Optimization under Uncertainty – Scalable Stochastic Programming (Optimization of Complex Systems)

Mihai Anitescu

Mathematics and Computer Science Division
Argonne National Laboratory

With
C. Petra, Miles Lubin
V. Zavala, E. Constantinescu,

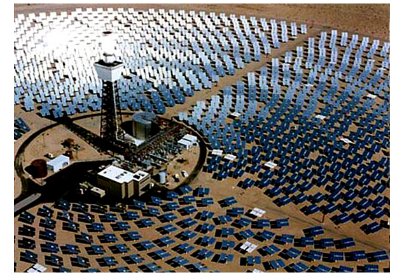
AMR PI Meeting Oct 2011

1. Motivation: Management of Energy Systems under Ambient Conditions Uncertainty

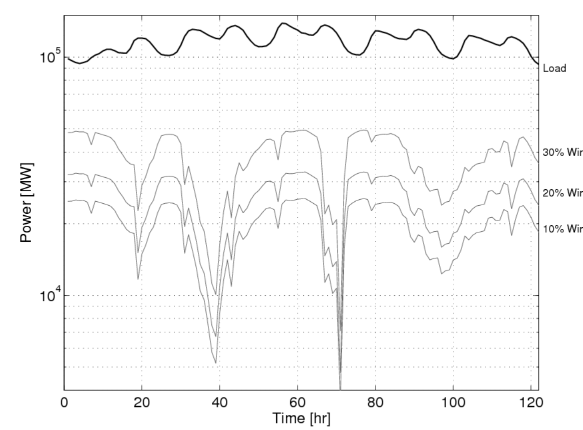
Ambient Condition Effects in Energy Systems

Operation of Energy Systems is Strongly Affected by Ambient Conditions

- **Power Grid Management:** Predict Spatio-Temporal Demands *(Douglas, et.al. 1999)*
- **Power Plants:** Generation levels affected by air humidity and temperature *(General Electric)*
- **Petrochemical:** Heating and Cooling Utilities *(ExxonMobil)*
- **Buildings:** Heating and Cooling Needs *(Braun, et.al. 2004)*
- (Focus) **Next Generation Energy Systems** assume a major renewable energy penetration: Wind + Solar + Fossil *(Beyer, et.al. 1999)*



- Increased reliance on renewables must account for variability of ambient conditions, which cannot be done deterministically …
- We must optimize operational and planning decisions accounting for the uncertainty in ambient conditions (and others, e.g. demand)
- **Optimization Under Uncertainty.**



Wind Power Profiles

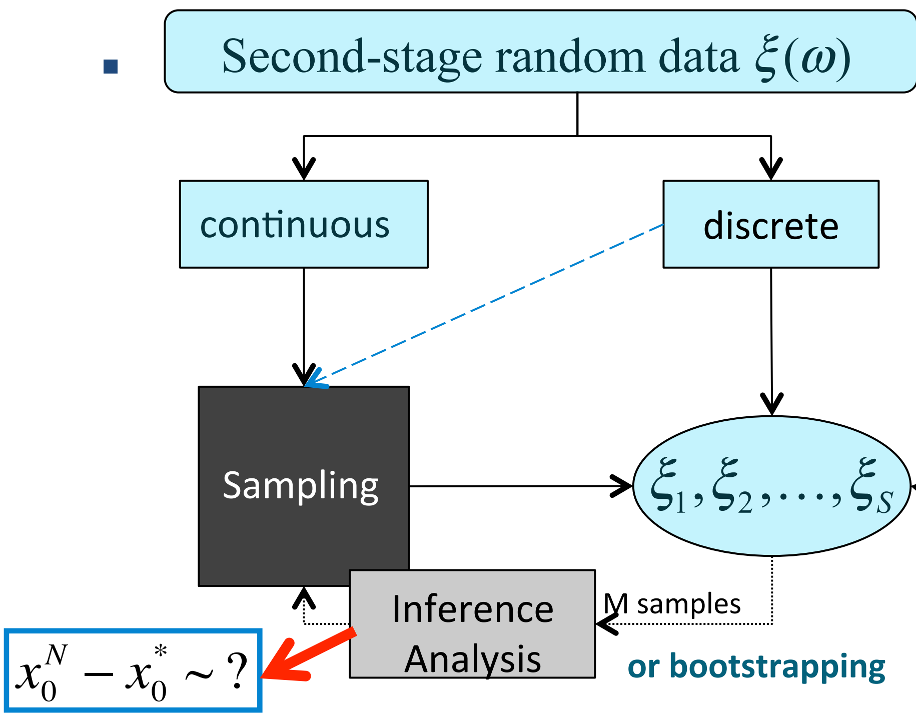
Other Optimization under Uncertainty Apps

- Increased Requirement for Uncertainty Quantification is Likely to Require Increased Emphasis on Optimization under Uncertainty.
- Possible Examples:
  - Optimal Nuclear Core Reloading (since fuel is reused in nuclear reactors).
  - Materials-by-Design accounting for model uncertainty.
  - Environmental Remediation – Nuclear Legacy Site Cleanup.
  - Whole Device Modeling for Fusion Applications
  - Infrastructure planning (electricity, gas, water, communication, transportation)
  - ……

A leading paradigm for optimization under uncertainty paradigm: stochastic programming.

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

$$\underset{x_0}{Min}\left\{f_0(x_0)+\mathbb{E}\left[\ \underset{x}{Min}\, f(x,\omega)\ \right]\right\} \longrightarrow \boxed{x_0^*}$$
$$\text{s.t.}\quad g_0(x_0)=b_0 \qquad \text{s.t.}\quad h(x,\omega)=b(\omega)-g(x_0,\omega)$$
$$x_0\geq 0 \qquad\qquad x(\omega)\geq 0$$

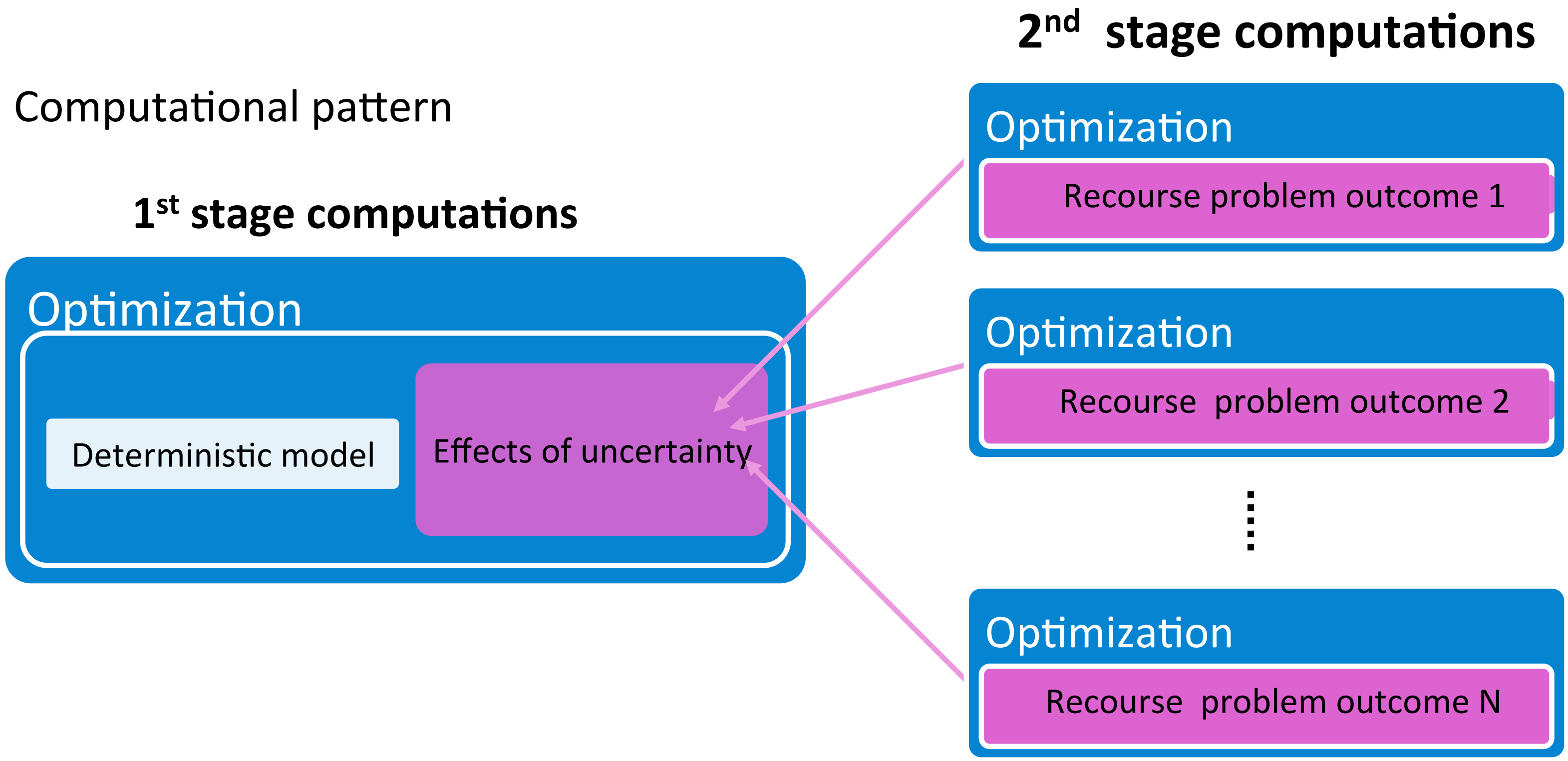
- 

Sample average approximation (SAA) $\longrightarrow \boxed{x_0^N}$

$$\underset{x_0,x_1,x_2,\ldots,x_S}{Min}\quad f_0(x_0)+\frac{1}{S}\sum_{k=1}^{S} f_k(x_k)$$
$$\text{subj. to. } g_0(x_0) \qquad\qquad = \quad b_0$$
$$g_k(x_0)+\quad h_k(x_k) \quad = \quad b_k,$$
$$x_0\geq 0,\quad x_k\geq 0,\qquad k=1,\ldots,S.$$

Stochastic programming – a non-trivial parallel paradigm suitable for next-generation supercomputers

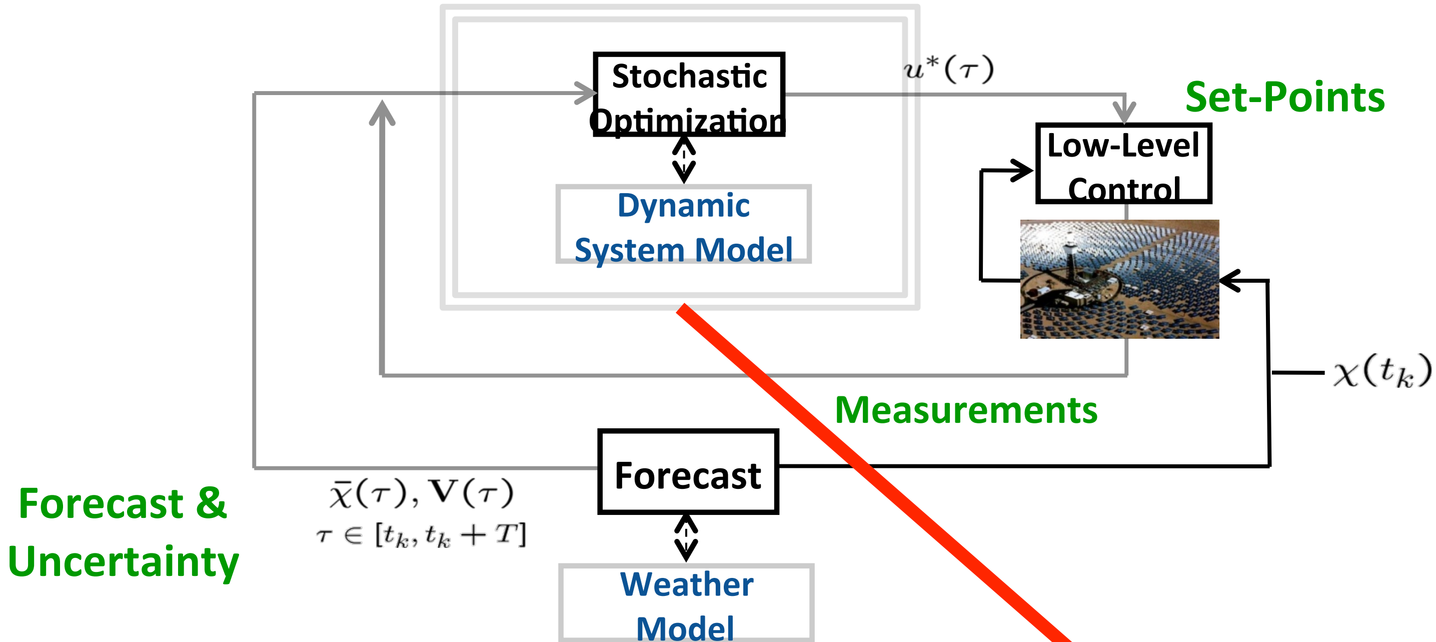
- Computational pattern



- Extra, in-node parallelization can be obtained for both 1st and 2nd stage.
- Algorithmic developments are needed to ensure efficient communication, fault resilience and good load balancing.
- Same pattern for statistical model CALIBRATION.

2. Impact: Stochastic Unit Commitment – Management of Energy Systems

Stochastic Predictive Control



Two-stage Stoch Prog

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

subj. to.

$$\begin{aligned} g_0(x_0) &= b_0 & \\ g_i(x_0, x_i) &= b_i & i = 1,2\ldots S \\ x_0 \geq 0, &\quad x_i \geq 0 & \end{aligned}$$

Stochastic NLMPC

$$\min_{u(t)} \quad \mathbf{E}_{\chi(t)\in\Omega}\left[\int_{t_\ell}^{t_{\ell+N}} \varphi(z(t), y(t), u(t), \chi(t))\,dt \right]$$

$$\left.\begin{aligned} \frac{dz}{dt} &= \mathbf{f}(z(t), y(t), u(t), \chi(t)) \\ 0 &= \mathbf{g}(z(t), y(t), u(t), \chi(t)) \\ 0 &\geq \mathbf{h}(z(t), y(t), u(t), \chi(t)) \\ z(0) &= x_\ell \end{aligned}\right\}$$

Stochastic Unit Commitment with Wind Power (SAA)

$$\min \qquad \text{COST} = \frac{1}{N_s}\sum_{s\in\mathcal{S}}\left(\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{T}} c^p_{sjk} + c^u_{jk} + c^d_{jk}\right)$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}} p_{sjk} + \sum_{j\in\mathcal{N}_{wind}} p^{wind}_{sjk} = D_k, s\in\mathcal{S}, k\in\mathcal{T}$$

$$\sum_{j\in\mathcal{N}} \overline{p}_{sjk} + \sum_{j\in\mathcal{N}_{wind}} p^{wind}_{sjk} \geq D_k + R_k, s\in\mathcal{S}, k\in\mathcal{T}$$

ramping constr., min. up/down constr.

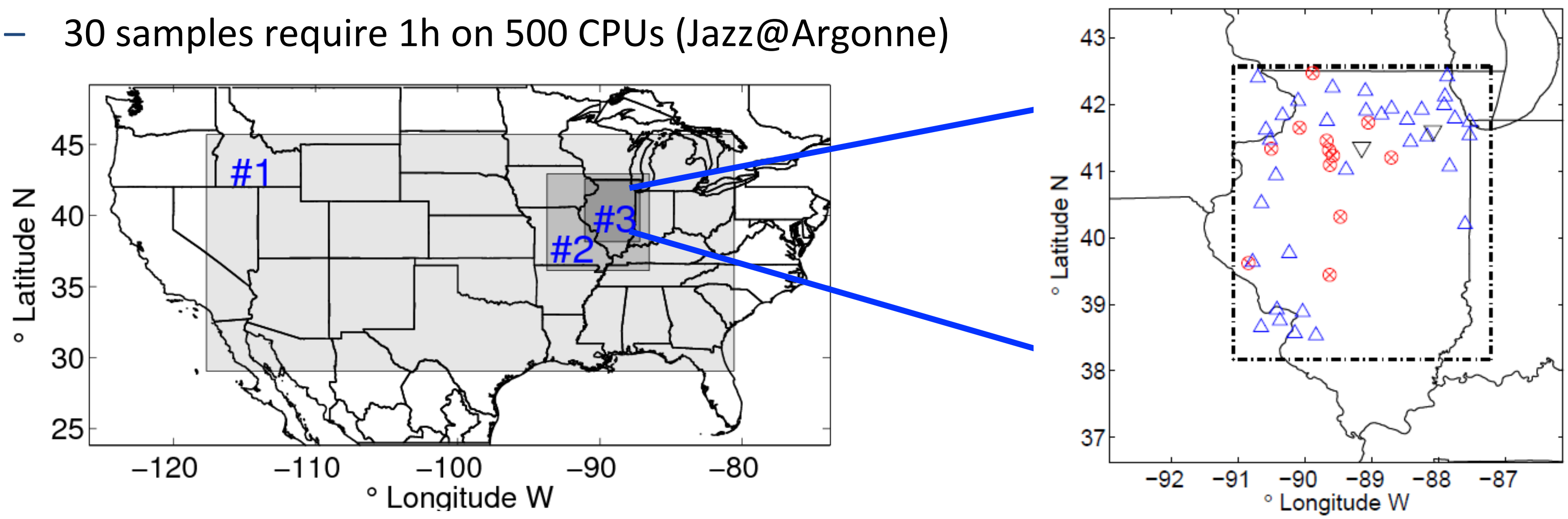
Thermal Units Schedule? Minimize Cost

Satisfy Demand

Have a Reserve

Dispatch through network

- Wind Forecast – WRF(Weather Research and Forecasting) Model
  - Real-time grid-nested 24h simulation
  - 30 samples require 1h on 500 CPUs (Jazz@Argonne)



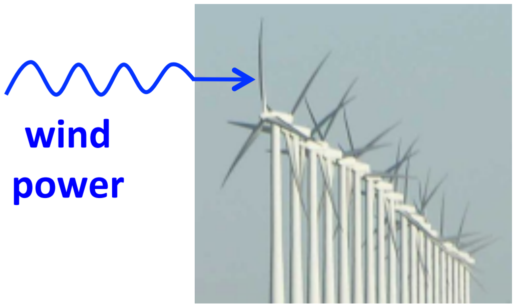
Zavala & al 2010.

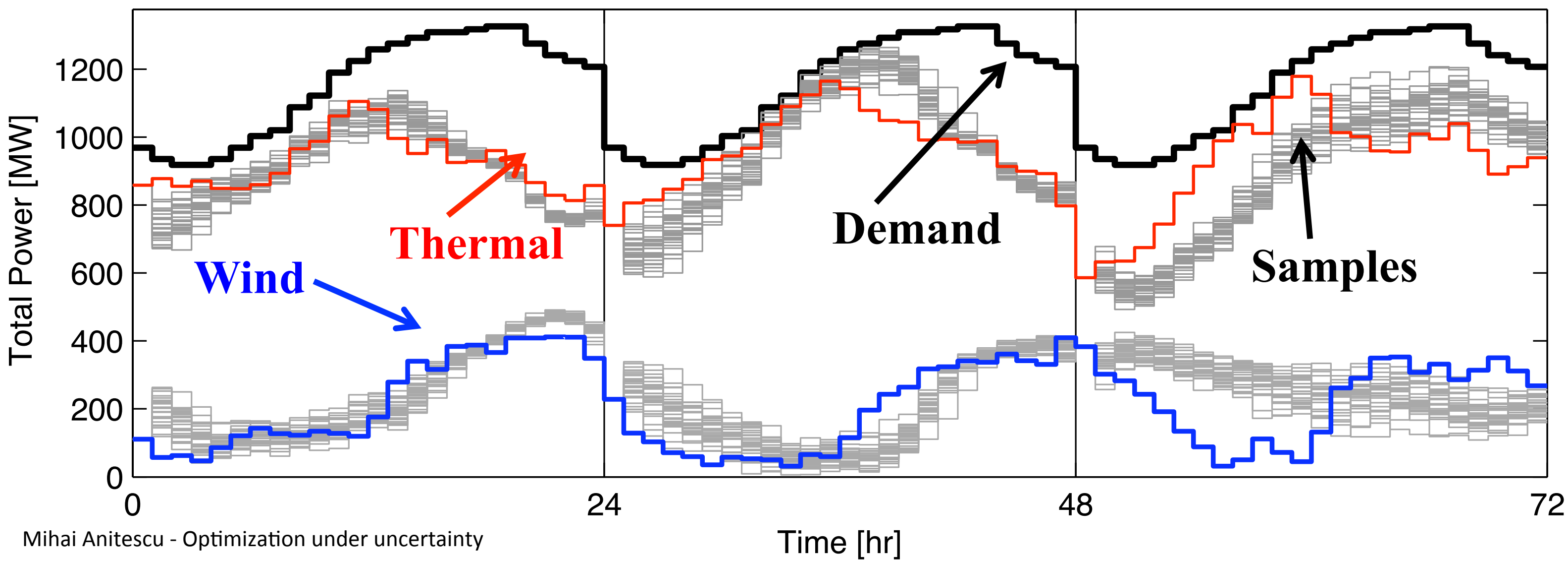
Wind power forecast and stochastic programming

- Unit commitment & energy dispatch with <u>uncertain wind power generation for the State of Illinois, assuming 20% wind power penetration, using the same windfarm sites as the one existing today.</u>

- Full integration with 10 thermal units to meet demands. Consider dynamics of start-up, shutdown, set-point changes

- The solution is only 1% more expensive then the one with exact information. Solution on average infeasible at 10%.





Some Considerations in Using Supercomputing for Power Grid

- Is it really worth using a supercomputer for this task? (We need the answer every 1hr with 24 hour time horizons.)
- Let's look at the most pressing item of Supercomputing usage: power.
  - BG/P (and exascale) needs <~ 20MW of power.
  - The Midwest US has 140GW of power installed, and the peak demands runs up to 110GW.
  - We will never reduce power consumption, but we will make it more reliable, less dependent on fossil, and cheaper by better managing the peak
- If we accept this will lead to 10% more renewable penetration (our SUC study), then this is worth on the order of 10-15GW, far above what BG/P costs in power consumption.
- In addition operational constraints makes supercomputing (if uncertainty needed to account for) necessary and not just useful or convenient.
- But, even if approximations will work, this tool will be helpful as the "gold standard" for validating other algorithms to be deployed on defined computational resources.

3. Low-Hanging Fruit Scalable Software: PIPS (Parallel Interior Point Stochastic Programming) – Petra, Lubin, Anitescu

PIPS – Our Scalable Stochastic Programming Solver Using Direct Schur Complement Method

- The arrow shape of H

$$
\begin{bmatrix} H_1 & & & & G_1^T \\ & H_2 & & & G_2^T \\ & & \ddots & & \vdots \\ & & & H_S & G_S^T \\ G_1 & G_2 & \dots & G_S & H_0 \end{bmatrix}
=
\begin{bmatrix} L_1 & & & & \\ & L_2 & & & \\ & & \ddots & & \\ & & & L_S & \\ L_{10} & L_{20} & \dots & L_{S0} & L_c \end{bmatrix}
\begin{bmatrix} D_1 & & & & \\ & D_2 & & & \\ & & \ddots & & \\ & & & D_N & \\ & & & & D_c \end{bmatrix}
\begin{bmatrix} L_1^T & & & & L_{10}^T \\ & L_2^T & & & L_{20}^T \\ & & \ddots & & \vdots \\ & & & L_S^T & L_{S0}^T \\ & & & & L_c^T \end{bmatrix}
$$

- Solving Hz=r

$$L_i D_i L_i^T = H_i,\ \ L_{i0} = G_i L_i^{-T} D_i^{-1},\quad i = 1,\dots S,$$

$$C = H_0 - \sum_{i=1}^{S} G_i H_i^{-1} G_i^T,\ \boxed{L_c D_c L_c^T = C.}$$

Implicit factorization, C is **dense**, H's are sparse.

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

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

Back substitution

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

Diagonal solve

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

$$z_i = L_i^{-T} \left(v_i - L_{i0}^T z_0 \right),$$

$$i = 1,\dots S.$$

Forward substitution

PIPS – Parallel solver for stochastic optimization

- Interior-point method implementation (Mehrotra's algorithm)
- Scenario-based decomposition of the linear algebra.
- PIPS reuses OOQP (Object-oriented quadratic programming solver) class hierarchy.
- New parallel linear algebra layer for block-angular IPM linear systems.
- Hybrid MPI+SMP parallelization

- First-stage Schur complement: **dense** linear algebra.
  - Distributed factorization and backsolves (by using Elemental) if needed.
  - Shared-memory parallelization(SMP) is obtained via Elemental.
  - Distributed assembling of the SC matrix is done by a streamlined **Reduced_scatter** that is also in-node SMP-accelerated.

- Second-stage linear systems are **sparse**.
  - Supports various sparse solvers: MA57 (HSL UK), WSMP(IBM).
  - SMP is obtained with WSMP

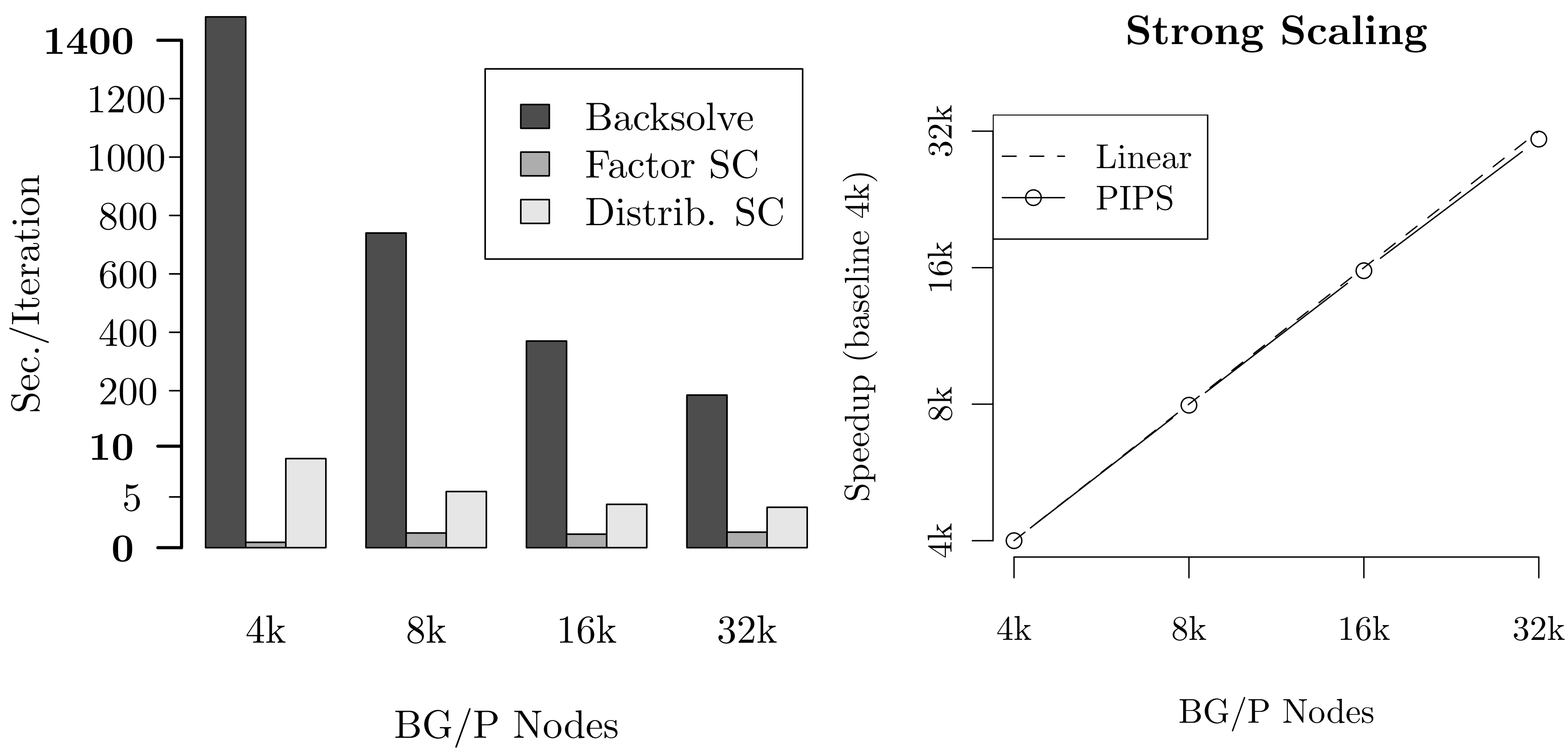
PIPS Solver Capabilities

- Hybrid MPI/SMP running on Blue Gene/P
  - Successfully (though incompletely due to allocation limit) run on up to **32,768** nodes (**96**% strong scaling) for Illinois problem with grid constraints. 3B variables, maybe largest ever solved?

- Handles up to 100,000 first-stage variables. Previous results dealt with O(20-50).

- Close to real-time solutions (24 hr horizon in 1 hr wallclock)
  - Further development needed, since users aim for
    - More uncertainty, more detail (x 10)
    - Faster Dynamics → Shorter Decision Window (x 10)
    - Longer Horizons (California == 72 hours) (x 3)

Components of Execution Time and Strong Scaling



- 32K nodes=130K cores (80% BG/P)
- "Backsolve" phase embarrassingly parallel, but not Schur Complement (SC)
- Communication for "Distrib. SC" not yet a bottleneck, but we will get there.

4. The harder problems need some mathematics

4.1 Q1: How do I deal with the impending first-stage bottleneck?

The Stochastic Preconditioner

- The limiting factor in the scalability of Schur method is the expensive solve with **dense** Schur complement matrix

$$C = \widetilde{Q}_0 + \frac{1}{N}\sum_{i=1}^{N}\left[A_i^T \left(B_i \widetilde{Q}_i^{-1} B_i^T\right)^{-1} A_i \right]$$

  - A computational bottleneck: workers sit idle waiting for the master to factorize C.

- Remedy – the preconditioned Schur complement (PSC)
  - 1. factorize incomplete matrix P in the same time C is computed.
  - 2. use the factorization of P to solve with C very fast.

- In linear algebra terms
  - P is a preconditioner for C. Our choice of P is

$$S_n = \widetilde{Q}_0 + \frac{1}{n}\sum_{i=1}^{n}\left[A_{k_i}^T \left(B_{k_i} \widetilde{Q}_{k_i}^{-1} B_{k_i}^T\right)^{-1} A_{k_i} \right],$$

  where $\mathcal{K} = \{k_1, k_2, \ldots, k_n\}$ is an IID subset of n scenarios.
  - Krylov iteratives solves (PCG or BiCGStab) replaces the direct solves

Preconditioned Schur Complement (PSC)

P is a C built from a subset of scenarios

$$L_i D_i L_i^T = H_i, \; L_{i0} = G_i L_i^{-T} D_i^{-1}, \quad i = 1, \dots N,$$
$$C = H_0 - \sum_{i=1}^{N} G_i H_i^{-1} G_i^T$$

$$L_M D_M L_M^T = P$$

(separate process)

$$w_i = L_i^{-1} r_i$$
$$\tilde{r}_0 = \left(r_0 - \sum_{i=1}^{N} L_{Ni} w_i \right)$$

$$v_i = D_i^{-1} w_i, \; i = 0, \dots, N$$
$$z_0 = \mathrm{Krylov}(C, P, \tilde{r}_0)$$

$$z_i = L_i^{-T} \left(v_i - L_{i0}^T z_0 \right)$$

Quality of the Stochastic Preconditioner

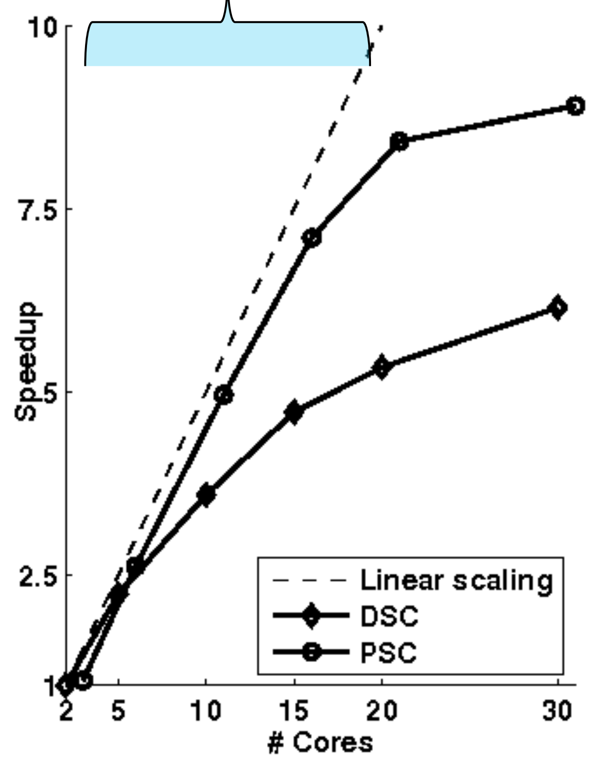
- **"Exponentially" better preconditioning** (Petra & Anitescu 2010)

$$\Pr(|\lambda(S_n^{-1}S_N)-1|\geq\varepsilon)\leq 2p^4\,\exp\left(-\frac{n\varepsilon^2}{2p^4L^2\|S_N\|_{max}^2}\right)$$

- A typical scaling behavior of our approach. Better scaling than the direct Schur complement method (DSC) is exhibited by PSC.

- DSC uses p processes, PSC uses p+1.

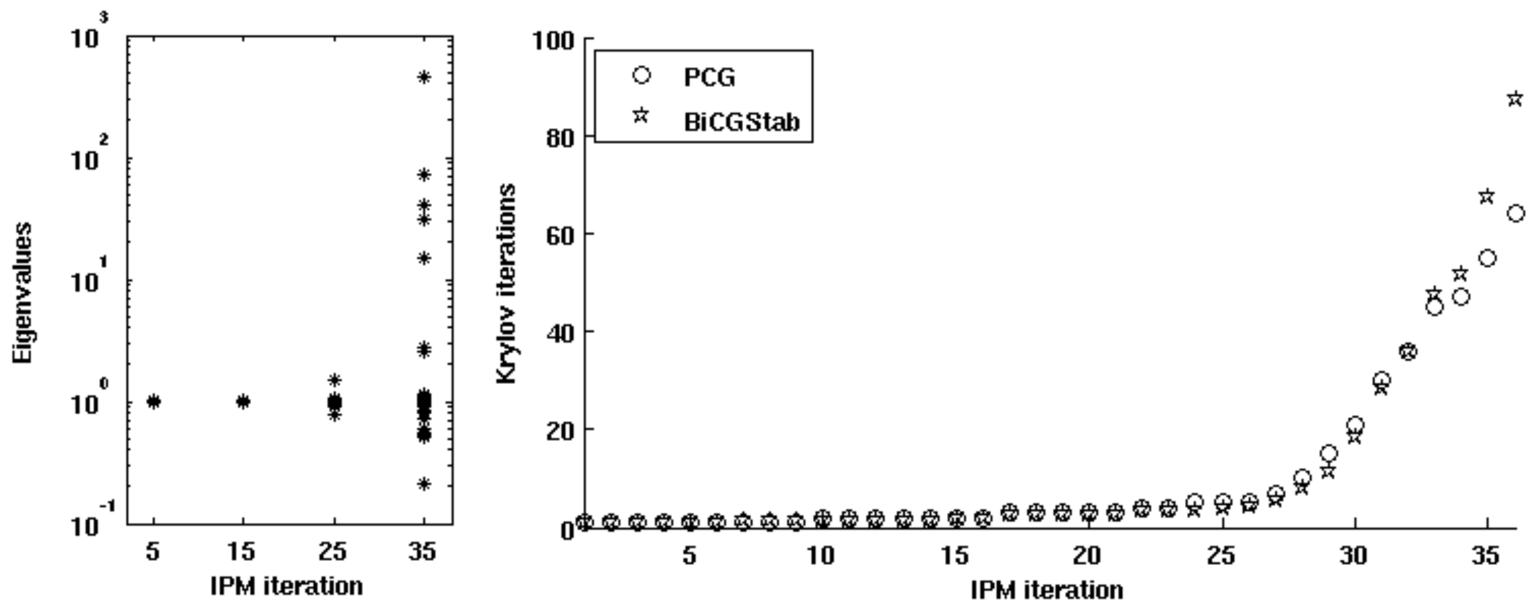
Optimal use of PSC – **linear scaling**



Factorization of the preconditioner can not be hidden anymore by the computation of C.

Performance of the preconditioner

- Eigenvalues clustering & Krylov iterations



- Affected by the well-known ill-conditioning of IPMs.

$$S_n \approx S_N \text{ and } \mathrm{S}_N \approx \mathsf{E}\left[S(\omega)\right], \text{ where}$$

$$S(\omega) = \left(Q_0 + D_0\right) + \left[A^T(\omega)\left(B(\omega)\left(Q(\omega) + D(\omega)\right)^{-1} B^T(\omega)\right)^{-1} A(\omega)\right]$$

4.2 Q2: How do I use very expensive samples?

Bootstrap for stochastic optimization of energy systems

- Sampling the uncertainty present in complex energy systems may be a computationally intensive task
  - Example: weather forecasting
  - May need 400K CPUs for 30 samples at the resolution we need.

- Obtaining uncertainty estimates (confidence intervals) on the optimal value is important in policy-making process.

- Only a small number of samples(scenarios) can be afforded. Therefore an operational constraint makes me start to care about the low-sample size regime and its asymptotics.

- But how good is the current state of the theory in that regime?

Theory situation for Stochastic Programming

- Most estimates for SAA are based on results of the following type:

$$N^{0.5}\left[\frac{\theta-\hat{\theta}}{\hat{\sigma}}\right]\xrightarrow{D} N(0,1)$$

- This allows, in principle, for the convergence of the confidence intervals to be arbitrarily slow.
- Current state of the area is built around application of the Delta Theorem , which provides the results of the type:

$$X_N(\omega)-X(\omega)=o_P(N^{-a})\Leftrightarrow P\left(N^a\left|X_N(\omega)-X(\omega)\right|\right)\to 0$$

- But this is not sufficient for similar results for the confidence intervals !!!

$$X(\omega)=\omega,\quad X_N(\omega)=\begin{cases}-1, & 0\le\omega<\dfrac{1}{\log(N+1)}\\ \omega, & \dfrac{1}{\log(N+1)}\le\omega\le 1.\end{cases}\Rightarrow\begin{cases}P(\lim\limits_{N\to\infty} N^a\left|X_N(\omega)-X(\omega)\right|=0)=1\\ P(X_N(\omega)\le 0)-P(X(\omega)\le 0)=\dfrac{1}{\log(N+1)}\gg N^{-b}\end{cases}$$

- Intuition: Convergence in probability tells me how well I behave on a "good" set increasing to probability 1, but tells me nothing about the bad set.

Large Deviation + Bootstrap

- Bootstrap is a resampling method that builds high-order confidence estimates.
- The idea of bootstrapping is to squeeze out information from a small number of samples by resampling (with replacement).
- But it applies only to finite-dimensional functions of means, and the optimal value of stochastic optimization is not one (due to nonlinearity).

- Idea: Use large deviations, to produce exponentially convergent probability sets

$$\mathbb{P}(|N^b(\hat{\theta}-\theta)|>\epsilon) = f_1(\epsilon)N^{f_2(\epsilon)}\exp(-f_3(\epsilon)N^c)$$

- Here, $\hat{\theta}$ depends on the expected value of the objective function and its higher derivatives at the solution of the SAA approximation problem. We can thus use bootstrap theory to produce confidence intervals for $\hat{\theta}$ and exponential convergence ensures order stays the same as for bootstrap !!

The estimator

- We proposed a corrected statistic, computable at the SAA approximation

$$\Phi = \mathbb{E}\left[f(x^N) \right] - \frac{1}{2} \begin{pmatrix} \mathbb{E}\nabla L(x^N, \lambda^N) \\ 0 \end{pmatrix}^T \begin{pmatrix} \mathbb{E}\nabla^2 L(x^N, \lambda^N) & J(x^N)^T \\ J(x^N) & 0 \end{pmatrix}^{-1} \begin{pmatrix} \mathbb{E}\nabla L(x^N, \lambda^N) \\ 0 \end{pmatrix}$$

- L is the Lagrangian of the problem and J is the Jacobian of the constraints.
- x^N is the solution of the SAA problem obtained from a sample of size N.

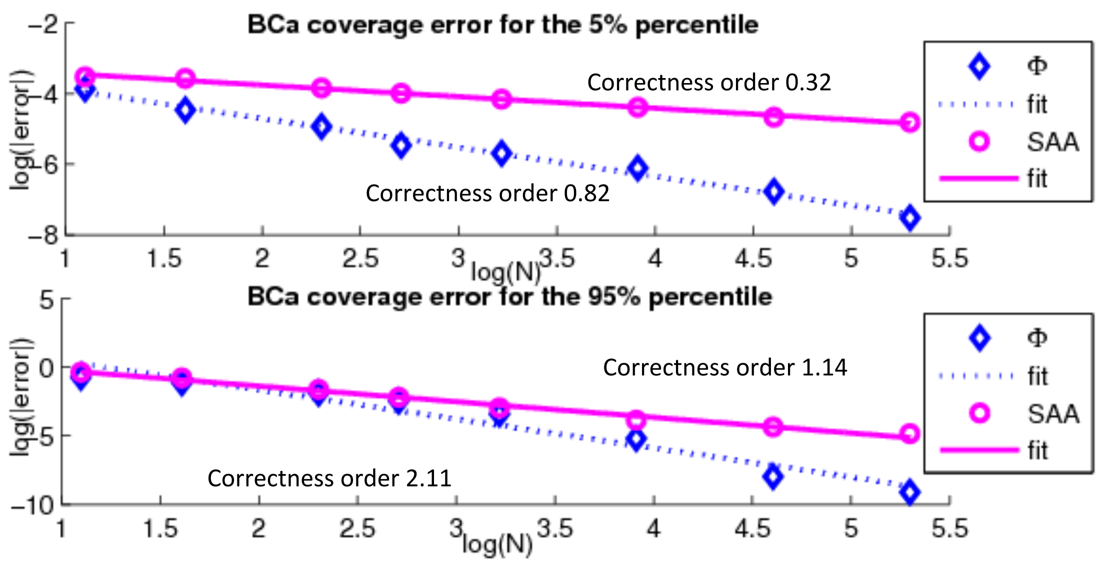
- Bootstrapping is performed based on a second sample of size M, and it works due to the fact that it is now applied at a set point x^N so finite dimensional results do apply.

Accuracy of the estimator's confidence levels

- We proved that bootstrap confidence intervals build using Φ are close to second order correct for the true optimal value $\theta = f(x^*)$:

$$P(\theta \in J_\alpha^b) = \alpha + O(N^{-1+a}), a > 0.$$

- We observed the predicted or better correctness in the numerical simulations



- bootstrapping Φ outperforms classical normal approximation method.
- We now have analytical techniques for asymptotics of confidence intervals in SP!

4.3 Q3: How do I interact with computationally intensive software-based models for stochastic optimization?

Our answer: Output learning with derivative information

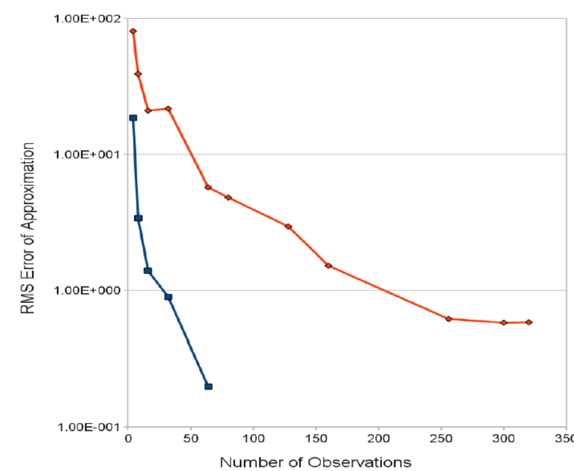
- Setup: outputs (100) << uncertain parameters (10^5-10^6)<< physical state space size (10^9-10^12).
- Outputs, e.g.
  - Coolant fuel temperature.
  - Peak fuel temperature
- Output (parameters)=? – it is a machine learning problem.
- The main observation: gradients of an output can be computed in 5*the effort (sometimes FAR less) for the output by adjoints, but provide NP times more information !!!
- This is radically different compared to machine learning from non-computing experiment data.
- Hypothesis: Using the derivative information, we can reduce the number of samples needed to compute the entire mapping.
- Hybrid sampling-sensitivity approach.

Polynomial Regression with Derivatives, PRD

- PRD procedure, regression equations:

$$\begin{pmatrix} \Psi_1(A_1) & \Psi_2(A_1) & \cdots \\ \frac{d\Psi_1(A_1)}{d\alpha_1} & \frac{d\Psi_2(A_1)}{d\alpha_1} & \cdots \\ \frac{d\Psi_1(A_1)}{d\alpha_2} & \frac{d\Psi_2(A_1)}{d\alpha_2} & \cdots \\ & \vdots & \\ \frac{d\Psi_1(A_1)}{d\alpha_m} & \frac{d\Psi_2(A_1)}{d\alpha_m} & \cdots \\ \Psi_1(A_2) & \Psi_2(A_2) & \cdots \\ \frac{d\Psi_1(A_2)}{d\alpha_1} & \frac{d\Psi_2(A_2)}{d\alpha_1} & \cdots \\ & \vdots & \\ \Psi_1(A_M) & \Psi_2(A_M) & \cdots \\ & \vdots & \\ \frac{d\Psi_1(A_M)}{d\alpha_m} & \frac{d\Psi_2(A_M)}{d\alpha_m} & \cdots \end{pmatrix} \cdot x = \begin{pmatrix} \Im(A_1) \\ \frac{d\Im(A_1)}{d\alpha_1} \\ \frac{d\Im(A_1)}{d\alpha_2} \\ \vdots \\ \frac{d\Im(A_1)}{d\alpha_m} \\ \Im(A_2) \\ \frac{d\Im(A_2)}{d\alpha_1} \\ \vdots \\ \Im(A_M) \\ \vdots \\ \frac{d\Im(A_M)}{d\alpha_m} \end{pmatrix}$$

- Note: the only interaction with the computationally expensive model is on the right side!
- We count on AD research to provide the adjoint information.
- Extra information from derivatives results in far fewer samples being needed to approximate system response!
- Raises a lot of new important mathematical questions that have not been studied (see Hickernell Poster)
  - How do I choose the basis?
  - How do I choose the sampling points?
  - What error model do I use?

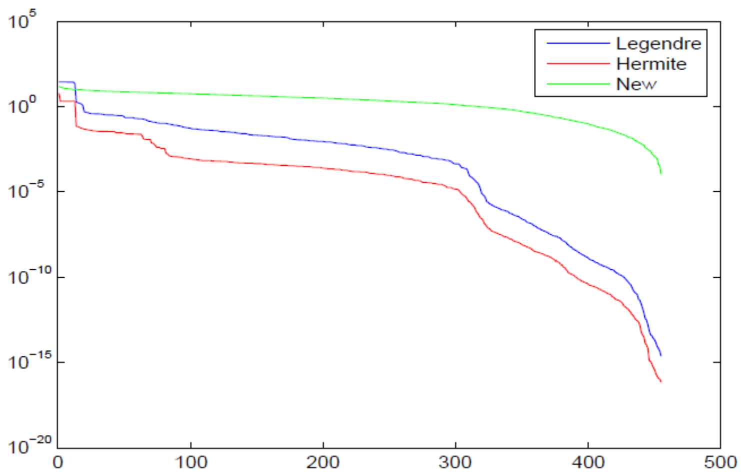


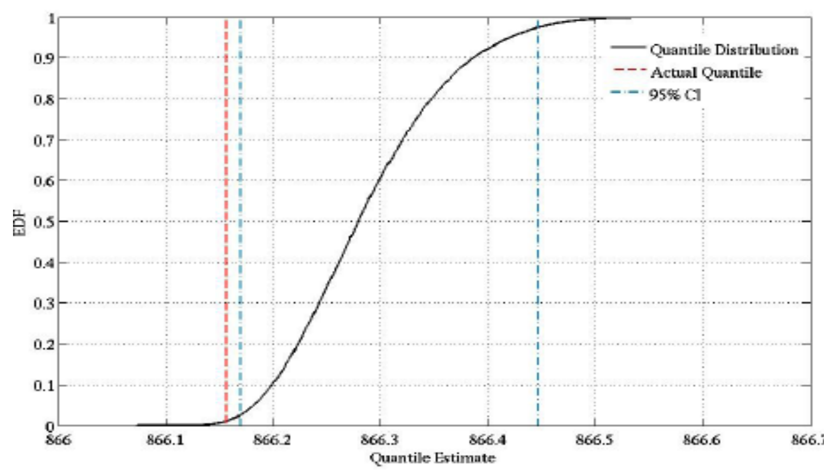
Some Results:

- Intuition: "good" basis should be orthogonal with respect to inner product that contains gradient information.

$$\int_{\Omega}\left(\Psi_j(A)\Psi_h(A)+\sum_{i=1}^{m}\frac{\partial\Psi_j(A)}{\alpha_i}\cdot\frac{\partial\Psi_h(A)}{\alpha_i}\right)\rho(A)dA=\delta_{ih}$$

- Intuition: Gaussian processes provide a good error model for regression (8 samples in 12 dimensions!!)
- For more results, see Hickernell poster.





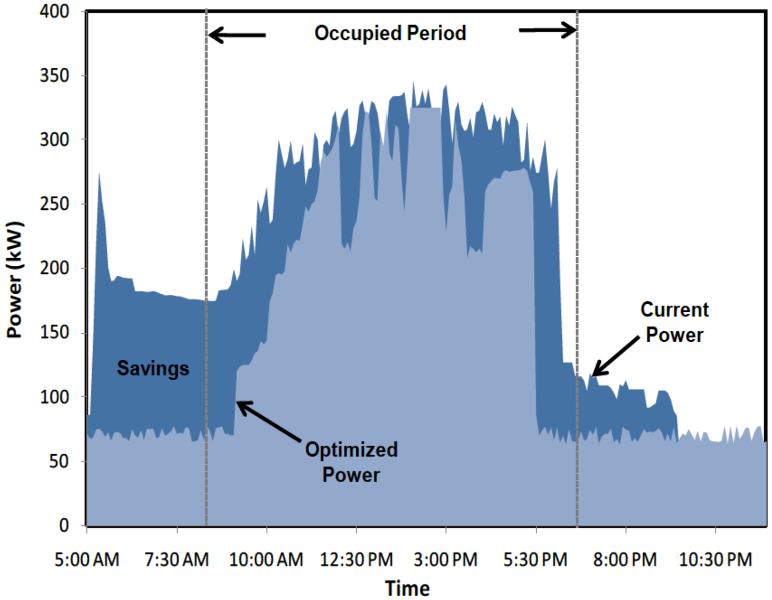
5. Secondary Impacts

New Platform: Optimization under Uncertainty for Next-Generation Building Systems

Deployment of Proactive Systems at Argonne National Laboratory

- Integrates New Sensors, Statistical Models, and Real-Time Optimization
- Tests in Commercial-Sized Building : ~ 500 Occupants, ~100,000 sq. ft.
- Energy Savings of Up to 30% in HVAC Energy Demand
    - Key: Anticipation Using Occupancy & Weather Information
    - Key: Adaptive Comfort and Equipment Conditions (Set-Points)
    - Key: Maximize Ambient Air Intake to Condition Building
    - Key: Provide algorithms for limited computational resources "in thermostat"

Predicted Energy Savings at Argonne's Building



Stochastic Real-Time Optimization for Building Systems

Minimize operation cost

Subject to: Comfort and Hardware Constraints.

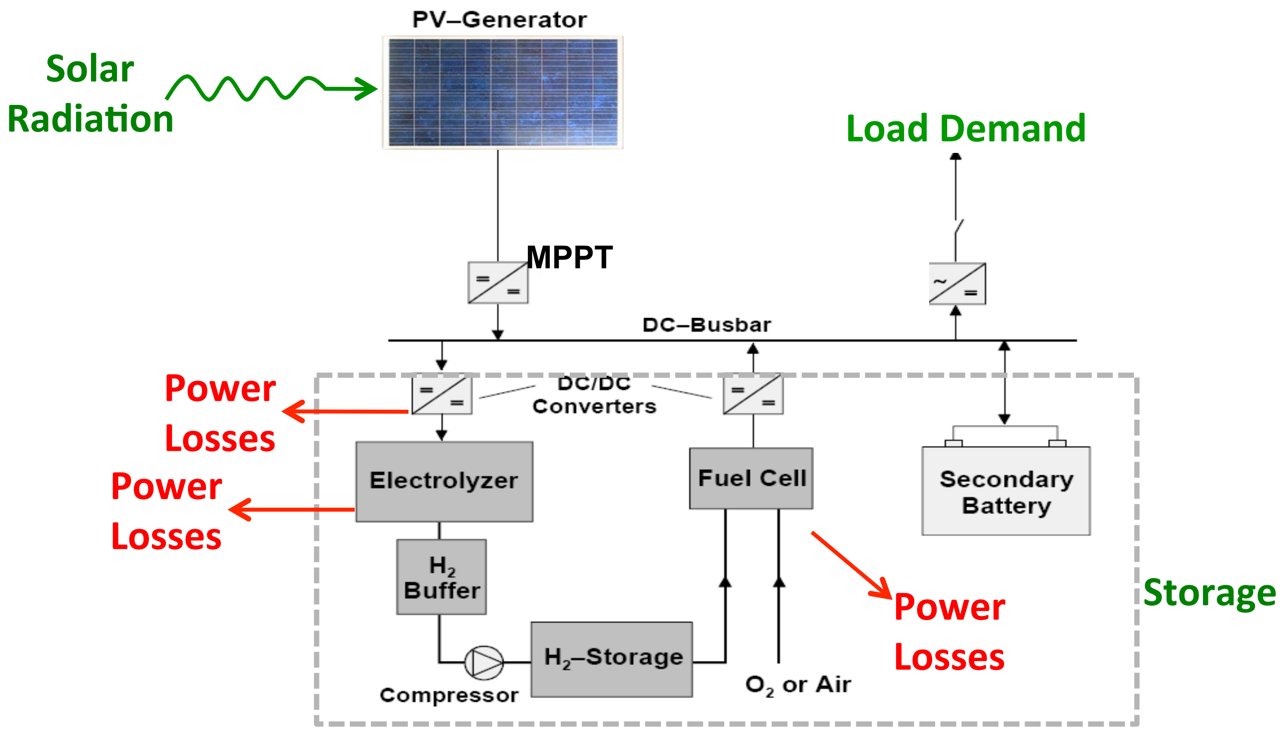
Uncertainty: occupancy, ambient conditions, state

Unusual Features:

- Data Analysis and Storage
(~1,000 Sensors per Building at High-Frequencies)
- Statistical and Gray-Box Modeling
(Minimize Technology Cost, Uncertainty Estimates)
- Resiliency Requirements
(Sustain Frequent Sensor and Equipment Faults)

SRTO-Latest Deployment (APS Building, August 23 2011)
15% Energy Savings (~ 1MWh) in First Day

Hybrid Photovoltaic-H_2 System



- **Operating Costs Driven by <u>Uncertain</u> Radiation** *Ulleberg, 2004*
  - **Performance Deteriorates by Multiple <u>Power Losses</u>**

Model:

Problem: Operate to Minimize Operating Costs +

Maximize H_2 Production

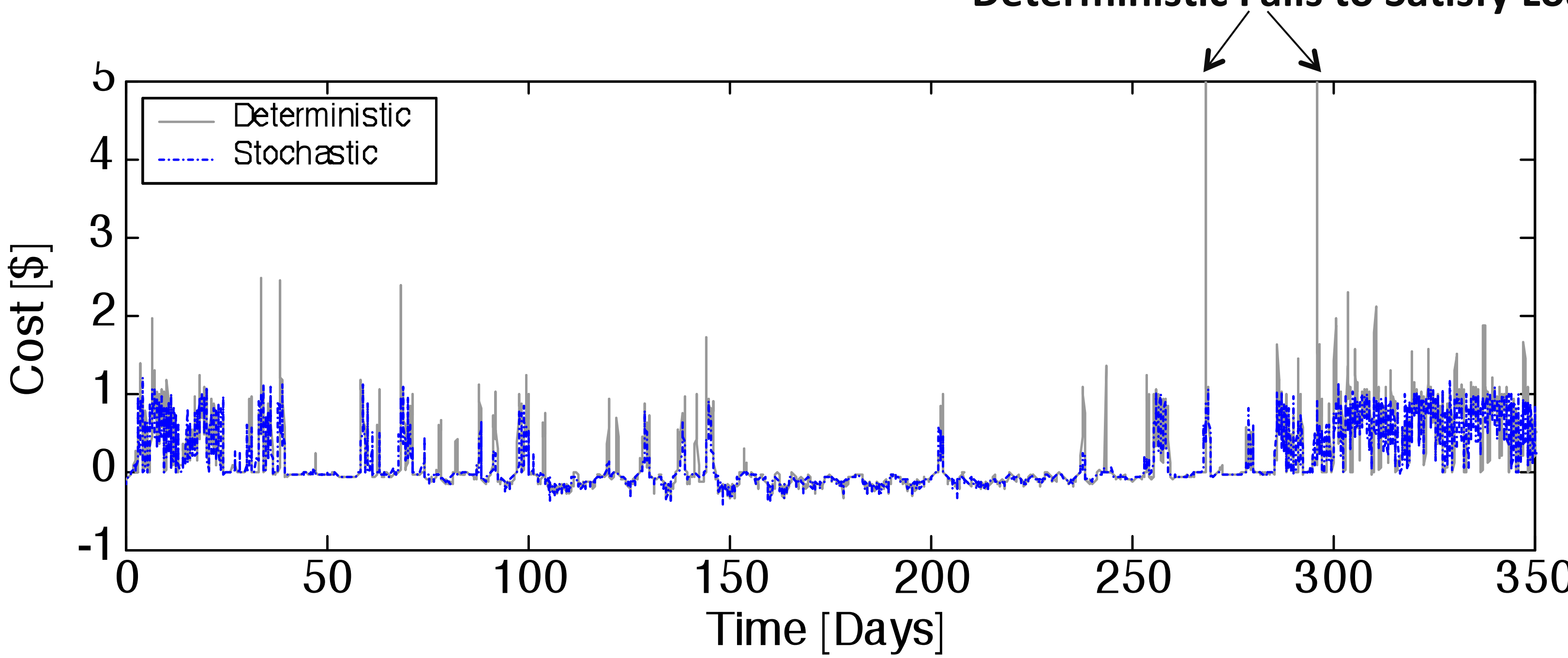
<u>Subject to:</u> Energy Balances; State-of-Charge, Fuel Cell

and Electrolyzer Limits

Hybrid Photovoltaic-H_2 System

Load Satisfaction Deterministic ("Optimization on Mean") vs. Stochastic

Deterministic Fails to Satisfy Load



Therefore, the alternative to stochastic programming can turn out infeasible !!

Handling Stochastic Effects Particularly Critical in Grid-Independent Systems

Conclusions

- Optimization under Uncertainty is a paradigm that is required by a broad array of applications.
- Its computational pattern is nontrivial, yet, very promising on new and probable architectures.
- We developed a scalable solver, PIPS, that scales strongly to up to 80% of BG/P for the power grid problems that require it.
- We resolved some of the probable bottlenecks by mathematical analysis, such as the impending loss of scalability due to the size of the master problems; and the slow convergence of the aposteriori error estimation in stochastic programming.
- Impact both highest end computing e.g. exascale and ubiquitous computing, e.g. your thermostat.

Future: How to deal with O(1000) larger problem (and new architectures?)

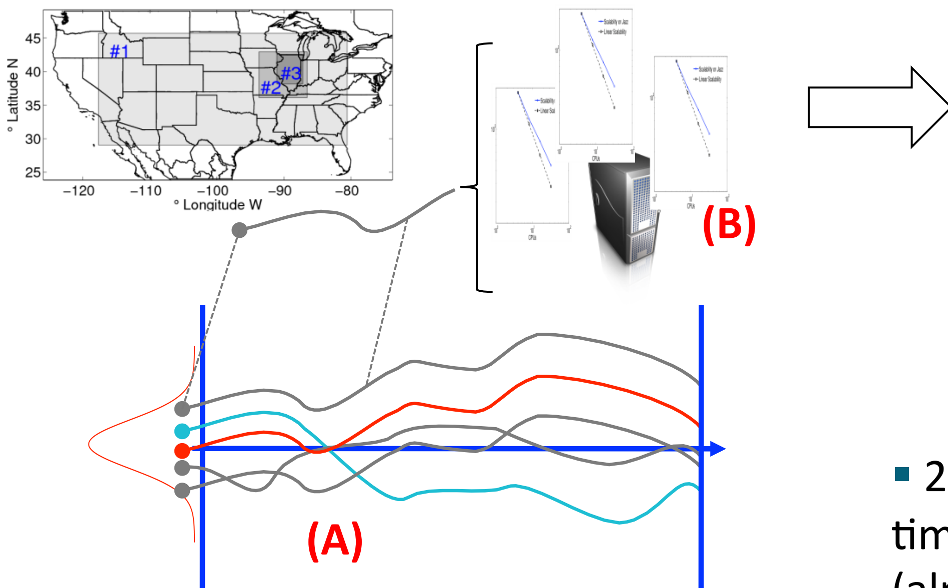
- Need modeling language for demonstrating the potential of stochastic programming for parallel computing on actual applications.
- More Asynchronous Algorithms (stochastic preconditioner a first step)
- Understand inference better for stochastic programming given the limitations of
  - Functional Convergence?
  - Resampling for infinite dimensional settings?
  - ….
- Extend approach to multistage stochastic programming?
- How to deal with integers for very large scale? Hybridize Simplex-Interior Point at exascale?

References

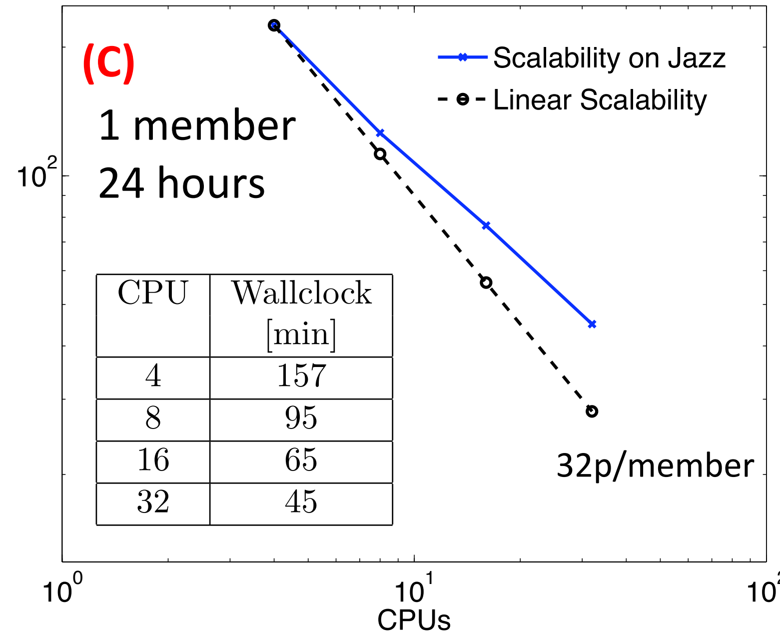
- E. Constantinescu, V. Zavala, M. Rocklin, S. Lee, M. Anitescu: *"Computational Framework for Uncertainty Quantification and Stochastic Optimization in Unit Commitment with Wind Power Generation",* IEEE Trans. Power Syst., 2009
- C.G. Petra, M. Anitescu: *"Preconditioning Schur complement systems for stochastic optimization"*, COAP (accepted), 2010.
- M. Lubin, C.G. Petra, M. Anitescu*: "Parallel dense linear solvers for stochastic optimization"*, OMS (in press), 2011.
- M. Lubin, C.G. Petra , M. Anitescu, V. Zavala: "*Scalable optimization of complex energy systems"*, SC11 (accepted), 2011.
- M. Anitescu, C.G. Petra: "*Using bootstrap for inference with higher-order confidence intervals in stochastic programming*", submitted to Mathematical Programming, 2011.
- Oleg Roderick, Mihai Anitescu, and Paul Fischer. "Stochastic finite element approaches using derivative information for uncertainty quantification" Preprint ANL/MCS 1551-1008. *Nuclear Science and Engineering,*Volume 164 (2), 2010, Pages 122-139.
- Yiou Li, Mihai Anitescu, Oleg Roderick and Fred Hickernell, "ORTHOGONAL BASES FOR POLYNOMIAL REGRESSION WITH DERIVATIVE INFORMATION IN UNCERTAINTY QUANTIFICATION". To appear in the *International Journal for Uncertainty Quantification*.
- Brian Lockwood and Mihai Anitescu. "Gradient-Enhanced Universal Kriging for Uncertainty Propagation". To appear in *Nuclear Science and Engineering*.

WRF scalability on Jazz

- Two-level parallelization scheme – very scalable: **(A)** realizations are independent, **(B)** each is parallelized, and **(C)** explicit



	Grid	Size
US:	#1 - 32 km^2	130×60
	#2 - 6 km^2	126×121
Illinois:	#3 - 2 km^2	202×232



CPU	Wallclock [min]
4	157
8	95
16	65
32	45

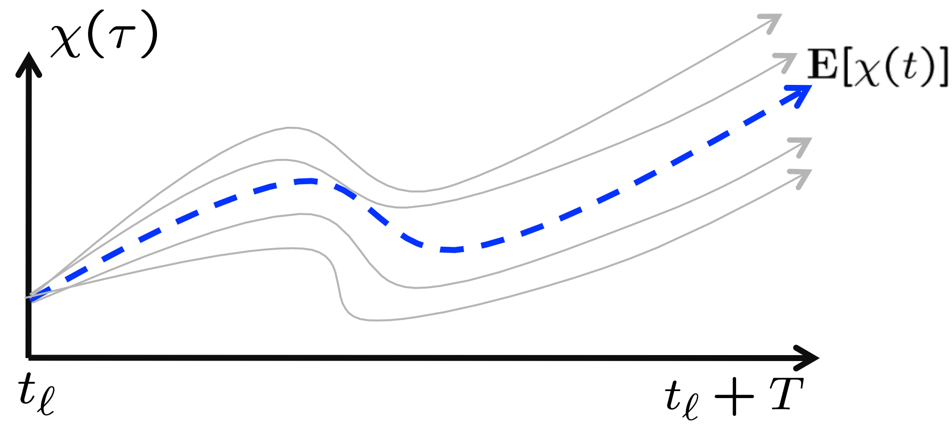
- 24 hours [simulation time] -> one hour [real time] on Jazz with 30 members; [2 km]; (almost) linear scalability with area **(C)**
  - ✔ Illinois [2km]: 500 processors
  - ☐ US [2 km]: ~50,000 processors
  - ☐ US [1 km]: ~400,000 processors

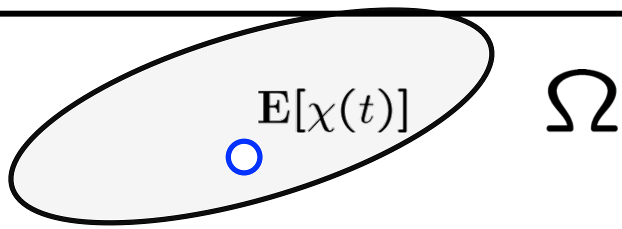
NLMPC Receding Horizon Optimization

Benefits: Accommodate Forecasts, Constraint Handling, Financial Objectives, Complex Models

Deterministic NLMPC

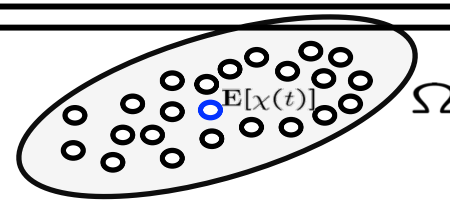
$$
\begin{aligned}
\min_{u(t)} \quad & \int_{t_\ell}^{t_{\ell+N}} \varphi(z(t), y(t), u(t), \mathbf{E}[\chi(t)])dt \\
\frac{dz}{dt} &= \mathbf{f}(z(t), y(t), u(t), \mathbf{E}[\chi(t)]) \\
0 &= \mathbf{g}(z(t), y(t), u(t), \mathbf{E}[\chi(t)]) \\
0 &\geq \mathbf{h}(z(t), y(t), u(t), \mathbf{E}[\chi(t)]) \\
z(0) &= x_\ell
\end{aligned}
$$





Stochastic NLMPC

$$
\begin{aligned}
\min_{u(t)} \quad & \mathbf{E}_{\chi(t)\in\Omega} \left[\int_{t_\ell}^{t_{\ell+N}} \varphi(z(t), y(t), u(t), \chi(t))dt \right] \\
\left.\begin{aligned}
\frac{dz}{dt} &= \mathbf{f}(z(t), y(t), u(t), \chi(t)) \\
0 &= \mathbf{g}(z(t), y(t), u(t), \chi(t)) \\
0 &\geq \mathbf{h}(z(t), y(t), u(t), \chi(t))
\end{aligned}\right\} & \forall \chi(t) \in \Omega \\
z(0) &= x_\ell
\end{aligned}
$$



Complexity (Solution Time)

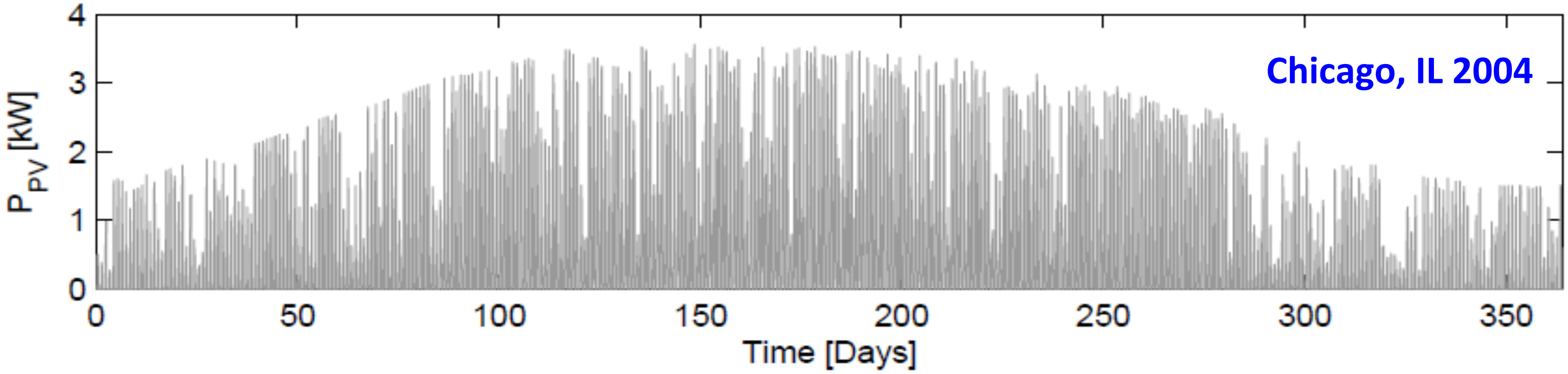
1,000 – 10,000 Differential-Algebraic Eqns

100-1000 Scenarios

First entry of control law implemented ➔ recede horizon ➔ restart

Hybrid Photovoltaic-H_2 System

Effect of Forecast on Economics *Z., Anitescu, Krause 2009*



True Future Radiation

$$\min_{u(t)} \int_{t_\ell}^{t_{\ell+N}} \varphi(z(t), y(t), u(t), \chi(t))dt$$

Minimize Operating Costs + Maximize H_2 Production

$$\frac{dz}{dt} = \mathbf{f}(z(t), y(t), u(t), \chi(t))$$

$$0 = \mathbf{g}(z(t), y(t), u(t), \chi(t))$$

Energy Balances

$$0 \geq \mathbf{h}(z(t), y(t), u(t), \chi(t))$$

State-of-Charge, Fuel Cell and Electrolyzer Limits

$$z(0) = x_\ell$$

- **Forecast Horizon of One Year – Highest Achievable Profit**
- **Receding-Horizon with 1hr, 1 Day, ...,14 Days Forecast - 8,700 Problems in Each Scenario**

Math Summary

- To deal with the impending bottleneck of the master problem we developed a stochastic preconditioner + Krylov instead of the direct solve. Proved exponential convergence to optimal preconditioning.
- To deal with the very expensive weather samples in aposteriori inference we removed the impreciseness of current technology in confidence intervals by using a resampling approach.
- $P(\theta \in \hat{J}) = P_{normal} + O(N^{-1+a}), a > 0.$

Parallelizing the 1st stage linear algebra

- We distribute the 1st stage Schur complement system.

$$C = \begin{bmatrix} \tilde{Q} & A_0^T \\ A_0 & 0 \end{bmatrix}, \quad \tilde{Q} \text{ dense symm. pos. def., } A_0 \text{ sparse full rank.}$$

- C is treated as dense.
- Alternative to PSC for problems with large number of 1st stage variables.
- Removes the memory bottleneck of PSC and DSC.
- We investigated ScaLapack, Elemental (successor of PLAPACK)
  - None have a solver for symmetric indefinite matrices (Bunch-Kaufman);
  - LU or Cholesky only.
  - So we had to think of modifying either.

Cholesky-based LDL^T -like factorization

- Can be viewed as an “implicit” normal equations approach.
- In-place implementation inside Elemental: no extra memory needed.
- Idea: modify the Cholesky factorization, by changing the sign after processing p columns.
- It is much easier to do in Elemental, since this distributes elements, not blocks.
- Twice as fast as LU
- Works for more general saddle-point linear systems, *i.e.,* pos. semi-def. (2,2) block.

Bootstrap for stochastic optimization of energy systems

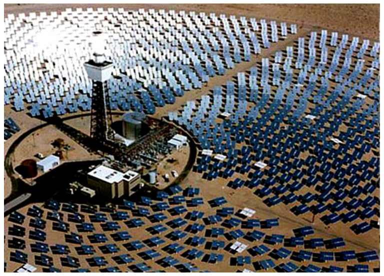
- Sampling the uncertainty present in complex energy systems may be a computationally intensive task
  - Example: weather forecasting
  - May need 400K CPUs for 30 samples at the resolution we need.

- Only a small number of samples(scenarios) can be afforded.
- Obtaining uncertainty estimates on the optimal value is important in policy-making process
- Classical statistical inference (e.g, building a 95% confidence interval J)
  - is only first order correct: $P(\theta \in J) = 0.95 + O(N^{-1/2})$
  - is unreliable for a small number of scenarios.

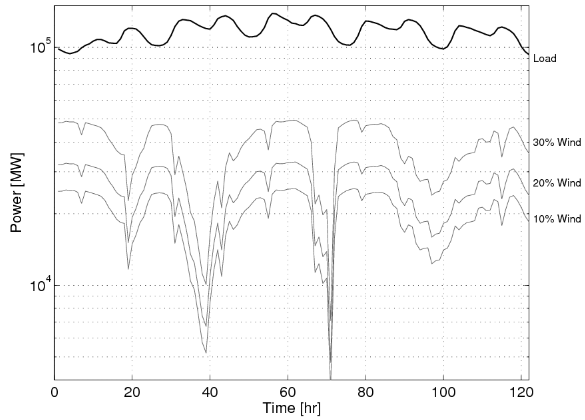
Ambient Condition Effects in Energy Systems

- Operation of 90% of Energy Systems is Affected by Ambient Conditions

- **Power Grid Management:** Predict Demands *(Douglas, et.al. 1999)*
- **Power Plants:** Production Levels *(General Electric)*
- **Petrochemical:** Heating and Cooling Utilities *(ExxonMobil)*
- **Buildings:** Heating and Cooling Needs *(Braun, et.al. 2004)*
- (Focus) **Next Generation Energy Systems** assume a major renewable energy penetration: Wind + Solar + Fossil *(Beyer, et.al. 1999)*



- But increased reliance on renewables must account for variability of ambient conditions, which cannot be done deterministically …
- We must optimize operational and planning decisions accounting for the uncertainty in ambient conditions (and other, e.g. load) –
- **Optimization Under Uncertainty.**

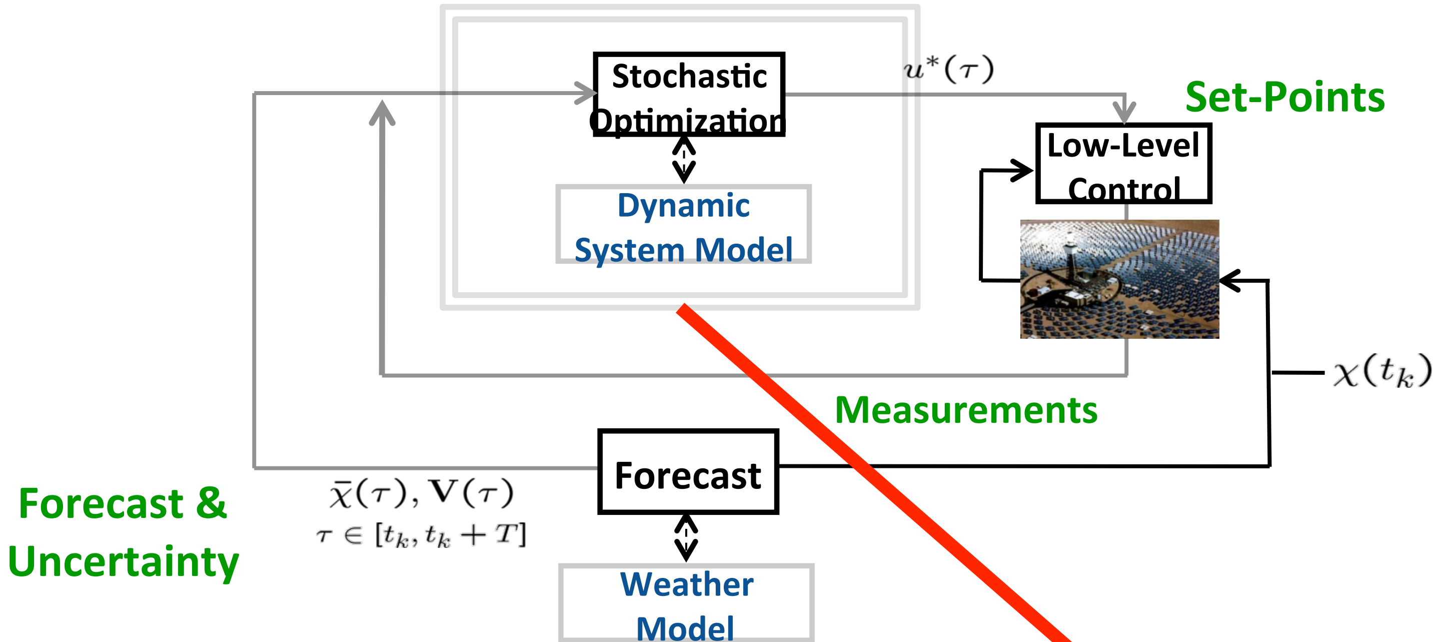


Wind Power Profiles

Other Optimization under Uncertainty Apps

- Increased Requirement for Uncertainty Quantification is Likely to Require Increased Emphasis on Optimization under Uncertainty.
- Possible Examples:
  - Optimal Nuclear Core Reloading (since fuel is reused in nuclear reactors).
  - Materials-by-Design accounting for model uncertainty.
  - Environmental Remediation – Nuclear Legacy Site Cleanup.
  - Whole Device Modeling for Fusion Applications
  - ……

Stochastic NLMPC



Two-stage Stoch Prog

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

subj. to.

$$\begin{aligned} g_0(x_0) &= b_0 \\ g_i(x_0, x_i) &= b_i \quad i = 1,2\ldots S \\ x_0 \geq 0, & \quad x_i \geq 0 \end{aligned}$$

Stochastic NLMPC

$$\min_{u(t)} \; \mathbf{E}_{\chi(t)\in\Omega} \left[\int_{t_\ell}^{t_\ell+N} \varphi(z(t), y(t), u(t), \chi(t))\,dt \right]$$

$$\left.\begin{aligned} \frac{dz}{dt} &= \mathbf{f}(z(t), y(t), u(t), \chi(t)) \\ 0 &= \mathbf{g}(z(t), y(t), u(t), \chi(t)) \\ 0 &\geq \mathbf{h}(z(t), y(t), u(t), \chi(t)) \end{aligned}\right\}$$

$$z(0) = x_\ell$$