Toward Scalable Solvers for Stochastic Multi-Stage Long-Term Generation and Transmission Capacity Expansion

Dr. Jean-Paul Watson
Discrete Math and Complex Systems Department
Sandia National Laboratories, Albuquerque, New Mexico

Key Algorithmic Collaborators:
Richard Chen (Sandia)
William Hart (Sandia)
Roger Wets (UC Davis)
David Woodruff (UC Davis)

Technical Conference on Planning Models and Software
Federal Energy Regulatory Commission
Washington, DC
June 9 - 10, 2010
Combinatorial Optimization R&D at Sandia

- Efforts are centered on two primary research thrusts
  - Risk Management
    - Multi-stage, general mixed-integer
    - Efficient risk versus cost tradeoff analysis
    - Scalable Conditional Value-at-Risk (CVaR) computation
  - Multi-Stage Stochastic Optimization
    - Multi-stage, general mixed-integer
    - Massively parallel environments
- Application drivers
  - Contamination sensor network design (INFORMS Edelman Finalist)
  - Network interdiction for critical infrastructure
  - Biofuel network design
  - Electrical grid generation and transmission capacity expansion
  - Scalable unit commitment with large renewables penetration
- Funding sources
  - DOE Office of Science, US EPA, Sandia LDRD
Resource Allocation: Integer and Stochastic Programming

- Deterministic Mixed-Integer Programming (MIP)
  - The PDE of Operations Research
    \[
    \begin{align*}
    \min & \quad c'x + h'y \\
    \text{s.t.} & \quad Ax + By \leq b \\
               & \quad x \in \mathbb{Z}_+^n (x \geq 0, \text{ } x \text{ integer}) \\
               & \quad y \in \mathbb{R}_+^n (y \geq 0)
    \end{align*}
    \]
  - Approximable for most real-world problems (NP-Hard)
- Stochastic Mixed-Integer Programming (SMIP)
  - SMIP = MIP + \textit{uncertainty} + \textit{recourse}
    \[
    \begin{align*}
    \min & \quad f(x) = c^T x + \mathbb{E}[Q(x, \omega)] \\
    \text{s.t.} & \quad Ax \geq b, \quad x \in \mathbb{R}_+^{n_1 - p_1} \times \mathbb{Z}_+^{p_1} \\
    Q(x, \omega) = & \min \quad q(\omega)^T y \\
    \text{s.t.} & \quad Wy \geq h(\omega) - T(\omega)x \\
               & \quad y \in \mathbb{R}_+^{n_2 - p_2} \times \mathbb{Z}_+^{p_2}
    \end{align*}
    \]
  - Still NP-Hard, but far more difficult than MIP in practice
Capacity Expansion as Stochastic Mixed-Integer Programming

- Many historical planning models are either deterministic or linear (or both)
  - Driven by combinations of data availability and solver maturity

- With advances in IT and solver technology, multi-stage stochastic mixed-integer formulations are becoming more prevalent in the literature
  - Singh et al. (2009), Wang and Ryan (2010), Huang and Ahmed (2009)
  - General paradigm captures key aspects of capacity expansion problems

- Key technological challenges to deploying multi-stage stochastic MIP models
  - No canonical generation and transmission capacity expansion model
  - Multi-stage stochastic MIP solvers are not yet general-purpose
  - The difficulty of multi-stage stochastic MIPs likely requires parallelism

- Key requirement to solve the deployment barrier
  - Modeling and solver framework to facilitate rapid prototyping of alternative solution strategies, supporting built-in parallelism
Stochastic Mixed-Integer Programming: The Algorithm Landscape

• The Extensive Form or Deterministic Equivalent
  – Write down the full variable and constraint set for all scenarios
  – Write down, either implicitly or explicitly, non-anticipativity constraints
  – Attempt to solve with a commercial MIP solver
  - Great if it works, but often doesn’t due to memory or time limits
• Time-stage or “vertical” decomposition
  – Benders / L-shaped methods (including nested extensions)
  – Pros: Well-known, exact, easy for (some) 2-stage, parallelizable
  – Cons: Master problem bloating, multi-stage difficulties
• Scenario-based or “horizontal” decomposition
  – Progressive hedging / Dual decomposition
  – Pros: Inherently multi-stage, parallelizable, leverages specialized MIP solvers
  – Cons: Heuristic (depending on algorithm), parameter tuning
• Important: Development of general multi-stage SMIP solvers is an open research area
Progressive Hedging: A Review and/or Introduction

1. $k := 0$

2. For all $s \in S$, $x_s^{(k)} := \arg\min_x (c \cdot x + f_s \cdot y_s) : (x, y_s) \in Q_s$

3. $\bar{x}^k := (\sum_{s \in S} p_s d_s x_s^{(k)}) / \sum_{s \in S} p_s d_s$

4. For all $s \in S$, $w_s^{(k)} := \rho (x_s^{(k)} - \bar{x}^{(k)})$

5. $k := k + 1$

6. For all $s \in S$, $x_s^{(k)} := \arg\min_x (c \cdot x + w_s^{(k-1)} x + \rho / 2 \|x - \bar{x}^{(k-1)}\|^2 + f_s \cdot y_s) : (x, y_s) \in Q_s$

7. $\bar{x}^{(k)} := (\sum_{s \in S} p_s d_s x_s^{(k)}) / \sum_{s \in S} p_s d_s$

8. For all $s \in S$, $w_s^{(k)} := w_s^{(k-1)} + \rho (x_s^{(k)} - \bar{x}^{(k)})$

9. $g^{(k)} := \frac{(1-\alpha)|S|}{\sum_{s \in S} p_s d_s} \sum_{s \in S} \|x_s^{(k)} - \bar{x}^{(k)}\|

10. If $g^{(k)} < \epsilon$, then go to step 5. Otherwise, terminate.

Rockafellar and Wets (1991)
Progressive Hedging as a Stochastic Mixed-Integer Heuristic

• Progressive Hedging does provably converge in the \textit{convex} case, in linear time
  – NOTE: As practitioners know well, linear time can take a \textit{long} time

• Progressive Hedging (PH) has been successfully used as a heuristic for multi-stage mixed-integer stochastic programming
  – Løkketangen and Woodruff (1996)
  – Numerous others (Birge, Gendreau, Crainic, Rei)

• Practical and critical issues of note
  – How to pick \( \rho \)?
  – Cycle detection
  – Convergence acceleration
    • Variable fixing
    • Slamming

Progressive Innovations for a Class of Stochastic Mixed-Integer Resource Allocation Problems
Example of PH Impact:
• Extensive form solve time: >20K seconds
• PH solve time: 2K seconds
The Impact of Decomposition: Wind Farm Network Design

• Where to site new wind farms and transmission lines in a geographically distributed region to satisfy projected demands at minimal cost?
• Formulated as a two-stage stochastic mixed-integer program
  – First stage decisions: Siting, generator/line counts
  – Second stage “decisions”: Flow balance, line loss, generator levels
• 8760 scenarios representing coincident hourly wind speed, demand
• Solve with Benders: Standard and Accelerated

• Summary: A non-trivial Benders variant is required for tractable solution

*Slide courtesy of Dr. Richard Chen (Sandia California)*
Mean versus Risk? Some Terminology

Conditional Value-at-Risk (CVaR) is a linear approximation of TCE.

Frequency

Mean

Value at Risk (VaR)

Tail-Conditional Expectation (TCE)

Worst-Case

(1 - α) Percentile

Cost
Progressive Hedging and Conditional Value-at-Risk

- Scenario-based decomposition of Conditional Value-at-Risk models is conceptually straightforward (Schultz and Tiedemann 2006)

**Proposition 5.1.** Assume that \( \mu \) is discrete with finitely many scenarios \( h_1, \ldots, h_J \) and corresponding probabilities \( \pi_1, \ldots, \pi_J \). Let \( \alpha \in (0, 1) \). Then the stochastic program

\[
\min \{ Q_{CVaR_\alpha}(x) : x \in X \}
\]

(11)

can be equivalently restated as

\[
\min_{x, y, y', v, \eta} \left\{ \eta + \frac{1}{1 - \alpha} \sum_{j=1}^{J} \pi_j v_j : W y_j + W' y'_j = h_j - T x, \right. \\
\left. v_j \geq c^T x + q^T y_j + q'^T y'_j - \eta, \right. \\
x \in X, \ \eta \in \mathbb{R}, \ y_j \in \mathbb{Z}_m, \\
y'_j \in \mathbb{R}_{m'}, \ v_j \in \mathbb{R}_+, \ j = 1, \ldots, J \right\}.
\]

- But
  - Computational issues are largely unexplored

Slide 11
Selecting Scenarios to Ignore in Stochastic Optimization: Advances in Probabilistic Integer Programming Solvers

Central Theme: The Need to Ignore a Small Fraction $\alpha$ of Scenarios During Optimization

\[
\begin{align*}
\text{minimize} & \quad c \cdot x + \sum_{s \in S} p_s (f_s \cdot y_s) \\
\text{subject to:} & \quad (x, y_s) \in \mathcal{Q}_s, \quad \forall s \in \{S : d_s = 1\} \\
& \quad \sum_{s \in S} p_s d_s \geq (1 - \alpha) \\
& \quad d_s \in \{0, 1\}, \quad \forall s \in S
\end{align*}
\]

Results for network design:
- 2-8% better solutions than CPLEX, 1440m versus ~10m

Impact: - Best available heuristic for solving probabilistic integer programs
- First demonstration on large-scale, real-world problems
An Open-Source Optimization Modeling Tool

Modeling Capabilities
- Abstract model definition
- LP and MILP models
- Manage multiple model instances
- Stochastic modeling extensions

Key Features
- Parallel solver execution
- Extensible framework
- Interface to many data sources
- Portability
- Embedded in modern programming language
- Freely available
- Unrestricted open source license

TO LEARN MORE VISIT >>

https://software.sandia.gov/pyomo
Hedging Against Uncertainty: A Modeling Language and Solver Library

You Plan

Stuff Happens

You Adjust

More Stuff Happens

Multi-Stage Planning for Uncertain Environments
- Explicitly capture recourse
- Uncertainty modeling framework
- Integrated solver strategies

What We Do:
- Mixed decision variables
  - Continuous
  - Integer/Binary
- General multi-stage
- Stochastic programming
  - Expected value
  - Conditional Value-at-Risk
  - Scenario selection
  - Cost confidence intervals

How We Do It:
- Deterministic equivalent
- Scenario-based decomposition
  - Progressive Hedging
  - Customizable accelerators
- Algebraic modeling via Pyomo
- SMP and cluster parallelism
- Integrated high-level language support
- Multi-platform, unrestricted license
- Open source, actively supported by Sandia
  - Co-Managed by Sandia and COIN-OR

PySP: Stochastic Programming in Python

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin company, for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000.
Stochastic Programming and High-Performance Computing

- Decomposition algorithms for solving multi-stage stochastic mixed-integer programs are “naturally” parallelizable
  - L-shaped and Progressive Hedging are particularly amenable
- Practical issues arise as the number of scenarios grows
  - Even the most modest branching processes in multi-stage decision environments lead to thousands to millions of scenarios
  - MIP solve times are heterogeneous, leading to poor parallel efficiency
- Current capabilities in PySP:
  - Scalability to order-thousand scenarios and processors
- In-progress efforts
  - Asynchronous decomposition algorithms
  - IBM Research Blue Gene deployment
  - EC2 / Gurobi deployment
- Major deployment issue: MIP solver licensing to thousands of processors
  - Mitigated in part by Gurobi EC2 deployment
Scenario Sampling: How Many is Enough?

- Discretization of the scenario tree is “standard” in stochastic programming
  - Often, no mention of solution or objective stability
  - Let alone rigorous statistical hypothesis-testing of stability
  - Don’t trust anyone who doesn’t show you a confidence interval

- Two general approaches in the literature
  - Has the solution converged? (Sample Average Approximation)
  - Has the objective converged? (Multiple Replication Procedure)

- Formal question we are concerned with
  - What is the probability that \( \hat{x} \)’s objective function value is suboptimal by more than \( \alpha \)%?

- Initial implementation available in PySP
  - Preliminary results for various network expansion and design problems indicates that we are using far too few samples
Conclusions

• Multi-stage stochastic mixed-integer programs are a natural modeling paradigm for solving generation/transmission capacity expansion problems

• Solver technologies capable of solving realistic instances are emerging
  – But many challenges remain, both in terms of research and deployment

• Sandia is developing software to address what we view as the challenges
  – Frameworks to support rapid modeling and solver prototyping
  – Scalable parallelization of decomposition strategies
  – Rigorous quantification of uncertainty bounds on solution costs
  – Open-source solutions
    • Sandia is mandated to collaborate with and aid industry – not compete

• For more information:
  – https://software.sandia.gov/trac/coopr/wiki/PySP -or- jwatson@sandia.gov