Exceptional service in the national interest

Sandia National Laboratories

Reassessing the State-of-the-Art in Stochastic Unit Commitment Solvers

Dr. Bernard Knueven
Dr. Jean-Paul Watson
Sandia National Laboratories
Albuquerque, New Mexico USA

U.S. DEPARTMENT OF ENERGY
NNSA National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Key Collaborators

- Sandia National Laboratories
  - Andrea Staid
  - Cliff Hansen
- University of California Davis
  - David Woodruff
  - Dominic Yang
- Purdue University
  - Benjamin A. Rachunok
- University of Tennessee
  - Jim Ostrowski

A Word From Our Sponsors…

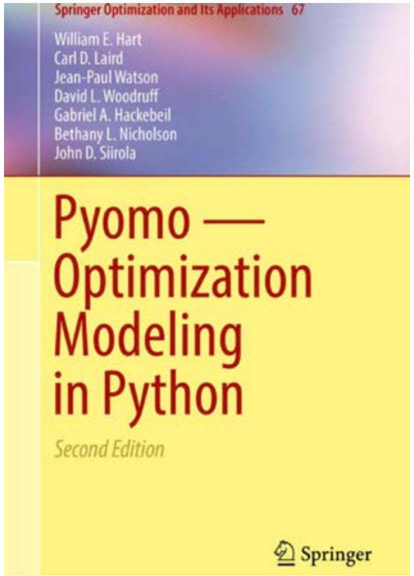
- DOE Grid Modernization Laboratory Consortium (GMLC)
  - Project 1.4.26 – Multi-Scale Production Cost Modeling
- Bonneville Power Administration (BPA)
  - Funded work on high-accuracy probabilistic wind forecasting
  - Provide real-world data sets, publicly available
- Department of Energy’s ARPA-E office
  - Scalable stochastic unit commitment project

High-Level Talk Goals

- Cover recent developments in stochastic UC solvers
- Convince you that the oft-repeated statement “stochastic UC is prohibitively computationally difficult” is unfounded
- Quickly highlight some work on developing a range of stochastic UC benchmarks

All of Our Research is Built on and Enabled by…

- Project homepage
  - www.pyomo.org
- “The Book”

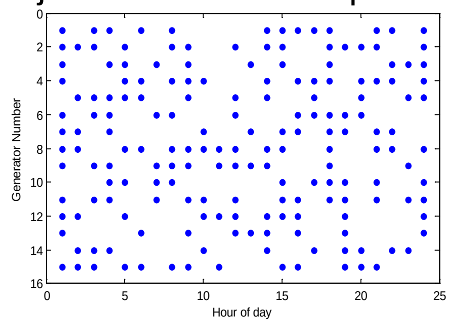


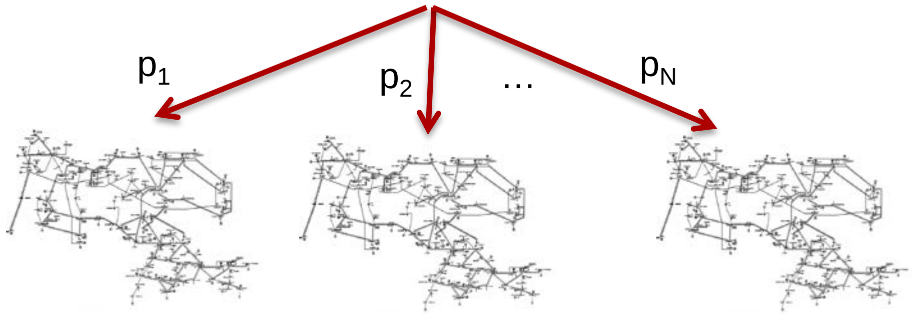
- Mathematical Programming Computation papers
  - Pyomo: Modeling and Solving Mathematical Programs in Python (Vol. 3, No. 3, 2011)
  - PySP: Modeling and Solving Stochastic Programs in Python (Vol. 4, No. 2, 2012)

Refresher: The General Structure of a Stochastic Unit Commitment Optimization Model

Sandia National Laboratories

Objective: Minimize expected cost





Scenario 1 Scenario 2 … Scenario N

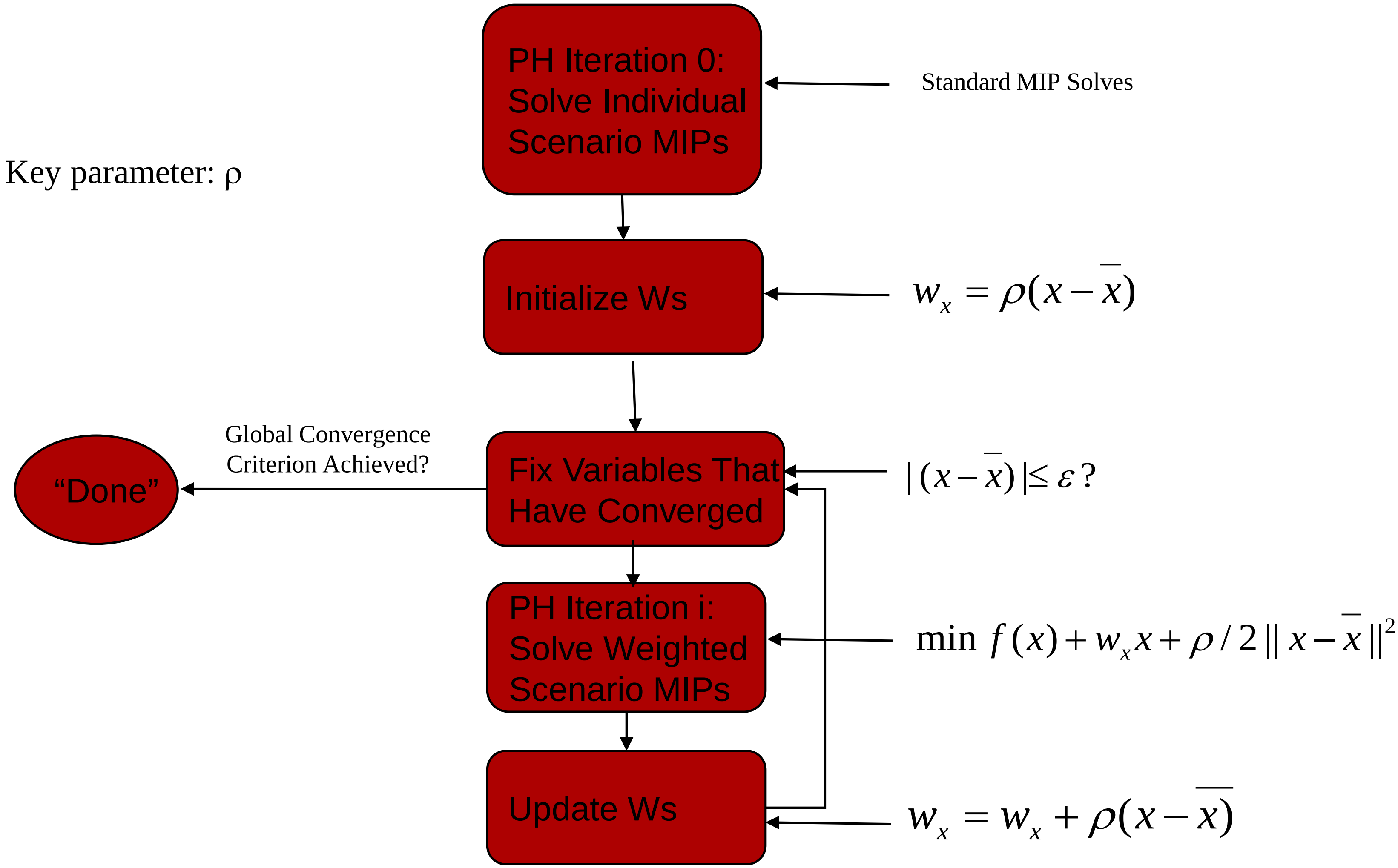
First stage variables:
- Unit On / Off

Nature resolves uncertainty
- Load
- Renewables output
- Forced outages

Second stage variables (*per time period*):
- Generation levels
- Power flows
- Voltage angles
- …

*Renewables are **not** modeled as must-take, allowing for curtailment without penalty*

Scenario-Based Decomposition via Progressive Hedging (PH)



Progressive Hedging: Some Algorithmic Issues and their Resolution

- We are dealing with mixed-integer programs
  - So we have to deal with the possibility of cycling and other manifestations of non-convergence
  - See: *Progressive Hedging Innovations for a Class of Stochastic Mixed-Integer Resource Allocation Problems*, J.P. Watson and D.L. Woodruff, Computational Management Science, Vol. 8, No. 4, 2011
- Good values for the ρ parameter are critical
  - Poor or ad-hoc values of ρ can lead to atrocious performance
  - The good news in unit commitment
    - We have a lot of information concerning the cost of using a generator
    - Cost-proportional rho is a known, effective strategy in Progressive Hedging
  - Also see Computational Management Science paper indicated above

A Novel Matching Formulation for Startup Cost in Unit Commitment

- Thermal generators have time-dependent startup costs, which are increasing the length of the generator’s off-time.
  - Need more energy to start cold thermal units than warm ones
- Can be modelled by appending additional variables and constraints to the MILP formulation for a thermal generator
- Contributions:
  - We introduce a new formulation for time-dependent startup costs.
  - We place the existing formulations into a formal dominance hierarchy based on their relative tightness.
  - We compare the effectiveness of the various formulations on large-scale unit commitment instances

A Novel Matching Formulation for Startup Cost in Unit Commitment

Idea: Introduce new variables $x(t, t')$ to match shutdowns $w(t)$ with with startups $v(t')$.

$$\sum_{t'=t-TC+1}^{t-TD} x(t', t) \leq v(t)$$

$$\sum_{t'=t+TD}^{t+TC-1} x(t, t') \leq w(t)$$

Startup costs are then calculated by putting appropriate coefficients on $v(t)$ and $x(t, t')$ in the objective function.

Comparing UC Formulations

- EF (Extended Formulation): based on shortest-path polytope (Pochet 2006)
- Match: our contribution
- STI (Startup Type Indicator): from Simoglou et al. (2010) and Morales-España et al. (2013)
- 3-bin: introduced to ease proofs, needs fewer variables than STI
- 1-bin*: strengthened 1bin from Silbernagl et al. (2015)
- 1-bin: from Carrion and Arroyo (2006)

Comparing Formulations

Formulation	# variables	# constraints
EF	$O(T^2)$	$O(T)$
Match	$O((TC - DT) \cdot T)$	$O(T)$
STI	$O(S \cdot T)$	$O(S \cdot T)$
3-bin	$O(T)$	$O(S \cdot T)$
1-bin*	$O(T)$	$O(S \cdot T)$
1-bin	$O(T)$	$O(S \cdot T)$

- T is the number of time periods
- S is the number of startup types
- TC is the time after which the generator goes cold
- DT is the generator's minimum downtime

- Match needs more variables than the other formulations besides EF.
- However, the strength of the formulations from strongest to weakest is EF -> Match -> STI -> 3-bin -> 1-bin* -> 1bin.

Computational Results: CAISO Instances

Instance	EF	Match	STI	3-bin	1-bin*	1-bin
2014-09-01 0%	357.21	**30.24**	46.80	53.55	(0.028%)	(0.041%)
2014-12-01 0%	169.30	23.89	**23.06**	65.81	(0.073%)	(0.068%)
2015-03-01 0%	166.04	24.68	41.76	**16.46**	(0.042%)	(0.053%)
2015-06-01 0%	163.38	**12.64**	18.76	24.74	(0.017%)	(0.020%)
Scenario400 0%	335.67	**26.60**	65.02	173.37	(0.403%)	(0.383%)
2014-09-01 1%	462.78	**20.39**	22.44	31.83	(0.055%)	(0.045%)
2014-12-01 1%	381.80	36.43	**28.90**	85.78	(0.072%)	(0.069%)
2015-03-01 1%	178.40	**20.41**	35.08	67.20	(0.079%)	(0.090%)
2015-06-01 1%	274.25	41.60	**39.03**	70.83	(0.020%)	(0.028%)
Scenario400 1%	(0.012%)	**46.08**	83.29	182.19	(0.376%)	(0.446%)
2014-09-01 3%	598.73	75.69	**63.26**	87.48	(0.043%)	(0.036%)
2014-12-01 3%	(0.011%)	63.64	**54.88**	93.39	(0.083%)	(0.087%)
2015-03-01 3%	217.10	**48.91**	73.06	99.57	(0.112%)	(0.110%)
2015-06-01 3%	329.79	84.66	**38.13**	83.26	(0.024%)	(0.022%)
Scenario400 3%	(0.013%)	**129.50**	243.01	356.10	(0.495%)	(0.538%)
2014-09-01 5%	412.24	46.80	**44.92**	119.49	(0.037%)	(0.037%)
2014-12-01 5%	(0.012%)	**86.41**	107.14	113.69	(0.104%)	(0.082%)
2015-03-01 5%	(0.010%)	**83.49**	87.22	94.95	(0.115%)	(0.105%)
2015-06-01 5%	(0.010%)	**28.28**	66.97	151.47	(0.031%)	(0.031%)
Scenario400 5%	(0.014%)	115.02	**107.02**	(0.014%)	(0.514%)	(0.570%)
Geometric Mean:	>370.5	**43.12**	52.84	>91.43	>600	>600

- Extra variables in Match do not increase its difficulty
  - Match closes 50-90% of the integrality gap relative to STI
- Overall 18% reduction in solve time on average over STI

Computational Results: FERC Instances

- Low wind (2%) instances
  - 30% mean reduction in solve time over STI
  - But only close 8% of the root gap over STI on average
- High wind (30%) instances
  - 45% reduction in solve time over STI
  - Close 34% of the root gap over STI on average
  - Most Match variants still solve at root node

(a) 2% Wind Penetration

Instance	EF	Match	STI	3bin	1bin*	1bin
2015-01-01	511.91	**111.34**	193.07	241.63	(0.017%)	(0.046%)
2015-02-01	586.95	**85.12**	314.07	463.66	(0.143%)	(0.172%)
2015-03-01	807.3	**152.24**	177.44	245.77	649.54	596.24
2015-04-01	(0.012%)	190.62	321.6	**177.27**	675.45	660.92
2015-05-01	512.55	**177.51**	191.29	186.68	334.17	416.03
2015-06-01	619.8	142.57	**139.16**	211.92	406.68	575.42
2015-07-01	(0.017%)	411.00	491.22	**260.41**	(0.014%)	901.87
2015-08-01	808.34	**113.13**	350.52	449.67	(0.11%)	(0.165%)
2015-09-01	(0.016%)	313.79	**284.31**	840.5	(0.101%)	(0.113%)
2015-10-01	605.11	132.95	**113.69**	133.48	582.63	582.58
2015-11-02	573.13	**109.88**	200.83	209.22	(0.073%)	(0.136%)
2015-12-01	(0.013%)	116.25	**114.15**	242.18	(0.055%)	(0.105%)
Geometric Mean:	>701.7	**153.60**	218.36	266.53	>711.7	>739.1

(b) 30% Wind Penetration

Instance	EF	Match	STI	3bin	1bin*	1bin
2015-01-01	712.53	**127.22**	(0.902%)	(1.334%)	(4.083%)	(3.808%)
2015-02-01	612.38	**114.78**	(0.043%)	(0.158%)	(0.952%)	(0.959%)
2015-03-01	895.97	647.78	**480.77**	496.35	(0.386%)	(0.460%)
2015-04-01	(0.024%)	**140.82**	236.23	425.71	(0.276%)	(1.054%)
2015-05-01	(0.016%)	**104.62**	119.06	110.24	312.33	337.55
2015-06-01	698.12	222.54	141.18	**110.06**	(0.408%)	(0.101%)
2015-07-01	(0.015%)	**126.98**	346.18	230.15	(0.222%)	(0.105%)
2015-08-01	(0.019%)	395.87	379.42	**227.92**	(0.768%)	(0.870%)
2015-09-01	(0.012%)	**245.73**	780.9	(0.035%)	(0.254%)	(0.256%)
2015-10-01	(0.036%)	439.03	**352.54**	533.72	617.14	607.19
2015-11-02	789.73	**182.40**	618.73	782.18	(0.803%)	(1.065%)
2015-12-01	674.84	**312.67**	361.35	421.08	(0.035%)	(0.035%)
Geometric Mean:	>808.1	**214.70**	>390.5	>401.1	>799.3	>803.9

Toward Scalable Stochastic Unit Commitment

Part 1: Load Scenario Generation

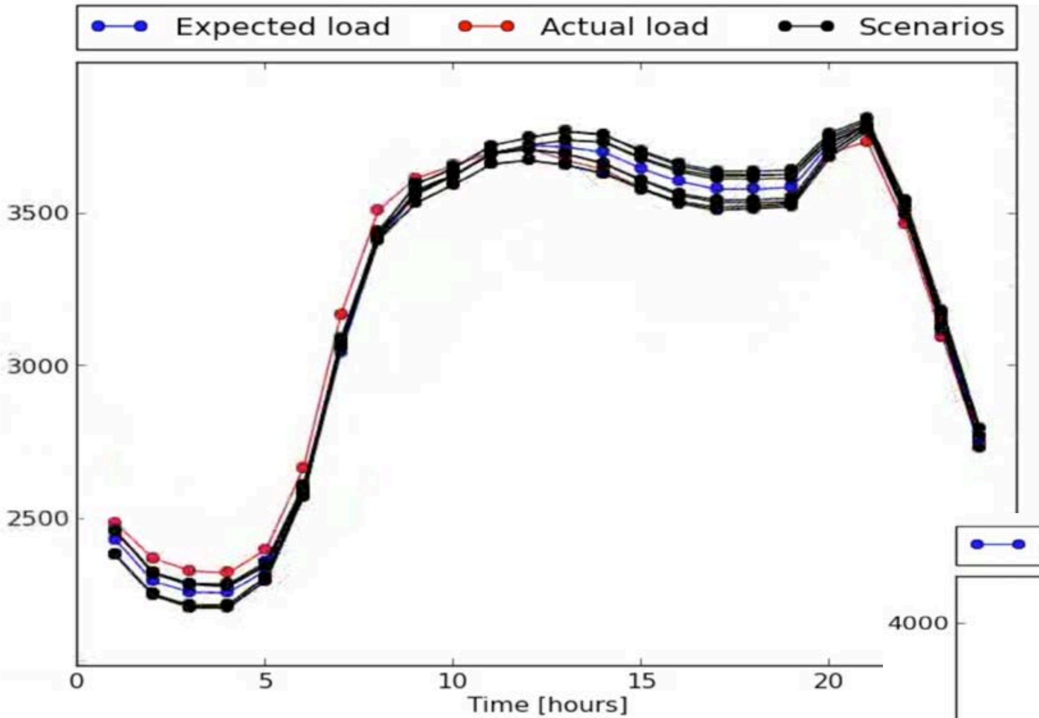
Yonghan Feng · Ignacio Rios · Sarah M. Ryan · Kai Spürkel · Jean-Paul Watson · Roger J-B Wets · David L. Woodruff

Revised: November 15, 2014

Abstract Unit commitment decisions made in the day-ahead market and during subsequent reliability assessments are critically based on forecasts of load. Traditional, deterministic unit commitment is based on point or expectation-based load forecasts. In contrast, stochastic unit commitment relies on multiple load scenarios, with associated probabilities, that in aggregate capture the range of likely load time-series. The shift from point-based to scenario-based forecasting necessitates a shift in forecasting technologies, to provide accurate inputs to stochastic unit commitment. In this paper, we discuss a novel scenario generation methodology for load forecasting in stochastic unit commitment, with application to real data associated with the Independent System Operator for New England (ISO-NE). The accuracy of the expected scenario generated using our methodology is consistent with that of point forecasting methods. The resulting sets of realistic scenarios serve as input to rigorously test the scalability of stochastic unit commitment solvers, as described in the companion paper. The scenarios generated

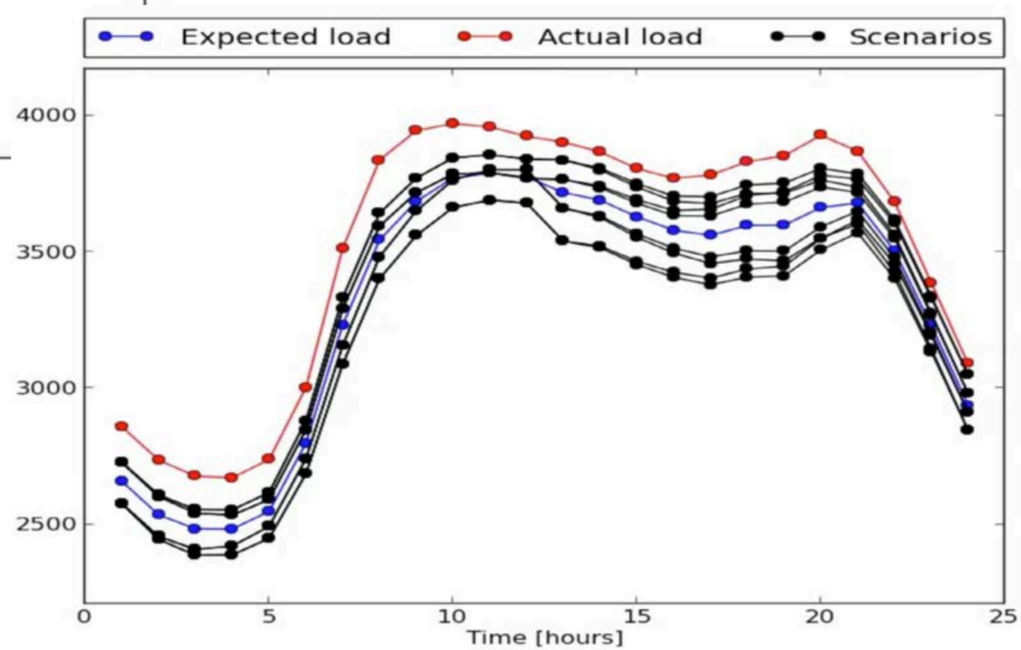
In Energy Systems (2015)

Probabilistic Load Scenarios



If the historical data indicates no variability, then the scenarios will reflect that consistency

Captures variability in load when present – but predictions are not perfect!



Toward Scalable Stochastic Unit Commitment

Part 2: Solver Configuration and Performance Assessment

Kwok Cheung · Dinakar Gade · César Silva-Monroy · Sarah M. Ryan · Jean-Paul Watson · Roger J.-B. Wets · David L. Woodruff

Received: April 30, 2014

Abstract In this second portion of a two-part analysis of a scalable computational approach to stochastic unit commitment, we focus on solving stochastic mixed-integer programs in tractable run-times. Our solution technique is based on Rockafellar and Wets' progressive hedging algorithm, a scenario-based decomposition strategy for solving stochastic programs. To achieve high-quality solutions in tractable run-times, we describe critical, novel customizations of the progressive hedging algorithm for stochastic unit commitment. Using a variant of the WECC-240 test case with 85 thermal generation units, we demonstrate the ability of our approach to solve realistic, moderate-scale stochastic unit commitment problems with reasonable numbers of scenarios in no more than 15 minutes of wall clock time on commodity compute platforms. Further, we demonstrate that the resulting solutions are high-quality, with costs typically within 1-2.5% of optimal. For

In Energy Systems (2015)

Stochastic UC Results: A Refresh

Table 10 Solve time (in seconds) and solution quality statistics for PH executing on the *WECC-240-r1* instance, with $\alpha = 0.5$, $\mu = 3$, and the MTR deterministic UC model.

# Scenarios	Convergence Metric	Obj. Value	PH L.B.	# Vars Fx.	Time
		64-Core Workstation Results			
3	0.0 (in 36 iters)	64141.771	64109.021	4080	237
5	0.0 (in 23 iters)	62628.532	62499.212	4080	161
10	0.0 (in 26 iters)	61384.016	61327.734	4080	215
25	0.0 (in 41 iters)	60927.903	60850.717	4080	366
50	0.0 (in 11 iters)	60617.311	60470.956	4044	318

Results generated circa 2013 (published 2015)

New UC model, fewer PH tweaks, and new persistent solver interfaces in PySP

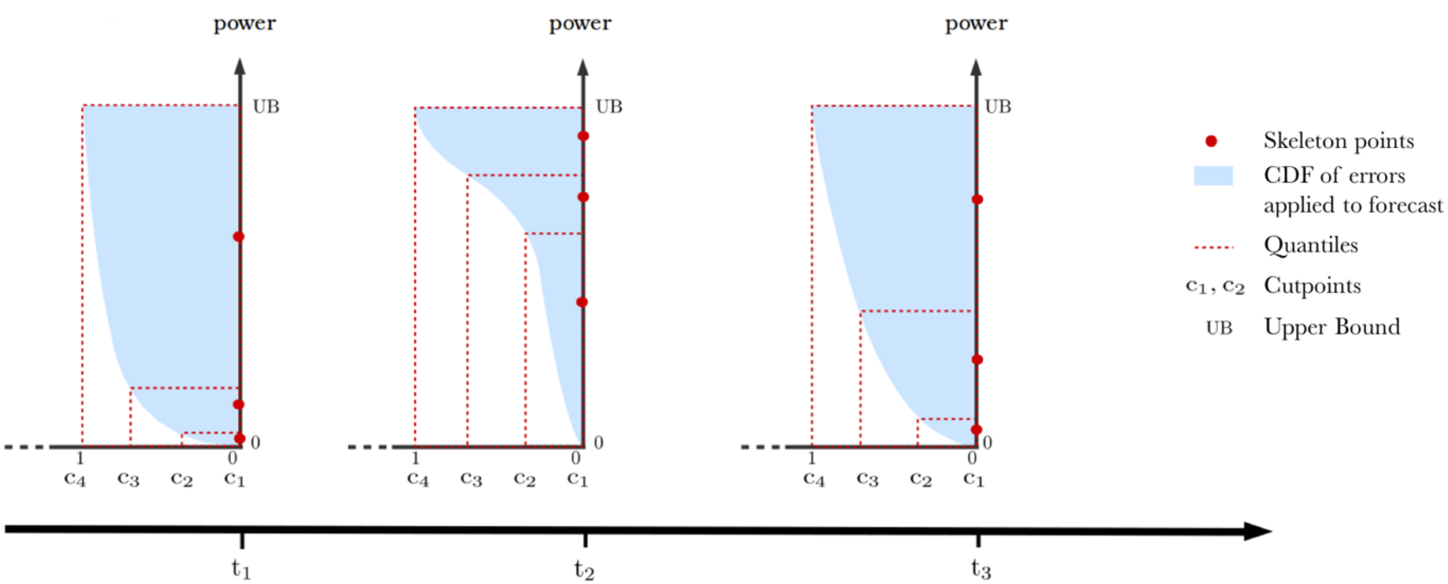
# Scenarios	Convergence Metric	Obj. Value	PH L.B.	Time
3	0.0 (in 5 iters)	64156.14	64107.06	41 s.
5	0.0011 (in 20 iters)	62669.10	62612.79	127 s.
100	0.0 (in 8 iters)	61386.90	61349.97	105 s.
25	0.0 (in 11 iters)	60933.85	60883.27	167 s.
50	0.0 (in 9 iters)	60618.77	60577.50	207 s.

Results generated March 2018

Note: All times are with out-of-the-box Pyomo

Epi-Spline-Based Scenario Creation

- For a subset of hours in day (i.e., hours 1, 12, 24), calculate empirical **forecast error** CDF from relevant* historical forecast/actual pairs
  - Correlations in forecast error drop off quickly with time, allowing for independent calculations
- Divide distribution at cut points, and calculate the weighted average of the distribution between each cut point pair
- Apply error value to next-day forecast to obtain scenario value



Sandia National Laboratories

RESEARCH ARTICLE

Generating Short-Term Probabilistic Wind Power Scenarios via Non-Parametric Forecast Error Density Estimators

Andrea Staid[1], Jean-Paul Watson[1], Roger J.-B. Wets[2], and David L. Woodruff[2]

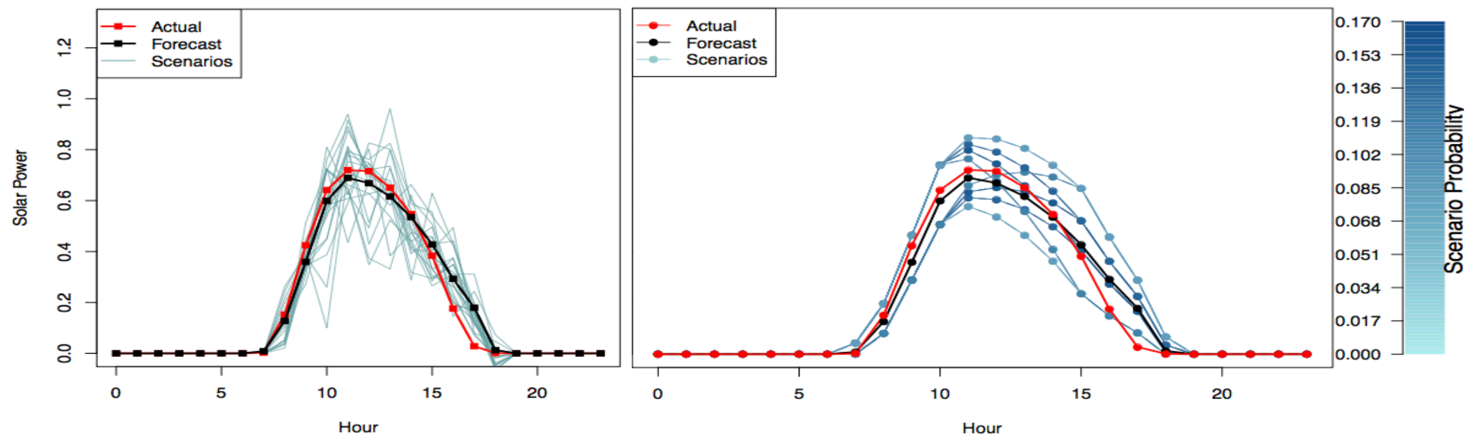
[1] Sandia National Laboratories, Albuquerque, New Mexico, USA
[2] University of California Davis, Davis, California, USA

ABSTRACT

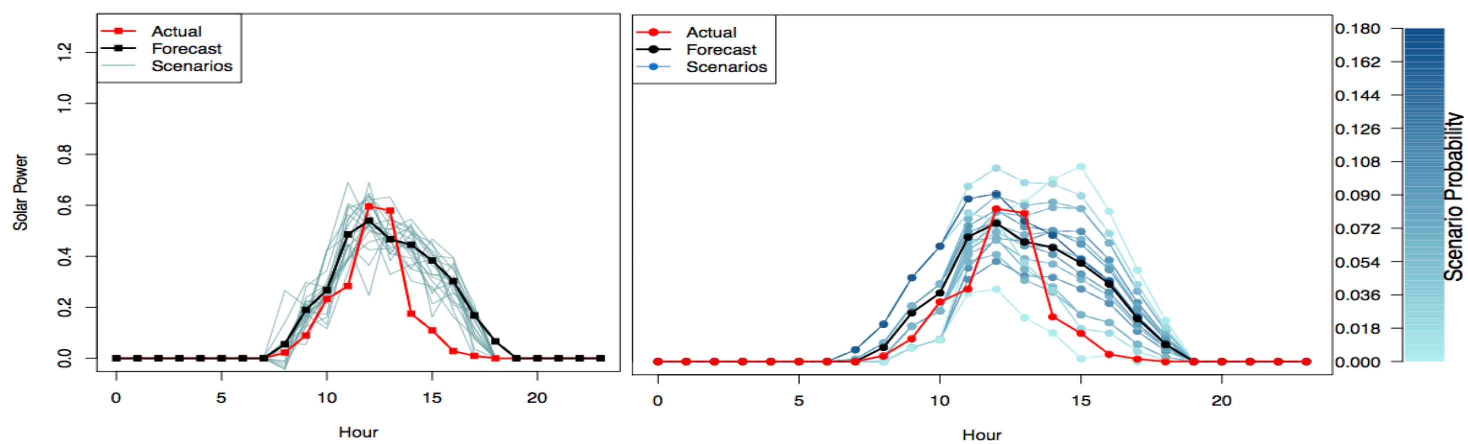
Forecasts of available wind power are critical in key electric power systems operations planning problems, including economic dispatch and unit commitment. Such forecasts are necessarily uncertain, limiting the reliability and cost-effectiveness of operations planning models based on a single deterministic or “point” forecast. A common approach to address this limitation involves the use of a number of probabilistic scenarios, each specifying a possible trajectory of wind power production, with associated probability. We present and analyze a novel method for generating probabilistic wind power scenarios, leveraging available historical information in the form of forecasted and corresponding observed wind power time series. We estimate non-parametric forecast error densities, specifically using epi-spline basis functions, allowing us to capture the skewed and non-parametric nature of error densities observed in real-world data. We then describe a method to generate probabilistic scenarios from these basis functions that allows users to control for the degree to

In Wind Energy (2017)

Probabilistic (Bulk) Solar Scenarios



(a) 2013-05-09 (b) 2013-05-09



(c) 2013-05-15 (d) 2013-05-15

Constructing Probabilistic Scenarios for Wide-Area Solar Power Generation

David L. Woodruff

Graduate School of Management, University of California, Davis, CA 95616-8609, USA

Julio Deride, Andrea Staid, Jean-Paul Watson

Discrete Math and Optimization Department, Sandia National Laboratories, Albuquerque, NM 87185, USA

Gerrit Slevogt

Department of Mathematics, University of Duisburg-Essen, Germany

César Silva-Monroy

Demand Energy, Liberty Lake, WA 99019, USA

Abstract

Optimizing thermal generation commitments and dispatch in the presence of high penetrations of renewable resources such as solar energy requires a characterization of their stochastic properties. In this paper, we describe novel methods designed to create day-ahead, wide-area probabilistic solar power scenarios based only on historical forecasts and associated observations of solar power production. Scenarios are created by segmentation of historic data, fitting non-parametric error distributions using epi-splines, and then computing specific quantiles from these distributions. Additionally, we address the challenge of establishing an upper bound on solar power output. Our specific application driver is for use in stochastic variants of core power systems operations optimiza-

Under review (revision) with Solar Energy

High-Quality Scenarios Yields Very Difficult Stochastic UCs

- There is a remarkably and unexpectedly strong correlation between the nature of probabilistic scenarios for renewables and difficulty of solution via decomposition (PH) algorithms

Stochastic Unit Commitment Performance Considering Monte Carlo Wind Power Scenarios

Benjamin Rachunok
Purdue University, West Lafayette IN

David L. Woodruff and Dominic Yang
University of California Davis, Davis, CA

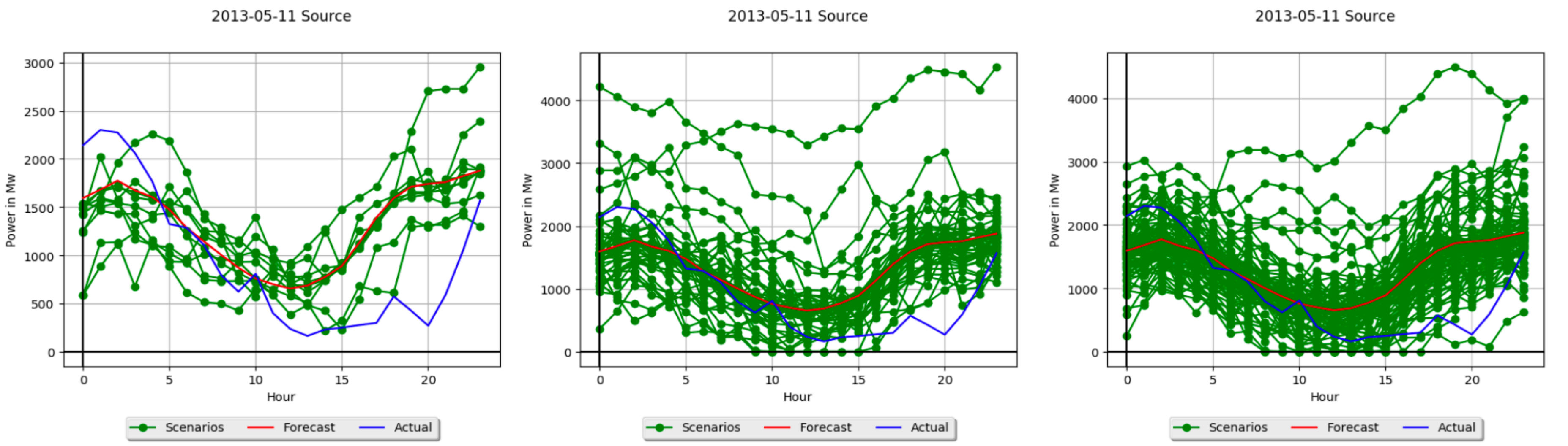
Andrea Staid and Jean-Paul Watson
Sandia National Laboratories, Albuquerque, NM

***Abstract*—Stochastic versions of the unit commitment problem have been advocated for addressing the uncertainty presented by high levels of wind power penetration. However, little work has been done to study trade-offs between computational complexity and the quality of solutions obtained as the number of probabilistic scenarios is varied. Here, we describe extensive experiments using real publicly available wind power data from the Bonneville Power Administration. Solution quality is measured by re-enacting day-ahead reliability unit commitment and real-time**

While researchers have devoted significant effort to developing scalable approaches to stochastic UC, relatively little effort has been devoted to studies that examine the performance of stochastic UC in the context of power system simulations, of which stochastic UC is only one component. In particular, the research literature lacks studies to provide insights into the following question of practical importance: "How many

To appear in Probabilistic Methods Applied to Power Systems (PMAPS) Conference - 2018

On Diversity of Renewables Scenarios and Problem Difficulty



(a) 10 scenarios (b) 50 scenarios (c) 100 scenarios

Figure 1: Probabilistic BPA wind power scenarios for May 11, 2013, considering 10, 50, and 100 equi-probable realizations.

Iteration counts and solution times (wall clock) by PH, considering WECC-240++ generator fleet

(a) 7 iterations – ~120 seconds

(b) 100 iterations (limit) – ~600 seconds (!)

(c) 200 iterations (limit) – ~1400 seconds (!!!)

A Solution: Cross-Scenario Feasibility Cuts

- Idea: Enforce feasibility in the first stage (the commitments) to help ensure there is sufficient generation online to meet the “worst case” net load in each time period

$$\sum_{g \in G} (\bar{P}^g \cdot u_t^g) \geq \max_{s \in S}\{D_t^s + R_t^s - W_t^s\}$$

  - This brings more generation online in scenarios with low net-load
- A similar idea works to help ensure there is no over-generation in any scenario

$$\sum_{g \in G} (\underline{P}^g \cdot u_t^g) \leq \min_{s \in S}\{D_t^s\}$$

  - This prevents too many generators from coming online in scenarios with high net-load

- Critical observation
  - These cuts can be computed in parallel
  - And prior to execution of progressive hedging
  - Bonus: These cuts are simple and “obvious”

The Impact of Cross-Scenario Cuts (1)

Sandia National Laboratories

```
Results collection time=5.87 seconds
PH complete

Convergence history:
Converger=Normalized term diff
Iteration    Metric Value
     0           0.2288
     1           0.0862
     2           0.0615
     3           0.0371
     4           0.0316
     5           0.0359
     6           0.0272
     7           0.0291
     8           0.0213
     9           0.0299
    10           0.0261
    11           0.0287
    12           0.0249
    13           0.0183
    14           0.0232
    15           0.0143
    16           0.0153
    17           0.0152
    18           0.0109
    19           0.0094
    20           0.0060
    21           0.0048
    22           0.0048
    23           0.0051
    24           0.0052
    25           0.0049
    26           0.0018
    27           0.0024
    28           0.0035
    29           0.0000

Final number of discrete variables fixed=4055 (total=4080)
Final number of continuous variables fixed=0 (total=0)

Computing objective inner bound at xhat solution
Deactivate PH objective proximal terms time=0.00 seconds
Deactivate PH objective weight terms time=0.00 seconds
Fixed variable synchronization time=0.00 seconds
Time queueing subproblems=0.00 seconds
Result load time statistics - Min: 0.03 Avg: 0.03 Max: 0.03 StdDev: 0.0
Time waiting for subproblems=4.72 seconds
Sub-problem solve time statistics - Min: 0.23 Avg: 0.34 Max: 0.48 StdDe
Sub-problem pyomo solve time statistics - Min: 1.56 Avg: 1.89 Max: 2.69
Aggregate sub-problem solve time=4.72 seconds
Results collection time=5.43 seconds
Fixed variable synchronization time=0.00 seconds

Computed objective upper bound=  64470.4507
```

”Standard” progressive hedging (350 seconds)

```
External function invocation request transmission time=0.00 seconds
PH complete

Convergence history:
Converger=Normalized term diff
Iteration    Metric Value
     0           0.4633
     1           0.0512
     2           0.0289
     3           0.0205
     4           0.0085
     5           0.0062
     6           0.0002
     7           0.0000

Final number of discrete variables fixed=4029 (total=4080)
Final number of continuous variables fixed=0 (total=0)

Computing objective inner bound at xhat solution
Deactivate PH objective proximal terms time=0.00 seconds
Deactivate PH objective weight terms time=0.00 seconds
Fixed variable synchronization time=0.00 seconds
Time queueing subproblems=0.00 seconds
Result load time statistics - Min: 0.03 Avg: 0.03 Max: 0.03 StdDev: 0.00 (seco
Time waiting for subproblems=4.80 seconds
Sub-problem solve time statistics - Min: 0.24 Avg: 0.32 Max: 0.46 StdDev: 0.06
Sub-problem pyomo solve time statistics - Min: 1.59 Avg: 1.93 Max: 2.73 StdDev
Aggregate sub-problem solve time=4.81 seconds
Results collection time=5.71 seconds
Fixed variable synchronization time=0.00 seconds

Computed objective upper bound=  64404.9471
```

Progressive hedging with cross-scenario (80 seconds)

The Impact of Cross-Scenario Cuts (2)

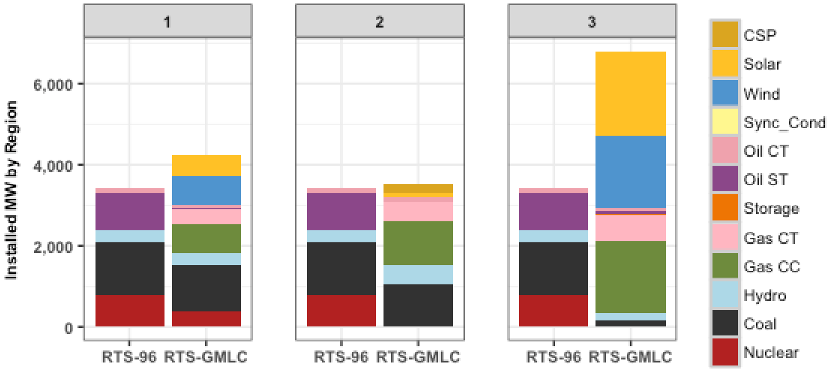
- Running on WECC-240++ scenarios for a simulated month (May) with high renewables penetration
  - 50 and 100 scenarios per instance
  - Generally observe wall clock reductions of 80%
  - In all but one case the run time did not exceed 210 seconds
  - In that one exception case, the reduction in wall clock time was only 20%

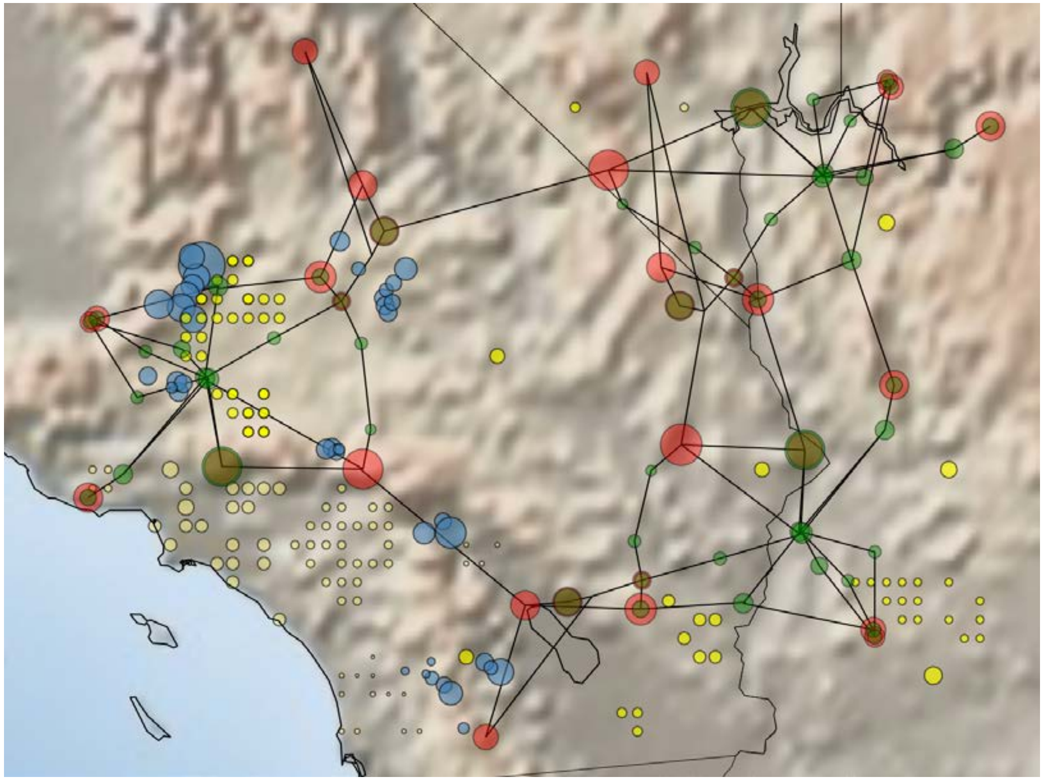
- Summary
  - We can address very difficult stochastic UC instances efficiently
  - But there are a few outliers that remain very difficult to solve
    - More research!

The RTS-GMLC Test Case

- A refresh of the IEEE RTS case
- github.com/GridMod/RTS-GMLC

A modern generation fleet…





Not intended to represent existing infrastructure

Toward Full-Scale Public UC Test Sets

- 365 deterministic and stochastic UC instances for RTS-GMLC
  - Developed using NREL wind and solar data sets
  - Publicly released once paper on deterministic RTS-GMLC is submitted
  - Contact me if you're interested for a preview case or three
- Larger instances are "available" – or at least have been constructed and analyzed
  - Derived from NREL's Eastern Renewable Generation Integration Study
- Not releasable due to CEII (Critical Infrastructure) issues

Conclusions

- Due to recent modeling, algorithmic, and implementation enhancements…
  - … we can actually solve at least modestly sized stochastic UC instances in operationally relevant time scales
  - … and with realistic probabilistic scenarios as input
- The availability of difficult, realistic stochastic UC instances is what drove this advance
  - We are moving to making such stochastic UC instances available to the general research community

Questions?

- Contact:
  - Ben Knueven, bknueve@sandia.gov

- Acknowledgements

  - Bonneville Power Administration for providing access to their data and for partial funding of this work
  - U.S. Department of Energy's Grid Modernization Laboratory Consortium, Project 1.4.26
  - U.S. Department of Energy's ARPA-E, Green Energy Network Integration (GENI) Project Portfolio