Optimization Driven Scenario Grouping for Stochastic Unit Commitment

FERC Technical Conference

Jun 27, 2017

Deepak Rajan*, Kevin Ryan, Shabbir Ahmed, Santanu Dey

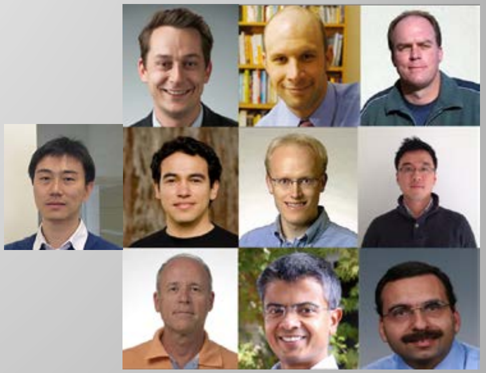
*rajan3@llnl.gov 925-424-6529

LLNL-PRES-733563
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Grid Modernization Laboratory Consortium (GMLC): Multi-Scale Production Cost Models

GRID MODERNIZATION LABORATORY CONSORTIUM U.S. Department of Energy

- An aggressive five-year grid modernization strategy for DOE
- Design and planning tools area: Multi-Scale Production Cost Models
  - Develop multi-scale production cost models with faster mathematical solvers
- PCM Goal
  - Substantially increase the ability of production cost models (PCM) to simulate power systems in more detail faster and more robustly
  - Both Deterministic and Stochastic

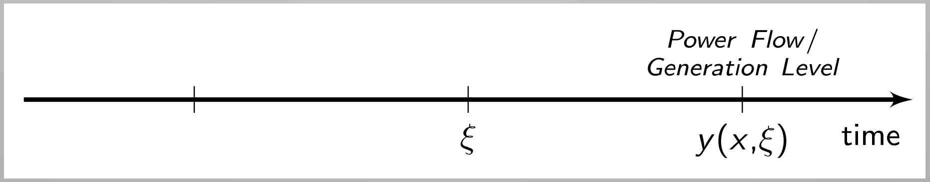


- Talks at Technical Conference
  - Session T1-B: Optimization Driven Scenario Grouping for Stochastic Unit Commitment (LLNL)
  - Session T2-B: Assessment of Wind Power Ramp Events in Scenario Generation for Stochastic Unit Commitment (SNL)
  - Session T3-A: Geographic Decomposition of Production Cost Models (NREL)
  - Session T3-A: Temporal Decomposition of Production Cost Modeling in Power Systems (ANL)

Two-stage stochastic programs and stochastic unit commitment

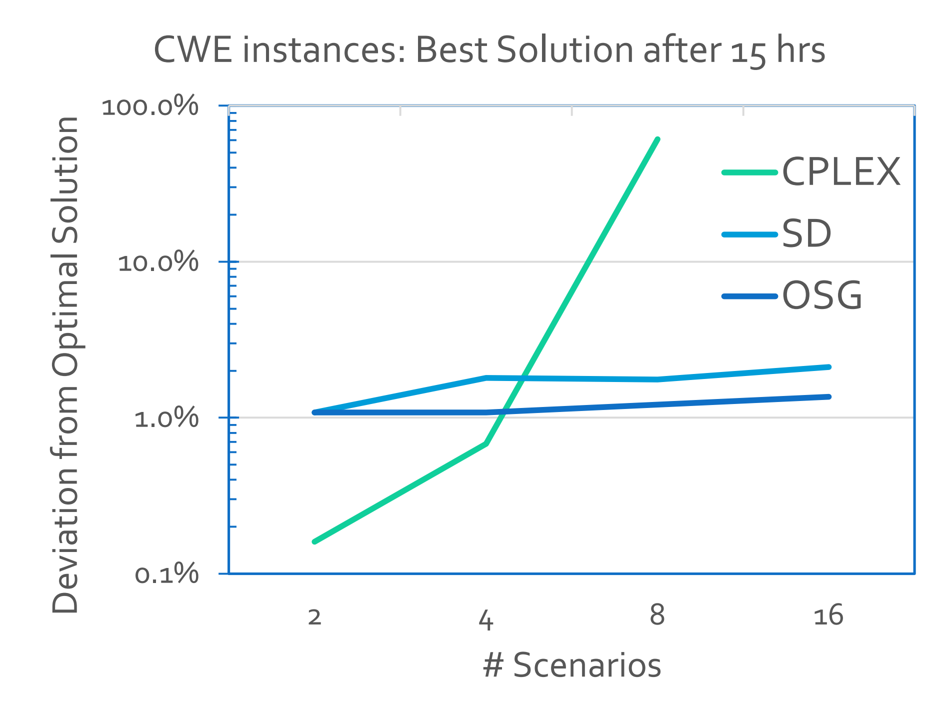
- Random realization: drawn from distribution
- GOAL: Make first stage decision to minimize Expected Cost

- Decide how to generate/transmit/dispatch power in the following day
- Uncertainty in solar and wind generation as well as customer demand



Scenario Grouping improves solution quality guarantees for Scenario Decomposition algorithms

- State-of-the art MIP solvers not suited for stochastic problems
  - CWE (Central-Western Europe) instances
    - — 679 nodes and 1037 lines, 637 thermal units
    - — Can not be solved to optimality in 15 hours
  - CPLEX obtains no solution for 16 scenarios



- Scenario Decomposition after one iteration provides high quality solution (~2%)
  - Parallelizing solution by decomposition
  - Capable of running on HPC

- Optimal Scenario Grouping (OSG) Techniques Improve Scenario Decomposition schemes by 40%
  - Provides higher-quality guarantee for solutions obtained (~1%)

Scenario Decomposition for Stochastic 0-1 programs

- We assume
- Scenarios are tied together using non-anticipativity constraints
- Relaxing non-anticipativity constraints, we decompose by scenario

Decomposition algorithms for Stochastic MIPs are not new…

- Dual Decomposition (Caroe and Schultz, 1999, Aravena and Papavasiliou, 2015, Kim and Zavala, 2016, ….)
- Benders Decomposition (Benders, 1962)
- L-Shaped (Van Slyke and Wets, 1969)
- Branch and Fix (Alonso-Ayuso, 2003)
- Disjunctive Decomposition (Ntaimo and Sen, 2005, 2008)
- Progressive Hedging (Watson and Woodruff, 2011)
- **Scenario Decomposition (Ahmed 2013)**

…and many others…

Scenario Decomposition algorithm

$B > LB$, GOTO 1

- Finitely convergent to optimality for binary first-stage
- Can easily include Lagrangian multiplier updates
- Easily parallelizes to a synchronous algorithm

Outline

- Improvements to Scenario Decomposition Algorithms
  - Asynchronous Implementation
    - Worker processes do not sit idle
    - Works well for instances where scenario solve/eval work is unbalanced
  - Incorporates performance improvements aimed at reducing upper bound evaluation time
  - Lower bound improvements (optimality cuts)
  - Solves some open **S**tochastic **I**nteger **P**rogramming **Lib**rary (SIPLIB) instances

- **New:** Optimal Scenario Grouping (OSG): Optimizes improvements in lower bounds, thus improving guarantee on solution quality
  - **Solves many more open SIPLIB instances**
  - **Provides much better optimality gaps at termination for Stochastic UC instances**

Scenario Decomposition improvements: Performance on SSCh instances

- Adding cuts (AS+Cut) solves all "easy" instances

Alonso-Ayuso (2003)
5 "hard" instances
4 "easy" instances

	181	(9,649)	**682**	**OPT**	**169**
c10	139,738	(1,732)	**126**	**OPT**	(1,259)

- Solves additional "hard" instance

		)	(5,901)	(9,832)	(18,427)
c4	201,454	(12,202)	(3,523)	(7,876)	(16,629)
c6	231,368	(10,514)	(4,828)	(10,273)	(8,825)
c8	100,523	(5,071)	**2,545**	(3,106)	(13,842)

Time limit: 3hrs
Nodes/Cores: 2/24

What about harder problems?
Lower Bounds from Scenario Decomposition scheme may be too weak…

- Increases time to convergence for "easy" problem instances
- Provides weak guarantee of solution quality for "hard" problem instances

Given multiplier at a particular iteration (λ),

how can we strengthen lower bound $z^*(\lambda)$?

Scenario Grouping to improve Lower Bounds: Motivation

- **Issue:** Scenario Decomposition Lower Bound may be too weak
- **Question:** Given multiplier λ, how can we strengthen lower bound $z^*(\lambda)$?
- **Idea:** 'Group' scenarios by re-enforcing some non-anticipativity constraints

- **Question:** What does it mean to 'group' scenarios?
  - Create 'multi-scenario' deterministic instances
- **Question:** Which scenarios do we group?
  - The groups that maximize bound improvement

Scenario Grouping to improve Lower Bounds: Motivation

- **Issue:** Scenario Decomposition Lower Bound may be too weak
- **Question:** Given multiplier λ, how can we strengthen lower bound $z^*(\lambda)$?
- **Idea:** 'Group' scenarios by re-enforcing some non-anticipativity constraints

Scenario 1 Scenario 2 Group 1 and 2

Scenario Grouping to improve Lower Bounds: Motivation

- **Issue:** Scenario Decomposition Lower Bound may be too weak
- **Question:** Given multiplier λ, how can we strengthen lower bound $z^*(\lambda)$?
- **Idea:** 'Group' scenarios by re-enforcing some non-anticipativity constraints

- **Question:** How much does the bound improve?
  - Maximizing bound improvement can be formulated as optimization problem

$$
\begin{aligned}
\max_{\theta,y} \quad & \sum_{m=1}^{M} \theta_m \\
s.t. \quad & \theta_m \leq \sum_{k=1}^{K} w_{ks} y_{km}, \quad \forall x \in \mathcal{S}, \forall m \in \mathcal{M} \\
& \sum_{m=1}^{M} y_{km} = 1, \quad \forall k \in \mathcal{K} \\
& \sum_{k=1}^{K} y_{km} \leq P, \quad \forall m \in \mathcal{M} \\
& y_{km} \in \{0,1\}, \ \theta_m \in \mathbb{R}, \quad \forall m \in M, \forall k \in \mathcal{K}
\end{aligned}
$$

Optimal Scenario Grouping (OSG): The best lower bound improvement

- OSG (Optimal Scenario Grouping)
  - When group size (P) = 2, can be solved as matching (polynomial!)

- Compare with SD/Asynch (Scenario Decomposition without Grouping)
- Compare with Rand (Random Grouping)

Is Scenario Grouping new?

- Bounding Schemes
  - (Sandikci et al. 2012). Expected Group Subproblem Objective Bounds (EGSO)
  - (Sandikci et al. 2014) (Maggioni et al. 2015), (Zenarosa et al. 2014). Extensions of EGSO
  - (Boland et al. 2016). Expected Partition Scenario Bounds
  - (Gade et al. 2016). Bounding in Progressive Hedging
- Grouping/Aggregation Schemes
  - (Crainic et al. 2014) K-means clustering.
  - (Song and Leudtke 2015) Solution driven scenario aggregation.

OSG (Optimal Scenario Grouping):

Grouping to maximize lower bound improvement

OSG performance for SSCh instances (“Easy”)

9	(1,732)	**20**	**13**	**41**

Time limit: 3hrs
Nodes/Cores: 1/12

- Optimal Scenario Grouping (Part) solves **all** instances when P=4
- Even random grouping helps

But there is no free lunch: Scenario Grouping increases the time per iteration for Scenario Decomposition schemes

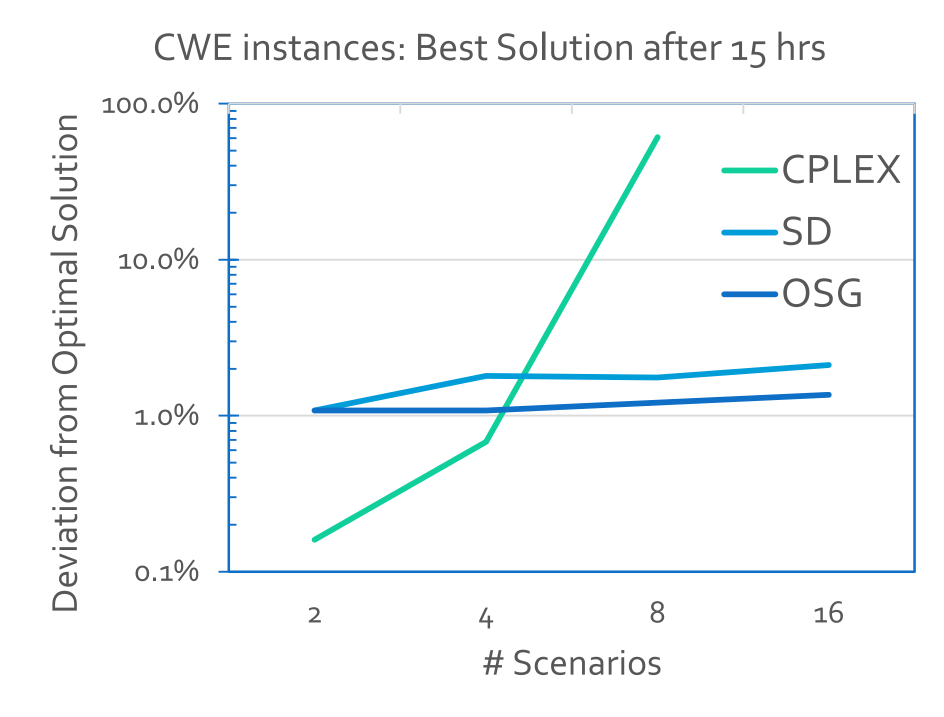
- The significant reduction in number of iterations offsets the increased time per iteration
  - As proved by SSCh experiments
- What about problems where 2-scenario grouped problems are too expensive?
  - Stochastic UC on realistic instances, perhaps?
- Use OSG as a post-processing scheme (one last iteration) to calculate better lower bounds
- What if that is too expensive?
- Use LP relaxations of 2-scenario grouped problems

Stochastic UC experiments:

1 SD iteration + OSG + LP relaxation of 2-scenario problems

OSG Performance: What about "hard" Stochastic Unit commitment instances?

- State-of-the art MIP solvers not suited for stochastic problems
  - CWE (Central-Western Europe) instances
  - CPLEX obtains no solution for 16 scenarios



- Scenario Decomposition after one iteration provides high quality solution (~2%)

- Can we provide better guarantee for solutions obtained?

- Optimal Scenario Grouping (OSG) Techniques Improve Scenario Decomposition schemes by 40%

Why not just group randomly? Do we need optimal grouping? Random works if the number of scenarios is small (4)

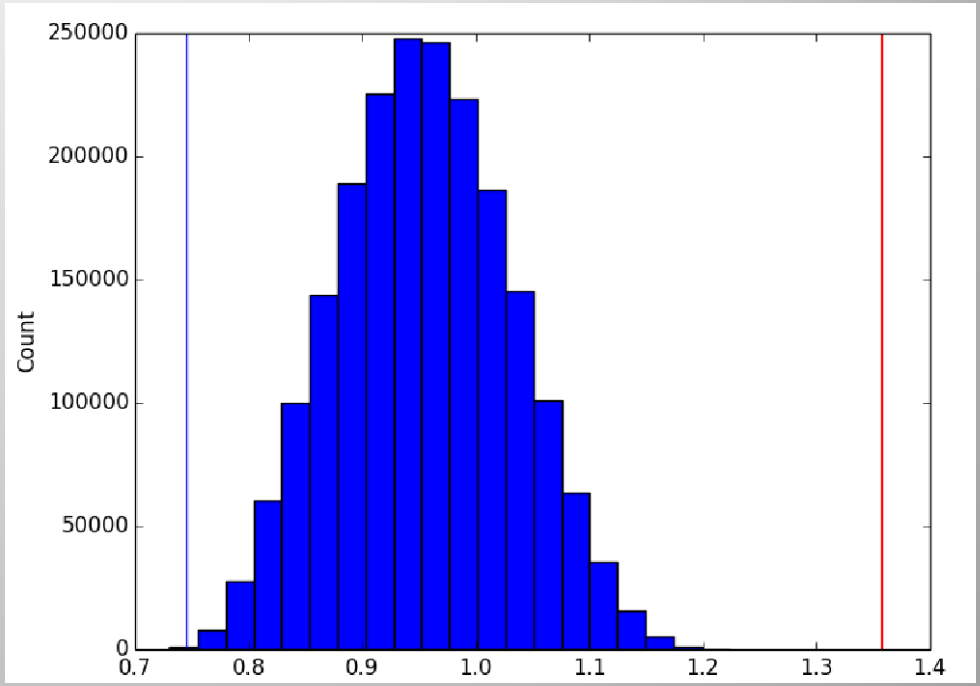
Problem		Best UB	Asynch	Rand P=2
Autumn	WD	36,177,240.0	1.62%	0.98%
	WE	27,180,129.1	2.12%	1.43%
Spring	WD	23,323,714.7	0.66%	0.22%
	WE	18,168,310.0	0.67%	0.16%
Summer	WD	23,950,144.7	0.70%	0.43%
	WE	17,640,484.7	1.10%	0.64%
Winter	WD	29,287,830.0	3.72%	2.34%
	WE	23,546,261.2	3.82%	2.41%
Average			1.80%	1.08%

CPLEX 12.5
Time limit: 15hrs
Nodes = #Scenarios/2 +1
Cores = 12 * #nodes

- Random grouping reduces gap by 40% for 4 scenario problems
- Random grouping does not scale

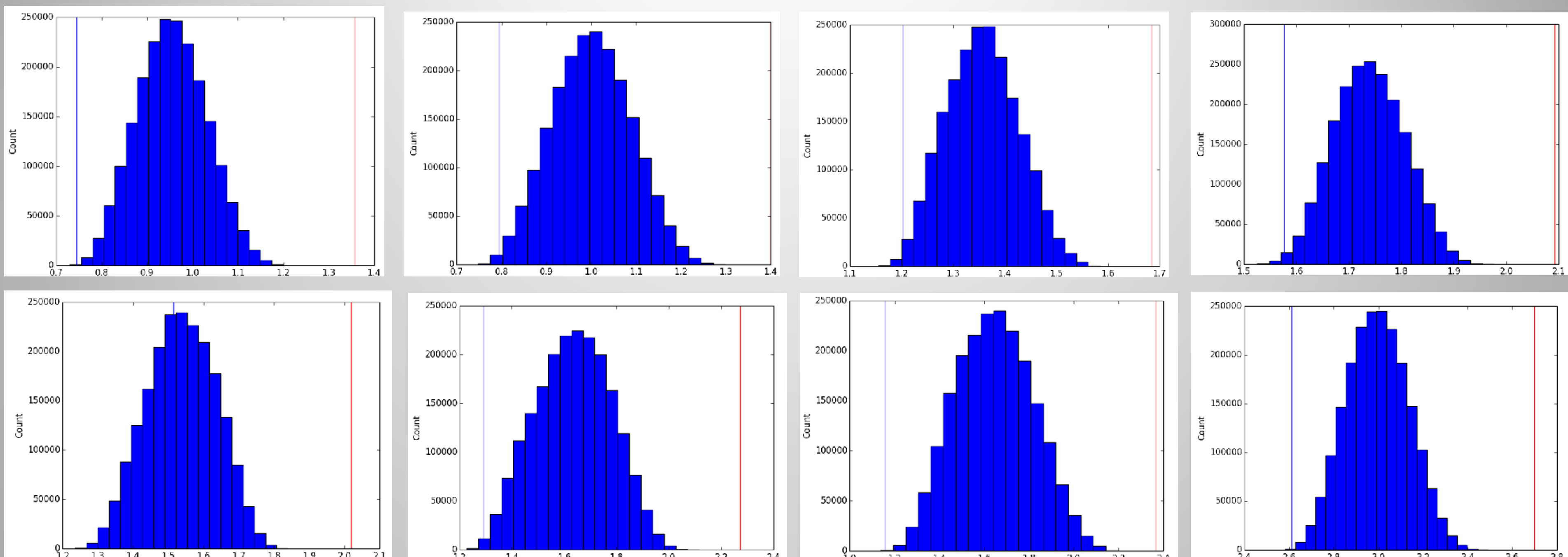
Why not just group randomly? Do we need optimal grouping? Random grouping does not work as well for 16 scenarios

- Not surprising: Too many groupings to choose from
- Spring Weekday (histogram of Deviation from optimal)
  - Asynch/SD bound (red line)
  - OSG bound (blue line)
  - Rand bound (from histogram)



- Random partitioning reduces gap by 25%
- Optimal scenario grouping reduces gap by more than 40%

What about other 16 scenario instances? Works very well on 7/8



- Optimal scenario grouping (OSG) comparable to random in 1/8
- Current research: Improving the quality of OSG (improve estimates)

Summary

- Scenario decomposition natural algorithm for solving stochastic integer programs: Improvements can significantly improve performance
- Optimal scenario grouping (OSG) solved previously unsolved instances, demonstrated effectiveness on standard test instances (SIPLIB)
- Optimal scenario grouping improves lower bound for stochastic unit commitment instances

Acknowledgments/Auspices

- Prepared by LLNL under Contract DE-AC52-07NA27344
- LLNL LDRD Funding (15-ERD-041), LLNL GMLC Funding (1.4.26 – PCM)


Thank you!