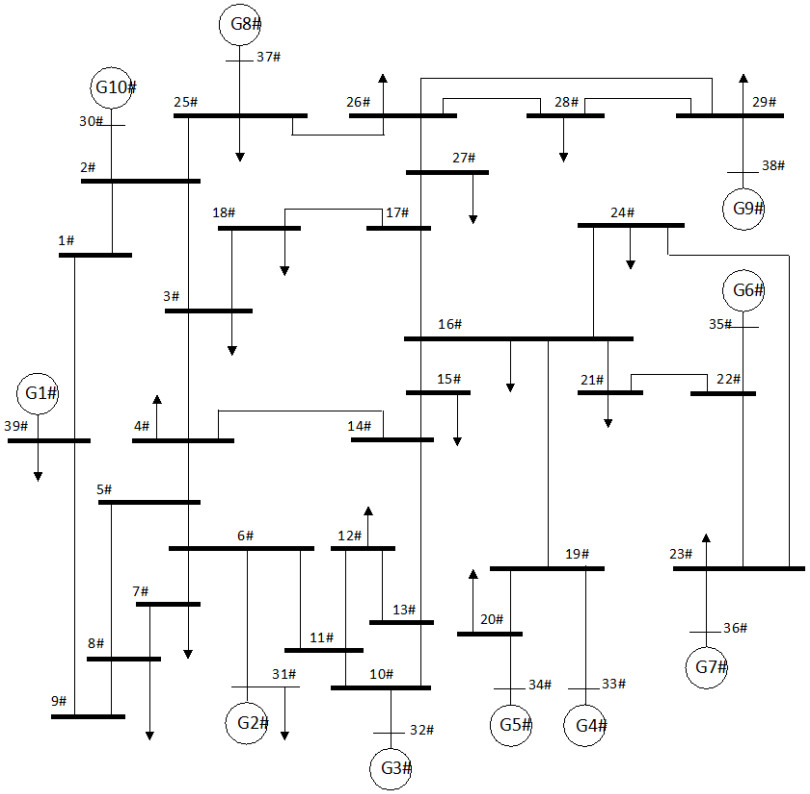
IEEE 39 Bus (New England Power System)

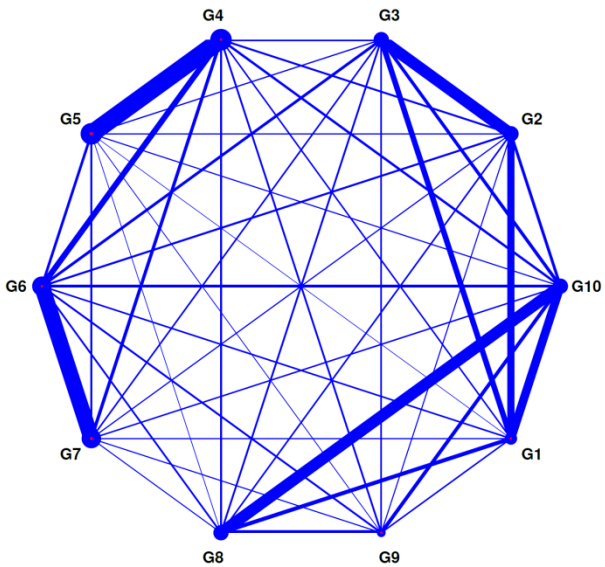




- **Problem:** Adjust the mechanical power of each generator based on the angle and frequency of neighboring generators in an optimal way (discretization time: 0.05 sec).

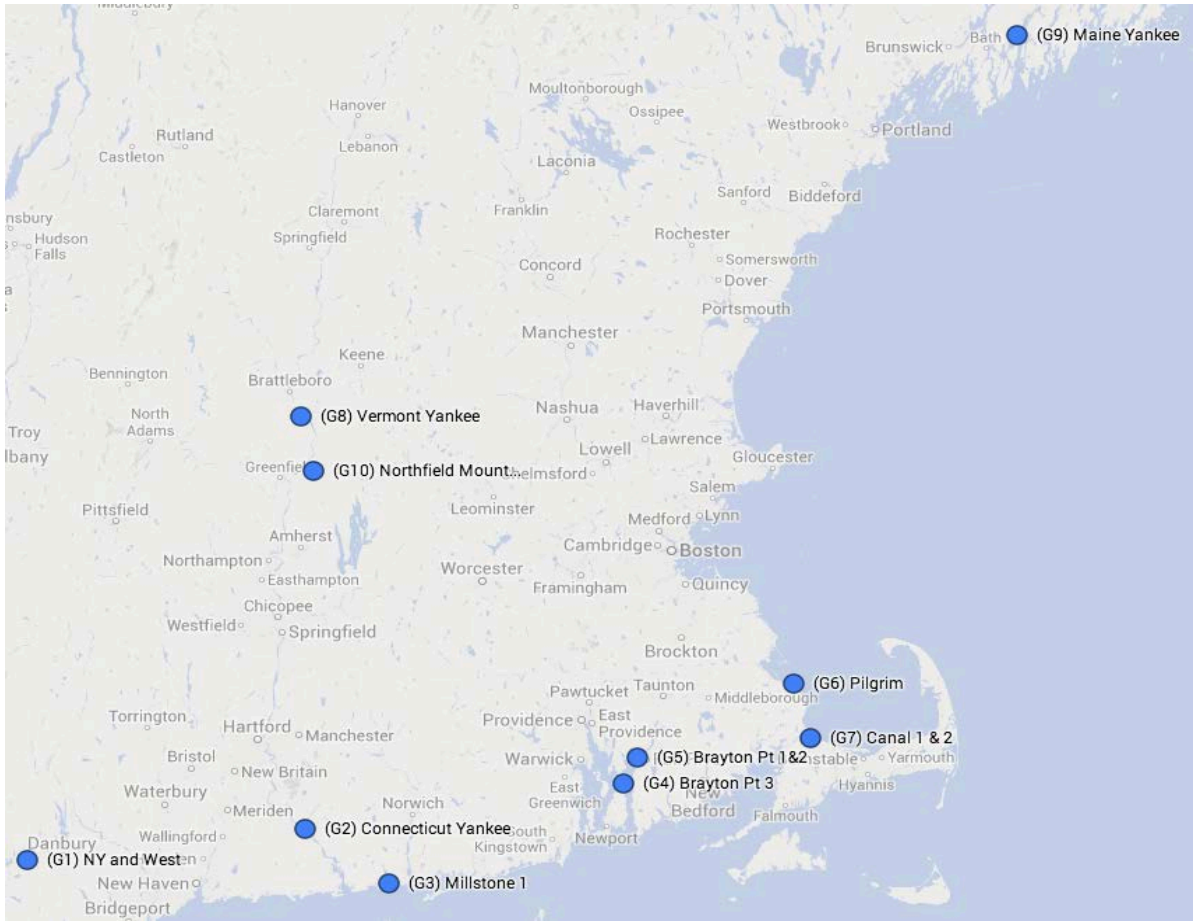
IEEE 39 Bus (New England Power System)





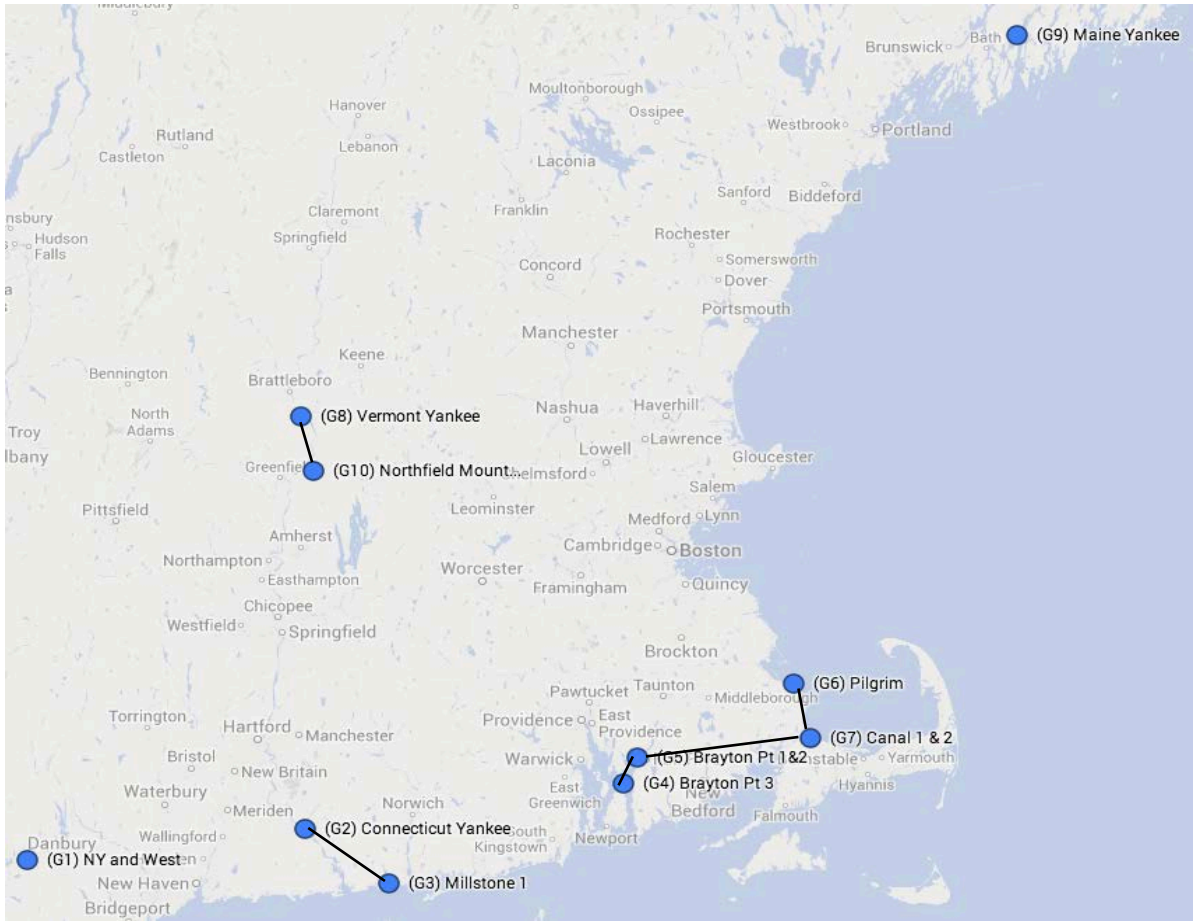
- Using Kron reduction, we look at the interaction among 10 generators.

Decentralized Communication Topology



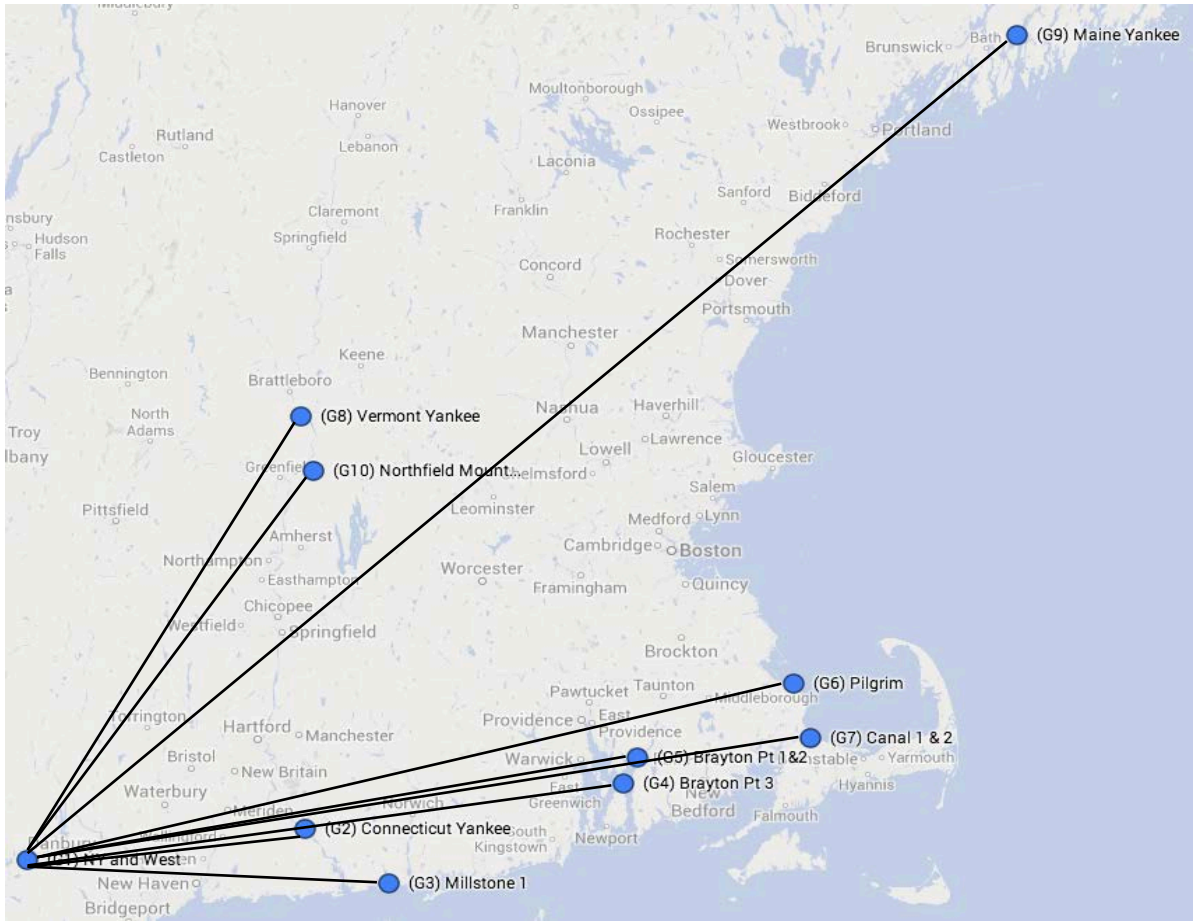
Penalty: 61000	Performance loss: 6.86%

Localized Communication Topology



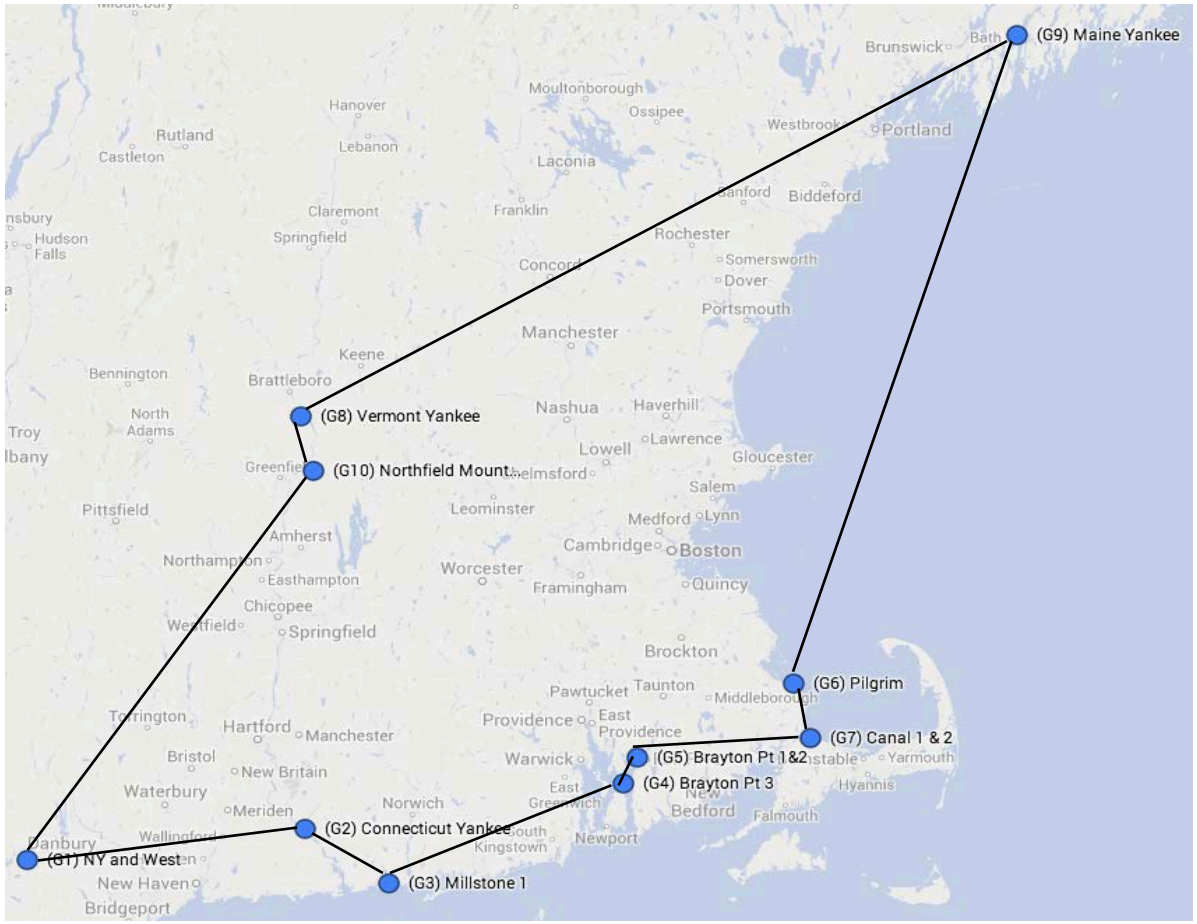
Penalty: 15000	Performance loss: 6.08%

Star Communication Topology



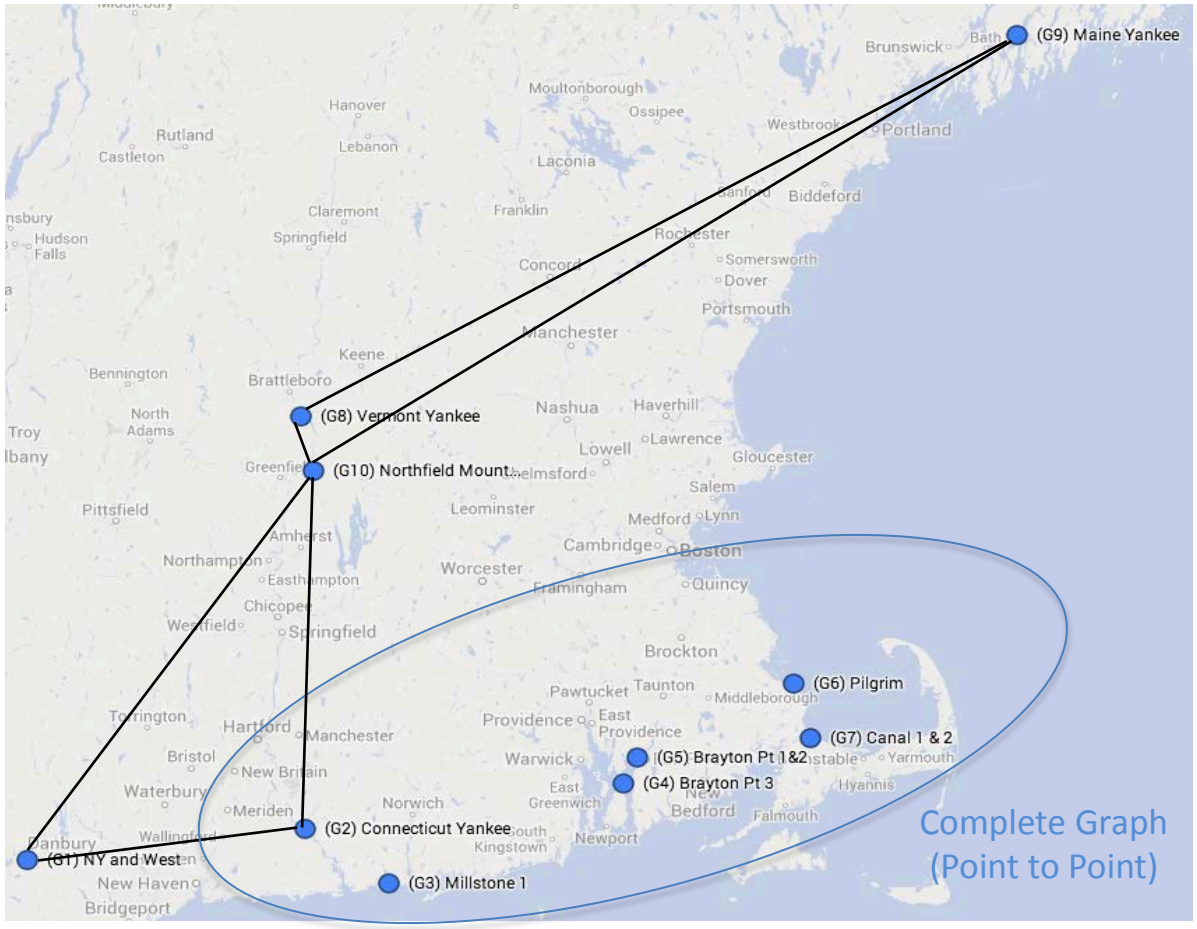
Penalty: 2300	Performance loss: 0.22%

Ring Communication Topology



Penalty: 670	Performance loss: 0.19%

Another Communication Topology



Penalty: 30	Performance loss: 0.009%

Four Communication Topologies

- **Finite horizon problem over 80 samples:**

Structure	Lower Bound	Upper Bound	Normalized Gap
1 (Fully Decentralized)	316.28	320.27	0.0125
2 (Localized)	316.28	319.62	0.0105
3 (Star Topology)	316.28	317.31	0.0033
4 (Complete Subgraphs)	316.28	316.55	0.0009

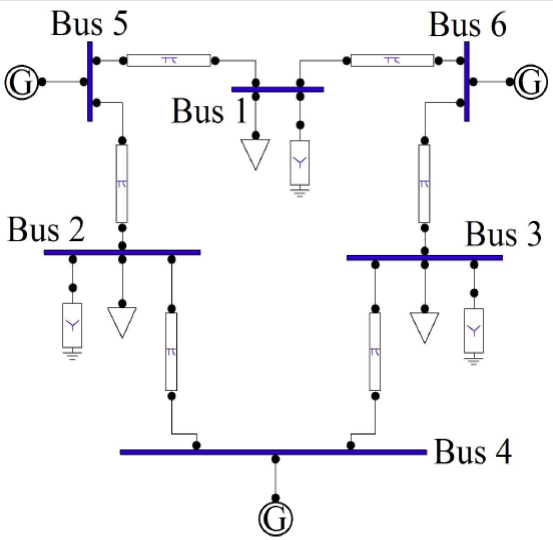
- Solution for the star communication topology:

Bus #	30	31	32	33	34	35	36	37	38	39
30	-0.21	1.04	-1.80	-0.70	3.88	2.17	-0.70	-2.08	-0.29	-1.41
31	-1.08	0.86								
32	-0.83		0.63							
33	-1.52			1.37						
34	-1.48				1.33					
35	-0.64					0.43				
36	-0.78						0.64			
37	-0.93							0.74		
38	-0.57								0.41	
39	0.37									-0.57

	30	31	32	33	34	35	36	37	38	39
30	2.20	1.90	-2.64	3.32	-1.97	-1.59	-0.59	0.58	-0.02	-1.75
31	-0.24	-0.37								
32	-0.28		-0.35							
33	-0.49			-0.17						
34	-0.49				-0.14					
35	-0.63					0.05				
36	-0.42						-0.21			
37	-0.28							-0.33		
38	-0.39								-0.19	
39	-0.46									-0.10

Conclusions

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



- Three thrusts:
  - Global optimization
  - Distributed control
  - Contingency analysis
- We have written two solvers to be posted online soon.