Scalable optimization of complex energy systems using high-performance computing

Cosmin G. Petra

Mathematics and Computer Science Division

Argonne National Laboratory (ANL)

Joint work with Naiyuan Chiang, Shri Abyankhar, Victor Zavala, Mihai Anitescu (ANL)

January 6, 2015

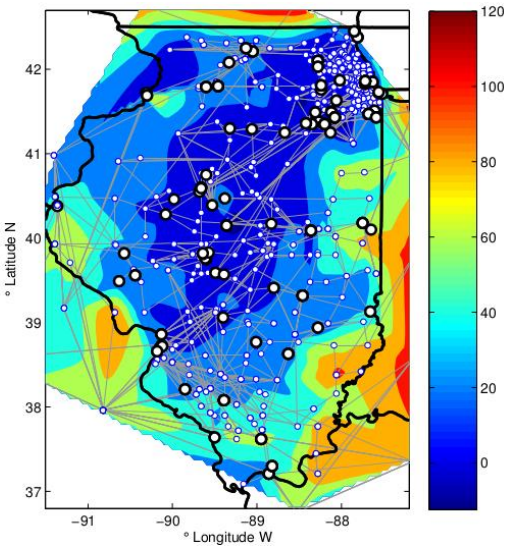
Outline

- Motivating applications
  - stochastic optimal power flow
  - optimal power flow with dynamic security constraints
- High-performance optimization solvers
- Computational simulations

Economic dispatch models in power grid

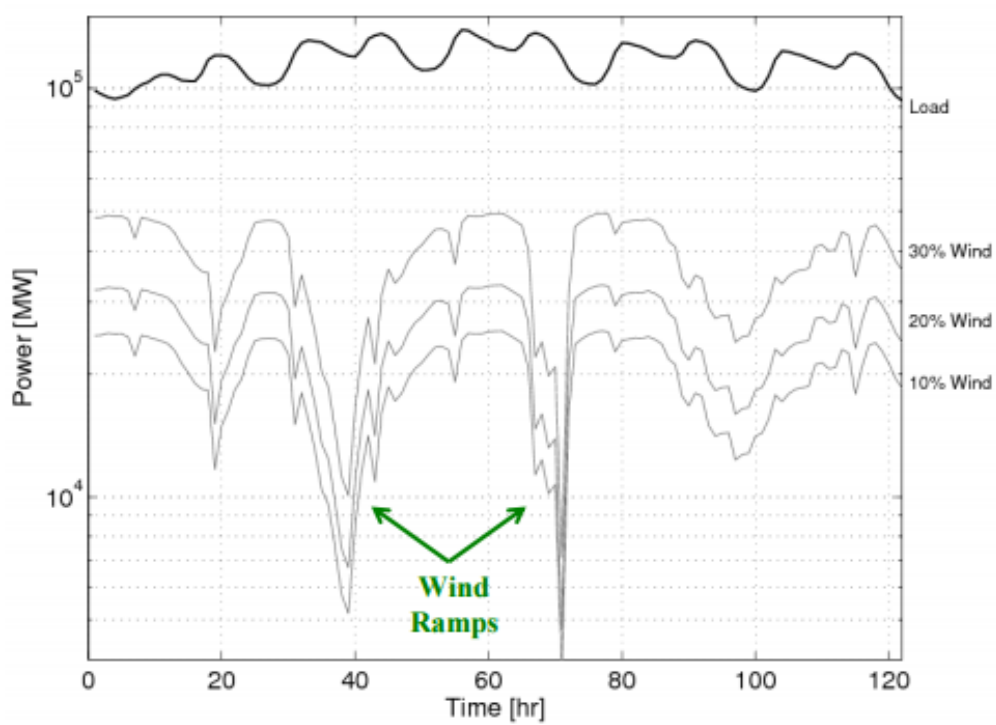
- Basis for the electricity distribution and electricity market

$$
\begin{aligned}
\min_{x,f} \quad & \sum_{i \in G} c_i x_i \\
\text{subj.to:} \quad & \tau_n(f) + \sum_{i \in T(n)} x_i = d_n, \forall n \in N \\
& f \in U \\
& x_i \in C_i, \forall i \in G
\end{aligned}
$$



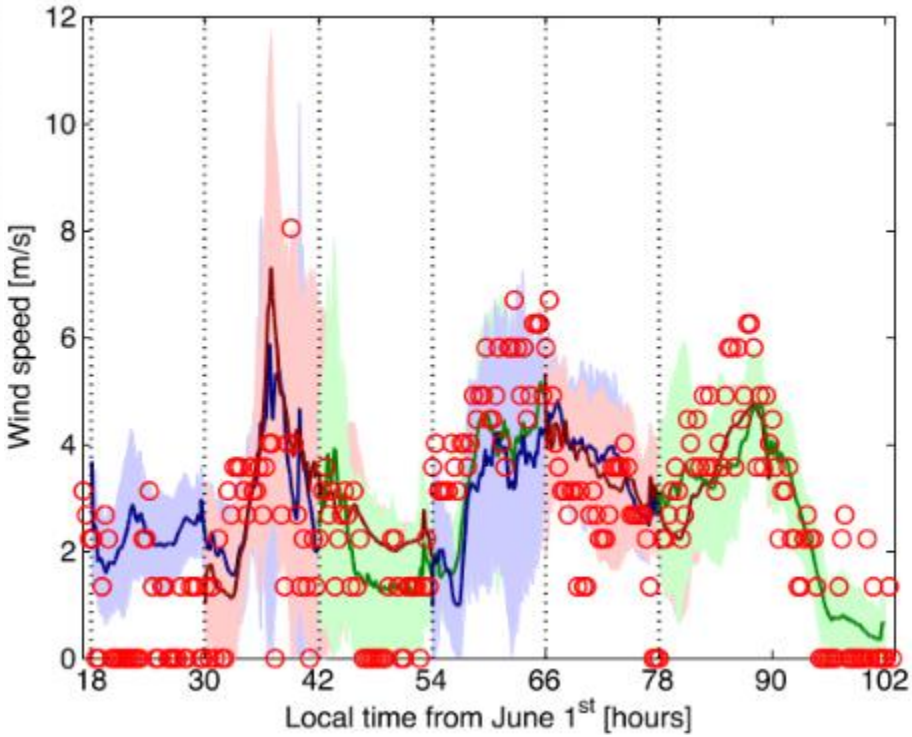
- Answers critical questions such as:
  - What is the cheapest way to ramp-up generation to satisfy an increase in demand given the grid transmission limits imposed?
  - What are the electricity prices at each demand node given a certain demand?

Electricity dispatch under uncertainty



The sharp drops in wind power need to be forecasted well in advanced to give the thermal generators enough time to ramp up generation.

Wind forecasting results in wind scenarios, requiring stochastic optimization



Wind/solar (stochastic) economic dispatch model

$$\min_{x,y(\omega),f,F(\omega)} \quad \sum_{i\in G} c_i x_i + \mathbb{E}_\omega \sum_{i\in G} p_i |x_i - y_i(\omega)|$$

$$\text{subj.to:} \quad \tau_n(f) + \sum_{i\in T(n)} x_i = d_n, \forall n \in N$$

$$\tau_n(F(\omega)) + \sum_{i\in T(n)} y_i(\omega) = d_n, \forall n \in N, \omega \in \Omega$$

$$f, F(\omega) \in U, \forall \omega \in \Omega$$

$$x_i, y_i(\omega) \in U_i, \forall i \in G, \omega \in \Omega$$

Ω is the set of **weather scenarios**

x_i	-	generation
f	-	vector of line flows
$\tau_n(f)$	-	net power imported at node n
N	-	set of network nodes
d_n	-	demand at node n
$T(n)$	-	set of generators at node n
U	-	line capacity constraints
G	-	set of generators (coal, gas, nuclear, wind, solar)
C_i	-	generation constraints
c_i	-	generation costs

Dynamically secure optimal power flow

- **Optimal power flow** (OPF) is the *economic model*.
  - cost-optimal electricity generation subject to generation and transmission constraints
  - steady-state model accounts only for gradual (slow) changes

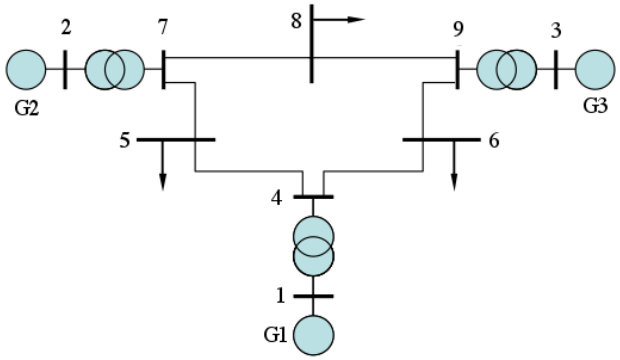
- **Transient stability (TS)** analysis studies *power system security* following a major disturbance: loss of lines, generators, drastic changes in load, etc.

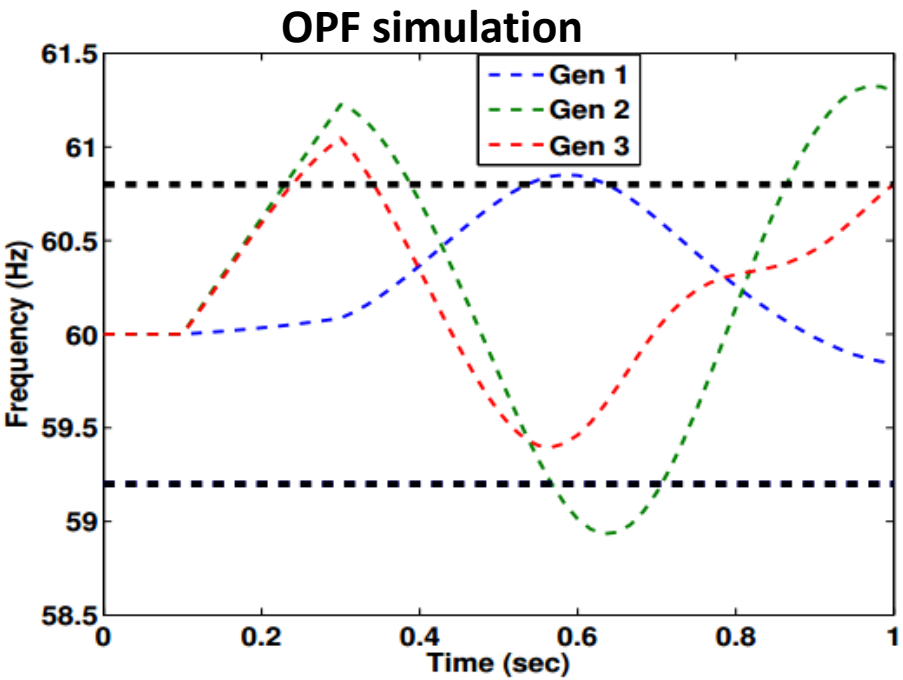
- TS and OPF are widely used in power grid operations by the balancing authorities and transmission controllers, however as **separate procedures**.

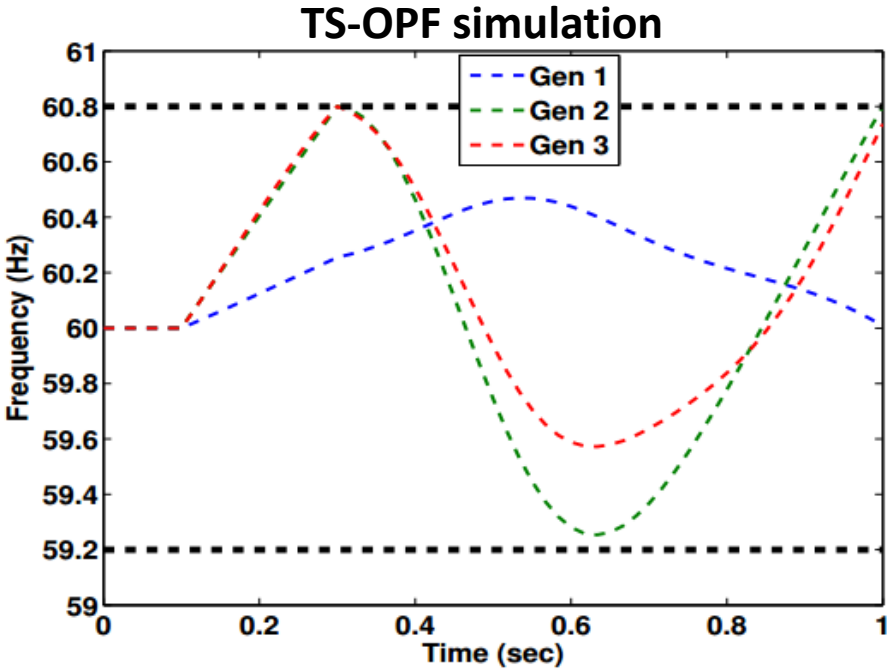
- **Transient stability optimal power flow** (TSOPF) ensures cost-efficiency and dynamical security

Case study – frequency stability under line fault

- The simulation window is t=0 to t=1 second
- Fault at Bus 7 from t=0.1 to t=0.3 seconds
- Safe limits for frequency: [59.2 Hz, 60.8 Hz]







S. Abhyankar, V. Rao, and M. Anitescu, *Dynamic Security Constrained Optimal Power Flow using Finite Difference Sensitivities*. Proceedings of IEEE PES General Meeting, July 2014.

Constitutive equations for TS

Generator equations

$$
\begin{aligned}
T'_{doi}\frac{dE'_{qi}}{dt} &= -E'_{qi} - (X_{di} - X'_{di})I_{di} + E_{fdi}, \\
T'_{qoi}\frac{dE'_{di}}{dt} &= -E'_{di} - (X_{qi} - X'_{qi})I_{qi} \\
\delta_i &= \omega - \omega_s \\
\frac{2H_i}{\omega_s}\frac{d\omega_i}{dt} &= T_{Mi} - E'_{di}I_{di} - (X'_{qi} - X'_{di})I_{di}I_{qi} - D_i(\omega_i - \omega_s) \\
T_{Ei}\frac{dE_{fdi}}{dt} &= -(K_{Ei} + S_{Ei}(E_{fdi}))E_{fdi} + V_{Ri} \\
T_{Fi}\frac{dR_{fi}}{dt} &= -R_{fi} + \frac{K_{Fi}}{T_{Fi}}E_{fdi} \\
T_{Ai}\frac{dV_{Ri}}{dt} &= -V_{Ri} + K_{Ai}R_{fi} - \frac{K_{Ai}K_{Fi}}{T_{Fi}}E_{fdi} + K_{Ai}(V_{refi} - V_i) \\
0 &= E'_{di} - V_d - R_sI_{di} + X'_{qi}I_{qi} \\
0 &= E'_{qi} - V_q - R_sI_{qi} - X'_{di}I_{di}
\end{aligned}
$$

(7+2 eqns per generator)

Current balance

$$0 = I_{gen} - YV - I_{load}$$

(2 eqns per bus)

Frequencies stabilization: $\underline{\omega} = 59.2\text{Hz} \leq \omega(t,p) \leq \bar{\omega} = 60.8\text{Hz}, \forall t \in [t_0, t_f] \Longleftrightarrow H(p) = 0$

$$H(p) = \int_{t_0}^{t_f} \left[(\omega(t,p) - \underline{\omega})_- + (\omega(t,p) - \bar{\omega})_+\right]^2 dt = 0$$

TSOPF with multiple contingencies

- Dynamically secure cost-optimal dispatch under **multiple contingency scenarios**
- Mathematically formulated as

$$
\begin{aligned}
\min \quad & C(p) \\
\text{s.t.} \quad & g_s(p) = 0 \\
& h_s(p) \leq 0 \\
& p^- \leq p \leq p^+ \\
& \underbrace{H_c(p) = 0,}_{\text{TS: simulation-based with adjoint sensitivities}} \quad c \in \mathcal{C}
\end{aligned}
$$

OPF: *algebraically (closed-form) described*

- Decomposition patterns similar to stochastic optimization
  - complicating computational features come from the simulation-based constraints

Stochastic programming

- Two-stage stochastic programming with recourse ("here-and-now")

$$\underset{x_0}{Min}\left\{ f_0(x_0) + \mathbb{E}\left[\underset{x}{Min}\ f(x,\omega) \right] \right\}$$

subj. to. $A_0 x_0 = b_0$

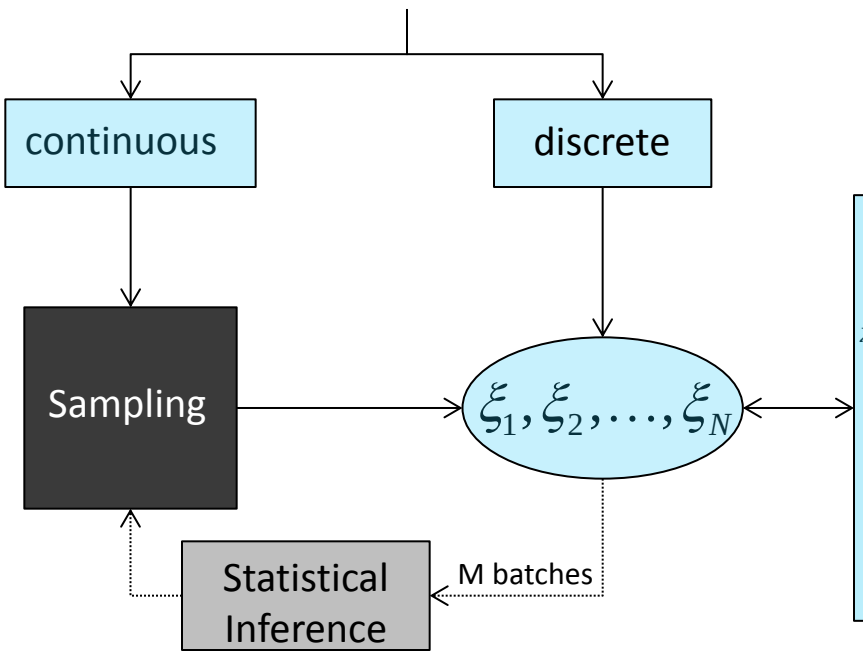
$x_0 \geq 0$

subj. to. $\mathrm{W}(\omega)x = b(\omega) - T(\omega)x_0$

$x \geq 0$

- $\xi(\omega) := (\mathrm{W}(\omega), \mathrm{T}(\omega), b(\omega), Q(\omega), c(\omega))$

$$f(x,\omega) = \frac{1}{2} x^T Q(\omega) x + x^T c(\omega)$$



Sample average approximation (SAA)

$$\begin{aligned} \underset{x_0, x_1, x_2, \ldots, x_N}{Min} \quad & f_0(x_0) + \frac{1}{N}\sum_{i=1}^{N} f_i(x_i) \\ \text{subj. to.} \quad & A_0 x_0 = b_0 \\ & T_k x_0 + W_k x_k = b_k, \\ & x_0 \geq 0, \quad x_k \geq 0, \quad k = 1, \ldots, \mathrm{N}. \end{aligned}$$

Large-scale (dual) block-angular LPs

Extensive form

$$
\begin{array}{lllllllllllll}
\min & c_0^T x_0 & + & c_1^T x_1 & + & c_2^T x_2 & + & \dots & + & c_N^T x_N & & \\
\text{s.t.} & Ax_0 & & & & & & & & & = & b_0, \\
& T_1 x_0 & + & W_1 x_1 & & & & & & & = & b_1, \\
& T_2 x_0 & & & + & W_2 x_2 & & & & & = & b_2, \\
& \vdots & & & & & & \ddots & & & & \vdots \\
& T_N x_0 & & & & & & & + & W_N x_N & = & b_N, \\
& x_0 \geq 0, & & x_1 \geq 0, & & x_2 \geq 0, & & \dots, & & x_N \geq 0. & &
\end{array}
$$

- In terminology of stochastic LPs:
  - First-stage variables (decision now): x_0
  - Second-stage variables (recourse decision): $x_1, \dots, x_N$
  - Each diagonal block is a realization of a random variable (scenario)

Computational challenges and difficulties

- May require many scenarios (100s, 1,000s, 10,000s …) to accurately model uncertainty

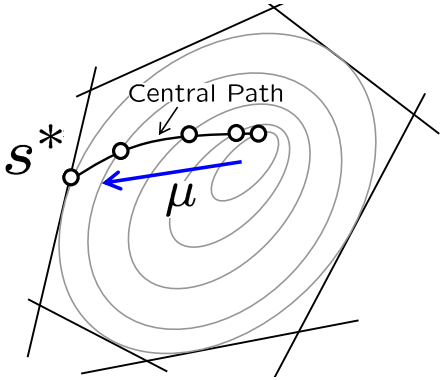
- "Large" scenarios (W_i up to 100,000 x 100,000)
- "Large" 1st stage (1,000s, 10,000s of variables)

- Easy to build a practical instance that requires 100+ GB of RAM to solve
  - ➔ Requires distributed memory

- Faster than real-time solution needed in many applications

PIPS solvers suite for stochastic optimization

- **PIPS-IPM** - interior-point method (IPM) - Mehrotra's predictor-corrector
  - Polynomial iteration complexity: $O(\sqrt{n}L)$ (worst-case theoretical bound)
  - Perform better in practice (infeasible primal-dual path-following)
  - *Specialized* parallel linear algebra



- **PIPS-NLP** – filter line-search IPM
  - Reuses the linear algebra of PIPS-IPM

- **PIPS-S** - revised simplex
  - *Specialized* parallel linear algebra

PIPS suite for structured optimization

Stochastic, Robust, Network

$$\min \sum_{\omega \in \Omega} f_\omega(x, x_\omega)$$
$$\text{s.t. } g_\omega(x, x_\omega) \geq 0, \omega \in \Omega$$

$$\begin{bmatrix} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_\Omega & & & & K_\Omega \end{bmatrix} \qquad K_\omega = \begin{bmatrix} H_\omega & J_\omega^T \\ J_\omega & \end{bmatrix}$$

Stochastic+Network

$$\begin{bmatrix} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_\Omega & & & & K_\Omega \end{bmatrix} \qquad K_\omega = \begin{bmatrix} \Phi_0 & C_1^\top & \cdots & C_n^\top \\ C_1 & \Phi_1 & & \\ \vdots & & \ddots & \\ C_n & & & \Phi_n \end{bmatrix} \qquad \Phi_\omega = \begin{bmatrix} H_\omega & J_\omega^T \\ J_\omega & \end{bmatrix}$$

Optimal Control PDAE/DAE

$$\min f(u, x)$$
$$\text{s.t. } g(u, x) \geq 0$$
$$J_x \text{ square \& invertible}$$

$$\begin{bmatrix} H_{xx} & H_{xu} & J_x^T \\ H_{ux} & H_{uu} & J_u^T \\ J_x & J_u & \end{bmatrix} \longrightarrow \left[\begin{array}{cc|c} J_x & & J_u \\ H_{xx} & J_x^T & H_{xu} \\ \hline H_{ux} & J_u^T & H_{uu} \end{array}\right]$$

Stochastic+Network+PDAE

$$\begin{bmatrix} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_\Omega & & & & K_\Omega \end{bmatrix} \qquad K_\omega = \begin{bmatrix} \Phi_0 & C_1^\top & \cdots & C_n^\top \\ C_1 & \Phi_1 & & \\ \vdots & & \ddots & \\ C_n & & & \Phi_n \end{bmatrix} \qquad \Phi_\omega = \left[\begin{array}{cc|c} J_x & & J_u \\ H_{xx} & J_x^T & H_{xu} \\ \hline H_{ux} & J_u^T & H_{uu} \end{array}\right]$$

Linear algebra of primal-dual interior-point methods

Convex quadratic problem

$$\text{Min } \frac{1}{2}x^T Q x + c^T x$$

$$\text{subj. to. } Ax = b$$

$$x \geq 0$$

IPM Linear Systems

$$\begin{bmatrix} Q+\Lambda & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = rhs$$

Multi-stage SP

Two-stage SP

nested arrow-shaped linear system (modulo a permutation)

$$\begin{bmatrix}
Q_1 & W_1^T & & & & & & 0 & 0 \\
W_1 & 0 & & & & & & T_1 & 0 \\
 & & Q_2 & B_2^T & & & & 0 & 0 \\
 & & W_2 & 0 & & & & T_2 & 0 \\
 & & & & \ddots & & & \vdots & \vdots \\
 & & & & & Q_S & W_S^T & 0 & 0 \\
 & & & & & W_S & 0 & T_S & 0 \\
0 & T_1^T & 0 & T_2^T & \dots & 0 & T_S^T & Q_0 & A_0^T \\
0 & 0 & 0 & 0 & \dots & 0 & 0 & A_0 & 0
\end{bmatrix}$$

Special Structure of KKT System (Arrow-shaped)

$$\begin{bmatrix} K_1 & & & B_1 \\ & \ddots & & \vdots \\ & & K_N & B_N \\ B_1^T & \dots & B_N^T & K_0 \end{bmatrix} \begin{bmatrix} \Delta z_1 \\ \vdots \\ \Delta z_N \\ \Delta z_0 \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ r_N \\ r_0 \end{bmatrix}$$

where,

$$K_i := \begin{bmatrix} \bar{Q}_i & W_i^T \\ W_i & 0 \end{bmatrix}, \qquad K_0 := \begin{bmatrix} \bar{Q} & A^T \\ A & 0 \end{bmatrix},$$

$$B_i := \begin{bmatrix} 0 & 0 \\ T_i & 0 \end{bmatrix}, \qquad i = 1, 2, \dots, N.$$

Block Elimination

$$\begin{bmatrix} K_1 & & & B_1 \\ & \ddots & & \vdots \\ & & K_N & B_N \\ B_1^T & \dots & B_N^T & K_0 \end{bmatrix} \begin{bmatrix} \Delta z_1 \\ \vdots \\ \Delta z_N \\ \Delta z_0 \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ r_N \\ r_0 \end{bmatrix}$$

Multiply row *i* by and sum all the rows to obtain

The matrix $C := K_0 - \sum_{i=1}^{N} B_i^T K_i^{-1} B_i$ is the Schur-complement of the diagonal $K_1, \dots, K_N$ block.

Solution Procedure for KKT System – a compact view

1. Calculate $B_i^T K_i^{-1} B_i,\ i = 1, \ldots, N$ ("Compute S.C.")
2. Form $C := K_0 - \sum_{i=1}^{N} B_i^T K_i^{-1} B_i$ ("Form S.C.")
3. Factorize $C = L_0 D_0 L_0^T$ ("Factor S.C.")
4. Solve $\Delta z_0 = C^{-1}(r_0 - \sum_{i=1}^{N} B_i^T K_i^{-1} r_i)$
5. Solve $\Delta z_i = K_i^{-1}(B_i \Delta z_0 - r_i),\ i = 1, \ldots, N$

Implementation - implicit (block) factorization

- Uses the arrow shape of H

$$\begin{bmatrix} K_1 & & & & B_1^T \\ & K_2 & & & B_2^T \\ & & \ddots & & \vdots \\ & & & K_N & B_N^T \\ B_1 & B_2 & \dots & B_N & K_0 \end{bmatrix} = \begin{bmatrix} L_1 & & & & \\ & L_2 & & & \\ & & \ddots & & \\ & & & L_N & \\ L_{10} & L_{20} & \dots & L_{N0} & L_c \end{bmatrix} \begin{bmatrix} D_1 & & & & \\ & D_2 & & & \\ & & \ddots & & \\ & & & D_N & \\ & & & & D_c \end{bmatrix} \begin{bmatrix} L_1^T & & & & L_{10}^T \\ & L^T{}_2 & & & L_{20}^T \\ & & \ddots & & \vdots \\ & & & L_N{}^T & L_{N0}^T \\ & & & & L^T{}_c \end{bmatrix}$$

1.Implicit factorization

$$L_i D_i L_i^T = K_i,$$

$$L_{i0} = B_i L_i^{-T} D_i^{-1}, i = 1, \dots N,$$

$$C = K_0 - \sum_{i=1}^{S} B_i K_i^{-1} B_i^T,$$

$$L_c D_c L_c^T = C.$$

2. Solving Kz=r

2.1. Back substitution

$$w_i = L_i^{-1} r_i, i = 1, \dots N,$$

$$w_0 = L_c^{-1} \left(r_0 - \sum_{i=1}^{N} L_{i0} w_i \right)$$

2.2. Diagonal Solve

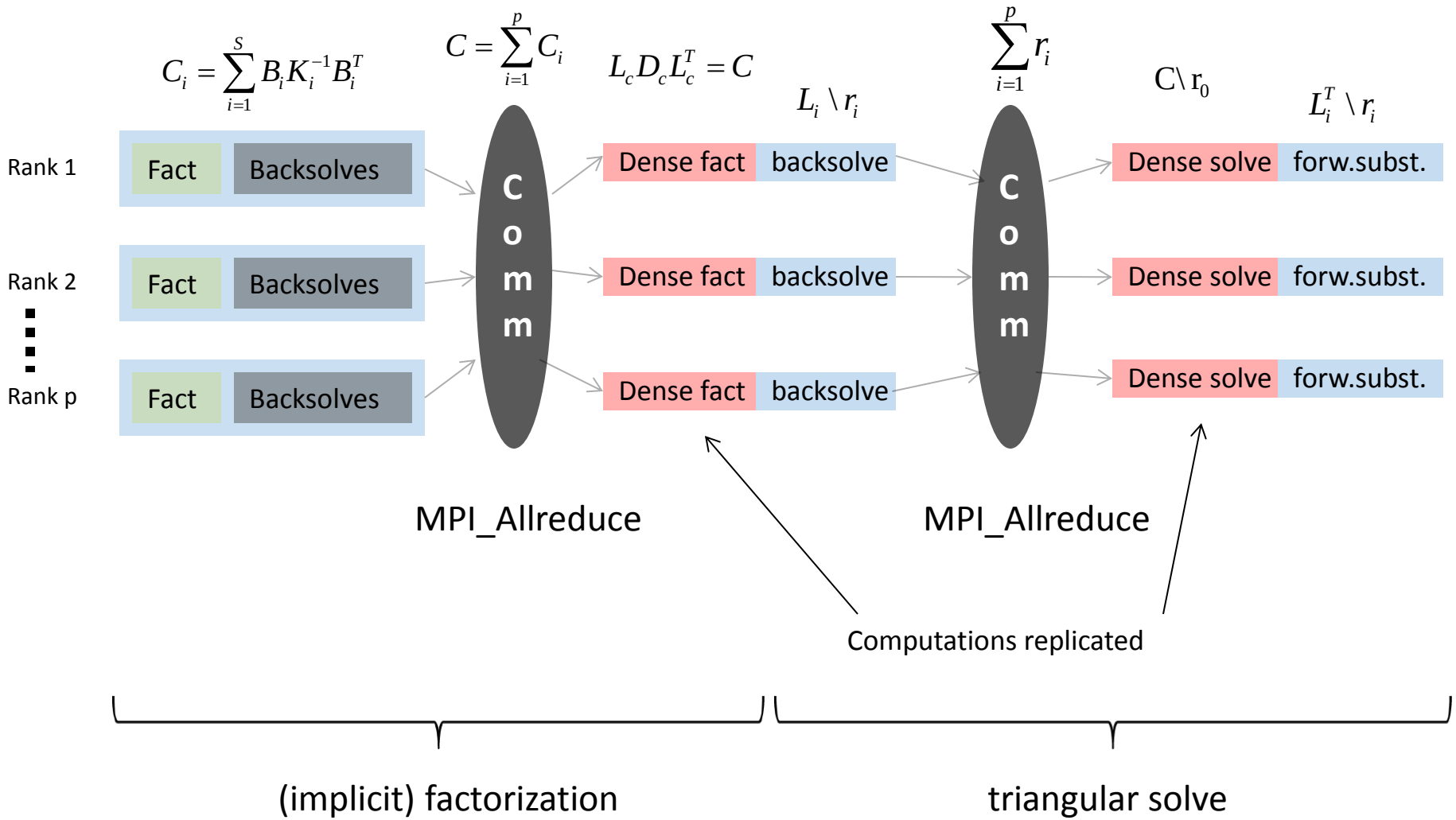
$$v_i = D_i^{-1} w_i, \ i = 0, \dots, \mathrm{N}$$

2.3. Forward substitution

$$z_0 = L_c^{-1} v_0$$

$$z_i = L_i^{-T} \left(v_i - L_{i0}^T z_0 \right), i = 1, \dots N.$$

Parallel computational pattern



High-performance computing

- Hybrid parallelism: MPI+OpenMP or MPI+GPU or MPI+MPI

- Supercomputers: IBM BG/P and BG/Q (Argonne), Cray XE6, XC30 (Swiss National Computer Center), and XK7 (Oak Ridge),

- Many incremental *algorithmic* and *implementation* developments over the years

Petra et al., "*A preconditioning technique for Schur complement systems arising in stochastic optimization,*" Journal of Computational Optimization and Applications, 2012.

Petra et al., "*Scalable Stochastic Optimization of Complex Energy Systems,*" in SC'11.

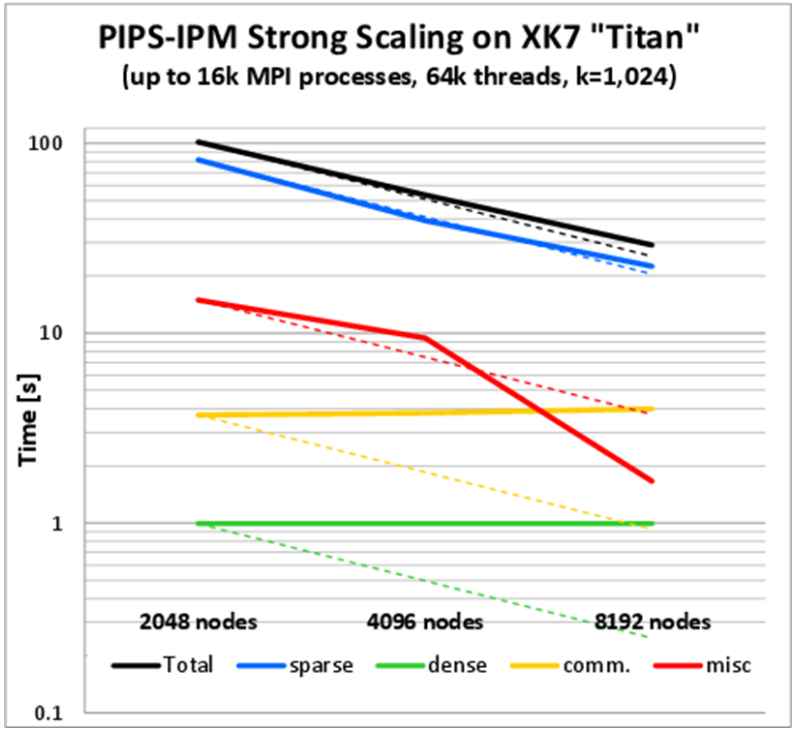
Petra et al., "*Real-time Stochastic Optimization of Complex Energy Systems on High Performance Computers,*" IEEE Computing in Science & Engineering (CiSE), 2014

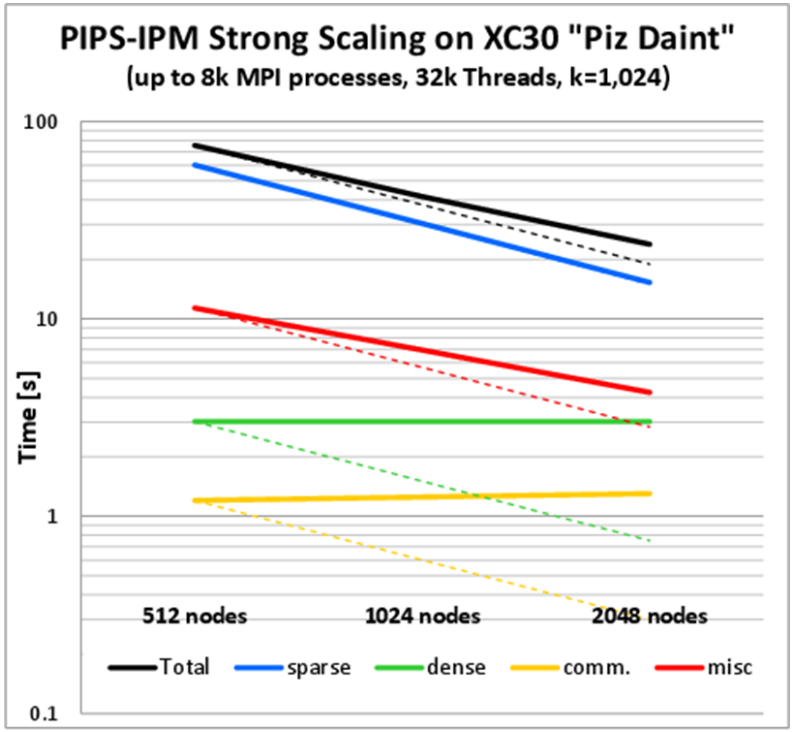
Lubin et al., "*On the parallel solution of dense saddle-point linear systems arising in stochastic programming,*" Journal of Optimization Methods and Software, 2012.

Petra et al., "*An augmented incomplete factorization approach for computing the Schur complement in stochastic optimization,*" SIAM Journal on Scientific Computing, 2014

- Economic dispatch for the State of Illinois power grid: 2522 lines, 1908 buses, 870 demand buses, 225 generators. Wind capacity is 17%.

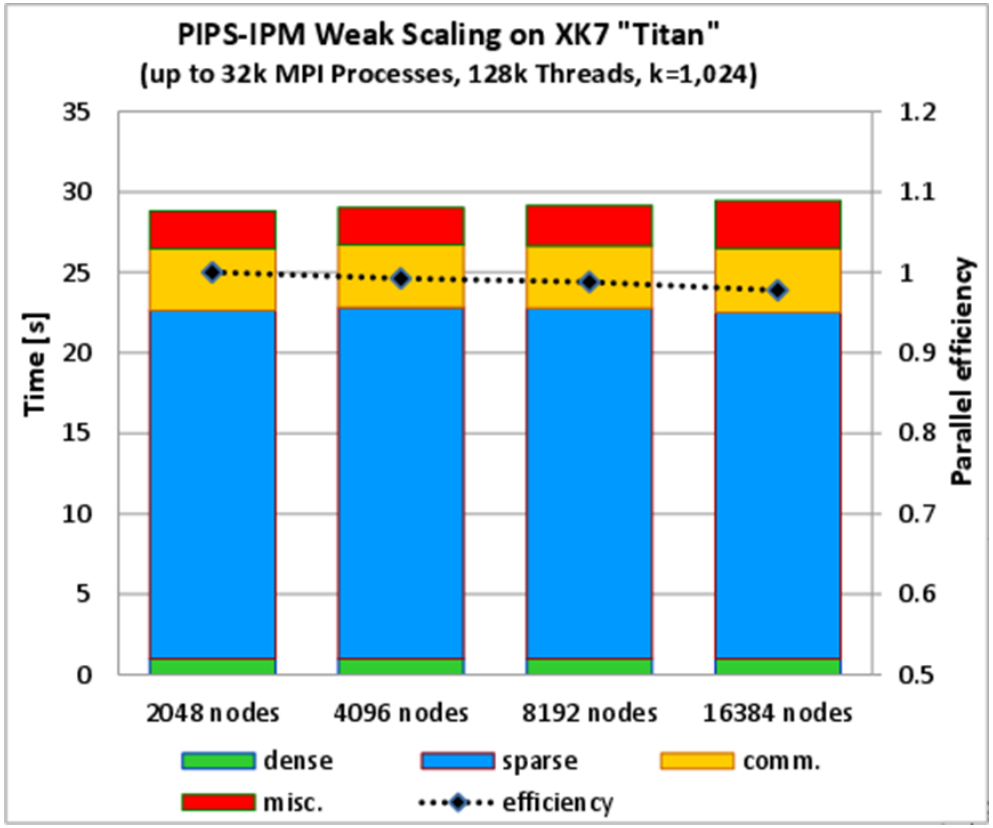
Strong scaling efficiency





The instance used in the XK7 runs has 4.08 billion decision variables and 4.12 billion constraints.

Weak scaling efficiency



Largest instance has 4.08 billion decision variables and 4.12 billion constraints.

Solve to completion – UC12 on BG/P

Nodes/scens	Wall time (sec)	IPM Iterations	Time per IPM iteration (sec)
4096	3548.5	103	33.57
8192	3883.7	112	34.67
16384	4208.8	123	34.80
32768	4781.7	133	35.95

In addition to implementation, **the algorithm is also scalable**

Time-to-solution multicore speed-up

Compute SC Times, UC12 – BG/P

Test Problem	Solver	Threads/MPI	$B_i^T K_i^{-1} B$ Time (Sec.)
UC4	MA57	1	29.65
	PARDISO-SC	1	2.45
	PARDISO-SC	2	1.32
	PARDISO-SC	4	0.79
UC12	MA57	1	292.02
	PARDISO-SC	1	50.39
	PARDISO-SC	2	26.09
	PARDISO-SC	4	13.81
UC24	MA57	1	>3600
	PARDISO-SC	1	308.193
	PARDISO-SC	2	157.63
	PARDISO-SC	4	81.11

Compute SC time, UC24 – XC30

# of MPI ranks	(b) Number of threads per process on XC30					
	1	1	2	4	8	16
	PARDISO	PARDISO-SC				
1	168.3	19.0	12.9	8.4	6.2	5.9
2	175.3	18.9	12.4	8.5	7.4	
4	194.6	19.5	13.8	12.1		
8	284.2	22.1	21.8			
16	281.6	38.5				

TSOPF – only incomplete Hessian is available

$$
\begin{aligned}
\min \quad & C(p) \\
\text{s.t.} \quad & g_s(p) = 0 \\
& h_s(p) \leq 0 \\
& p^- \leq p \leq p^+ \\
& H_c(p) = 0, \quad c \in \mathcal{C}
\end{aligned}
$$

OPF: *algebraically (closed-form) described*
First and second order derivatives readily available (AutoDiff or hand coding)

TS: *simulation-based with adjoint sensitivities*

$$H(p) = \int_{t_0}^{t_f} h(x(t,p), y(t,p))dt$$

(one DAE solve)

$$
\begin{aligned}
M\dot{x} &= f(t,x,y,p), \\
0 &= g(t,x,y,p), \\
x(t_0) &= I_{x_0}(p), \\
y(t_0) &= I_{y_0}(p)
\end{aligned}
$$

$$\nabla H(p) = \int_{t_0}^{t_f} f_p^T \lambda dt - ((Mx_p)^T \lambda)_{t=0}$$

(one adjoint solve)

$$
\begin{aligned}
M^T\dot{\lambda} &= -f_x^T\lambda + g_x^T\mu - h_x, \\
0 &= -f_y^T\lambda + g_y\mu - h_y, \\
M^T\lambda(t_f) &= 0, \\
\mu(t_f) &= 0
\end{aligned}
$$

Structured secant updates for incomplete Hessian

$$\min_{x \in \mathbb{R}^n} \quad f(x) := k(x) + u(x)$$

Available problem information

$$k(x), \quad \nabla k(x), \quad \nabla^2 k(x)$$
$$u(x), \quad \nabla u(x)$$

- Want an approximation of the Hessian that uses the available Hessian, namely

$$B_{i+1} = \nabla^2 k(x_{i+1}) + A_{i+1}, \text{ where } A_{i+1} \approx \nabla^2 u(x_{i+1})$$

- Secant equation – match only missing curvature

$$A_{i+1} s_i = \bar{y}_i := \nabla u(x_{i+1}) - \nabla u(x_i)$$

- Proximity criteria – the tricky part
  - Enforced for A_{i+1} or for B_{i+1}?

- Insight from the trace-determinant variational characterization: a minimal deterioration of the eigenspectrum of the Hessian (DFP) or inverse Hessian approximation (BFGS) occurs if proximity is enforced on B_{i+1}.

Deriving the structured BFGS update

$$\min_{A} \quad \|(K(x_{i+1})+A)^{-1}-(K(x_{i+1})+A_i)^{-1}\|_{F,W}$$
$$\text{s.t.} \qquad A=A^T,\ As_i=\bar{y}_i$$

Notation: $K(x)=\nabla^2 k(x)$

To derive an analytical solution take

$$H:=(K(x_{i+1})+A)^{-1},\ H_i:=(K(x_{i+1})+A_i)^{-1}.$$

Then write

$$\min_{H} \qquad \|H-H_i\|_{F,W}$$
$$\text{s.t.} \quad H=H^T,\ H(K(x_{i+1})s_i+\bar{y}_i)=s_i$$

H is a projection on a linear subspace (Dennis et. al. (1979), Guler et. al. (2009)) given by

$$H=(I-\bar{\gamma}_i s_i(K(x^{i+1})s_i+\bar{y}_i)^T)H_i(I-\bar{\gamma}_i(K(x^{i+1})s_i+\bar{y}_i)s_i^T)+\bar{\gamma}_i s_i s_i^T$$

Using Sherman-Morison-Woodbury to derive a formula for the inverse of H leads to

$$A_{i+1}^{BFGS}=A_i-\frac{(A_i+K(x_{i+1}))s_i s_i^T(A_i+K(x_{i+1}))}{s_i^T(A_i+K(x_{i+1}))s_i}+\bar{\gamma}_i\bar{y}_i\bar{y}_i^T,\quad \bar{\gamma}_i=\frac{1}{(K(x_{i+1})s_i+\bar{y})^T s_i}$$

A structured quasi-Newton method

- We use the framework of line-search methods
- Hereditary positive definiteness via a modified version of Wolfe conditions
  - algorithm still globally convergent
- Superlinear convergence proved under the assumption of Lipschitz continuous Hessian
- These are standard convergence results for quasi-Newton methods with secant updates
- Provably better than general quasi-Newton when the unknown part of Hessian is low-rank

C.G. Petra, N. Chiang, *Structured secant updates for unconstrained optimization with incomplete Hessian information*. In preparation.

Numerical results on CUTEst test problems

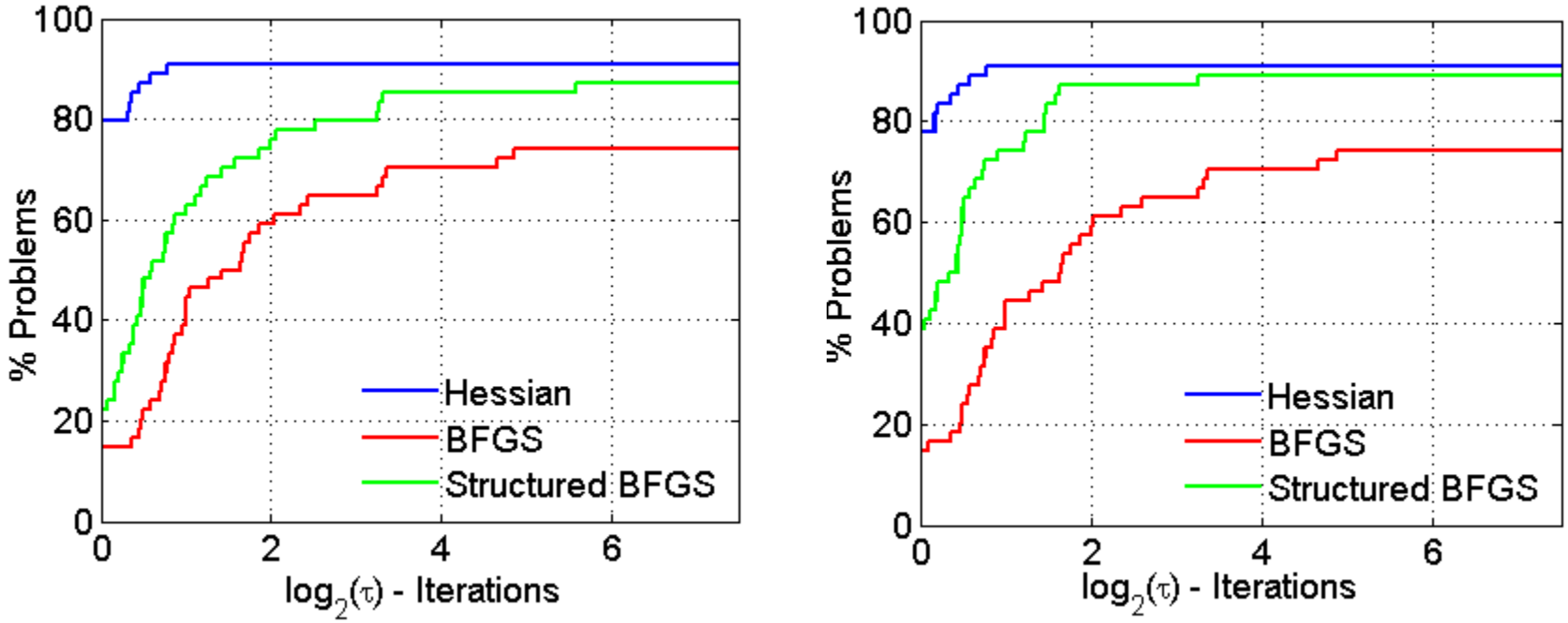


Figure 1: Performance profiles for nonlinear unconstrained CUTEst problems: left plot for problems with full-rank missing Hessian terms, right plot for low-rank missing Hessian terms.

A HPC study of the economic impacts of wind correlation on power dispatch operations

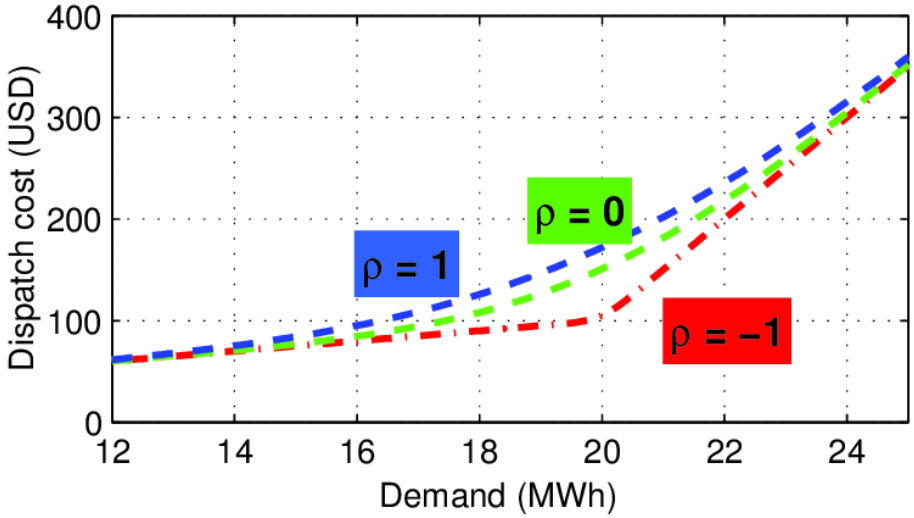
How should ISO integrate wind scenarios?

- **Approach 1**: Receive power bids from wind farms that are based on independent forecasting, therefore, wind correlation is lost (**indep.**)
- **Approach 2**: Centralized wind forecast at the ISO level (**corr.**)

- A simplistic model: 3 generators (2 wind farms and 1 thermal
- Power outputs of the wind farms are $W_1 \sim \mathcal{N}(w_1, \sigma_1)$ and $W_2 \sim \mathcal{N}(w_2, \sigma_2)$, and the correlation is ρ

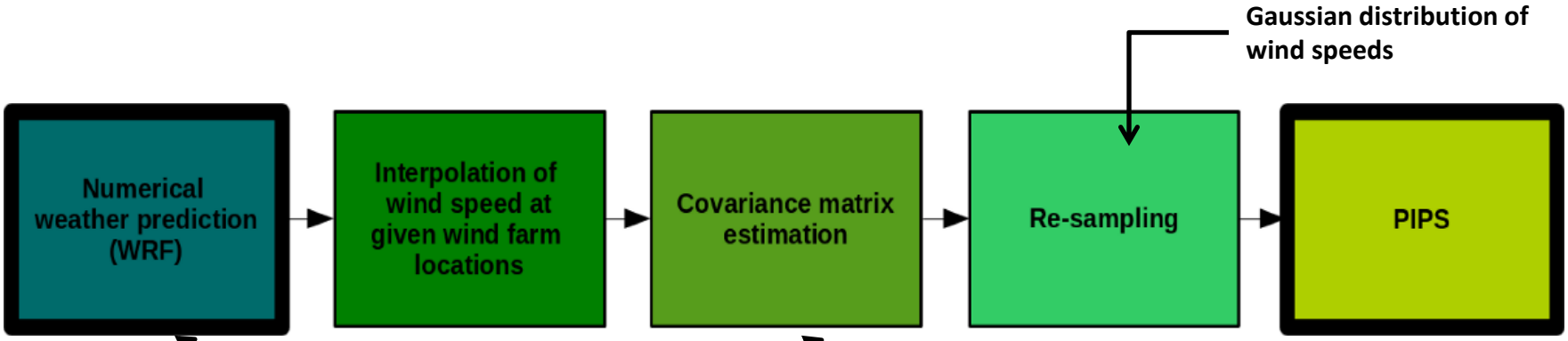
$$c_d(\rho) = c_w d + (c_{th} - c_w)((d - w_1 - w_2)\Phi(d, \sigma_1^2 + 2\rho\sigma_1\sigma_2 + \sigma_2^2) + \sigma^2\phi(d, \sigma_1^2 + 2\rho\sigma_1\sigma_2 + \sigma_2^2))$$

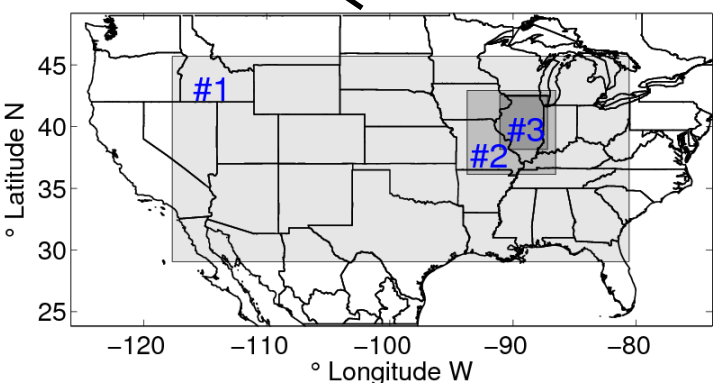
- The optimal dispatch cost is an **increasing function** of the correlation ρ!



What about real-world large-scale power grid systems?

- Simulation framework based on PIPS-IPM
- **Illinois power grid**: 2522 lines, 1908 buses, 870 demand buses, 225 generators, of which 32 are wind farms. Wind "installed" capacity is 17%. Adoption in around 15%.





Weather forecasting @ Argonne HPC simulation – 30 samples in RT (E. Constantinescu)

$$S_\Theta = \rho_1 \cdot I + \rho_2 \cdot S$$

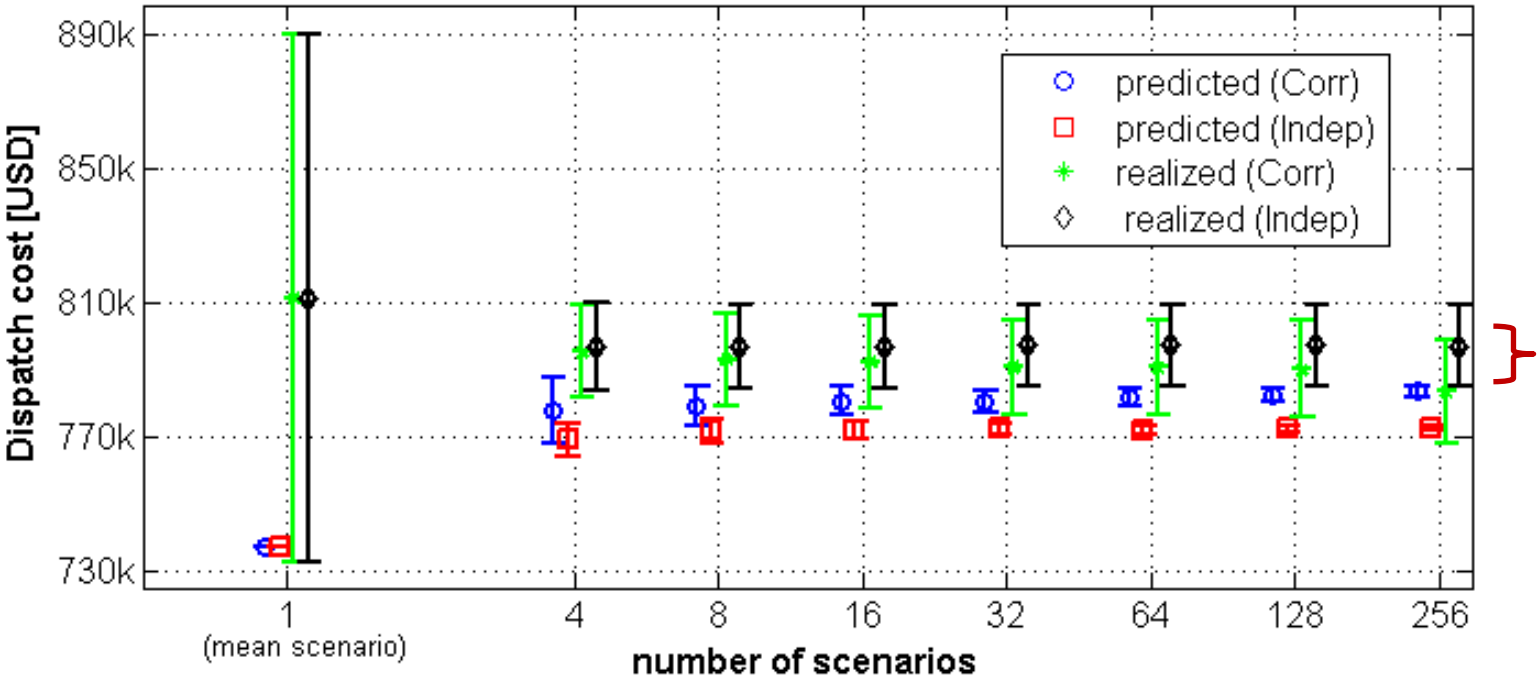
$$\min_{\rho_1, \rho_2} \mathbb{E}\left[\|Q - S_\Theta\|\right]$$

$$\widehat{\Sigma}_{RBLW} = \rho_{RBLW} \cdot I_{p\times p} + (1 - \rho_{RBLW}) \cdot S, \text{ where}$$

$$\rho_{RBLW} = \min\left(\frac{\frac{n-2}{n}\cdot \mathrm{tr}\left(S^2\right) + \mathrm{tr}^2(S)}{(n+2)\cdot\left[\mathrm{tr}(S^2) - \frac{\mathrm{tr}^2(S)}{p}\right]}, 1\right)$$

Biases in dispatch cost – independent vs. centralized whether forecasting

95% confidence intervals for the dispatch cost for predicted and realized costs, each with (Corr) and without (Indep) correlation information

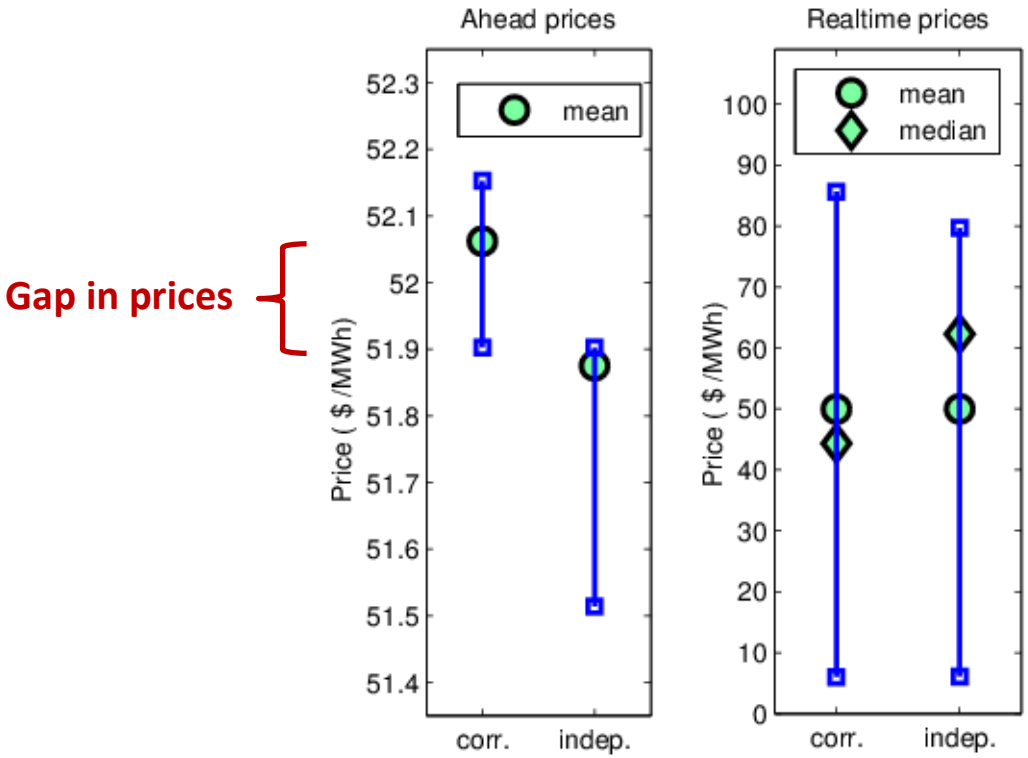


"Bias" in dispatch cost

- 1.42% bias/gap ($10,967 per trading period, potentially $100 million/year)

Electricity market implications

95% confidence intervals for prices at a typical bus



- "Bias" is also present in the ahead prices.
- Opportunities for market arbitrage for players with better covariance information.

Thank you for your attention!

Questions?