Algorithmic Innovations and Software for Dual Decomposition of Stochastic Mixed Integer Programming

Kibaek Kim Victor M. Zavala
Mathematics and Computer Science Division
Argonne National Laboratory

Federal Energy Regulatory Commission
June 24, 2015

Stochastic Mixed-Integer Programming (SMIP)

General formulation of SMIP:

$$\begin{aligned} \min \quad & c^T x + \mathbb{E}[Q(x,\omega)] \\ \text{s.t.} \quad & Ax \geq b, \\ & x \in \mathbb{R}^{n_1 - p_1} \times \mathbb{Z}^{p_1} \end{aligned}$$

Make *here-and-now* decision $\mathbf{x}$

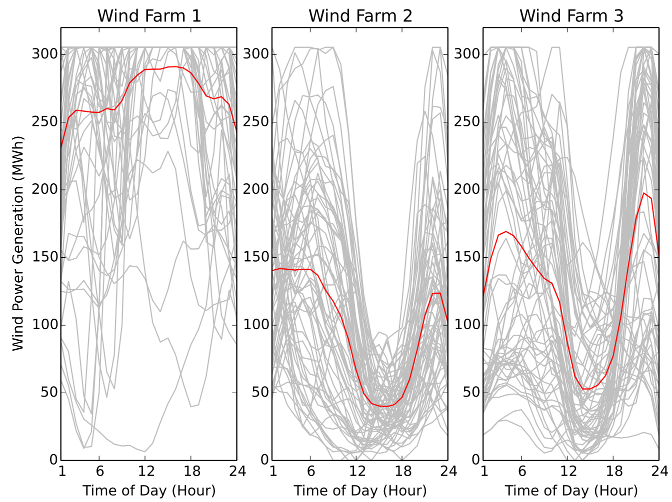
- Operational decisions
- Logical decisions
- Countable items

Recourse function of an integer program:

$$\begin{aligned} \mathbb{E}[Q(x,\omega)] = \min \quad & q(\omega)^T y \\ \text{s.t.} \quad & W(\omega) y \geq h(\omega) - T(\omega) x, \\ & y \in \mathbb{R}^{n_2 - p_2} \times \mathbb{Z}^{p_2} \end{aligned}$$

Make *wait-and-see* decision $\mathbf{y}$ for given first-stage decision $\mathbf{x}$ and event ω

- Recourse action to event realization (e.g. re-scheduling, system restoration)
- Time-dependent decisions



Observe *stochastic event* ω

- System Failure
- Demand and supply
- Cost and price
- Weather

DOE-Relevant Applications of Stochastic MIP

Stochastic Unit Commitment:

- Uncertain wind power generation
- Fast generators (e.g., gas-fired) introduce integer variables in the second stage.
- Coupling day-ahead decisions (multi-stage)

Stochastic Infrastructure Design:

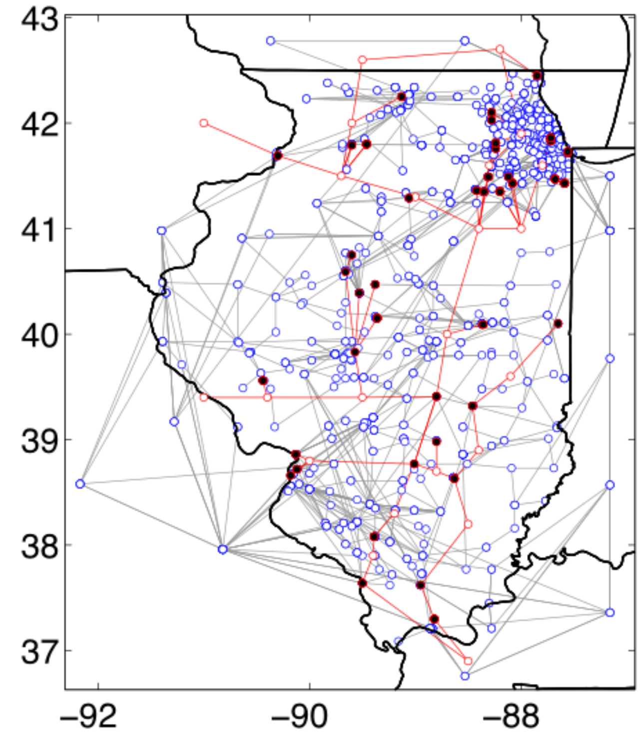
- Uncertain short-term dynamics in long-term planning
- Multi-scale problem in time and space
- Piecewise linearization of nonlinearity

Infrastructure Interdependency (Gas/Electric):

- Uncertain impact between infrastructure systems
- Multi-scale problem in time and space

Resource Planning for Infrastructure Restoration:

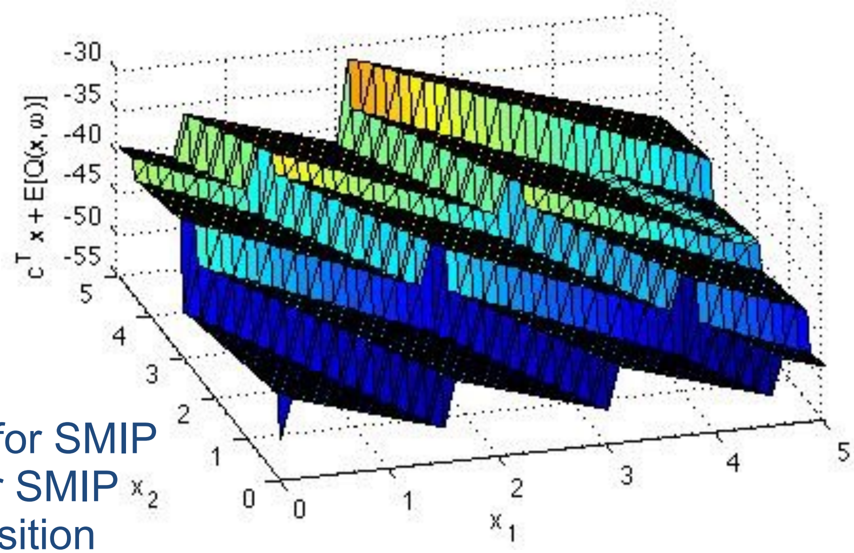
- Uncertain outage location and restoration time
- Scheduling of crews and resources



Technical Challenges of SMIP

1. The **size** of the optimization problem increases in the number of scenarios.

2. **Non-convexity** and **discontinuity**:
   - Difficult to find a global optimal solution
   - Does not fit in traditional Benders decomposition framework

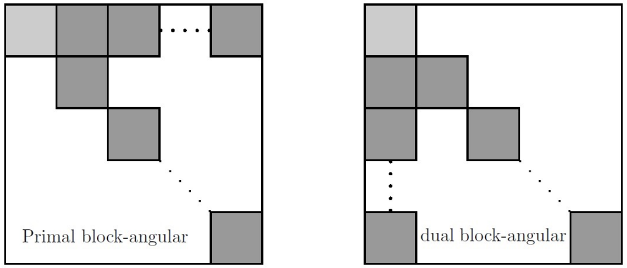


[Ahmed and Tawarmalani, 2004]

3. Few solution tools available
   - *PIPS [1]* (Argonne): continuous
   - *PySP [2]* (Sandia): progressive hedging for SMIP
   - *ddsip [3]*: dual decomposition method for SMIP
   - *FortSP [4]* (OptiRisk): Benders decomposition only

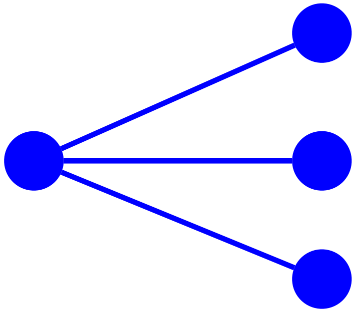
1. Petra, C.G., Anitescu, M.: A preconditioning technique for schur complement systems arising in stochastic optimization. Computational Optimization and Applications 52(2), 315–344 (2012)
2. Watson, J.P., Woodruff, D.L., Hart, W.E.: PySP: modeling and solving stochastic pro- grams in python. Mathematical Programming Computation 4(2), 109–149 (2012)
3. M̈arkert,A., Gollmer, R.: Users Guide to ddsip–A C Package for the Dual Decomposition of Two-Stage Stochastic Programs with Mixed-Integer Recourse (2014)
4. OptiRisk Systems: FortSP: A Stochastic Programming Solver, Version 1.2 (2014). http://www.optirisk-systems.com/manuals/FortspManual.pdf

DSP: Decomposition methods for Structured Programming



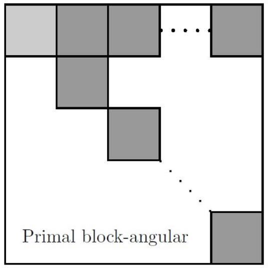
- DSP: An ***parallel*** open-source software package
  - ***Decomposition*** methods for ***structured programming***
  - Solving stochastic mixed-integer programs
  - Using parallel computing

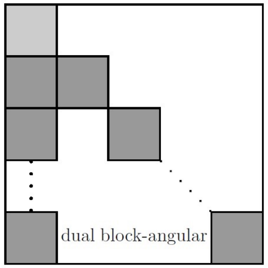
- Decompositions for Stochastic Mixed-Integer Program



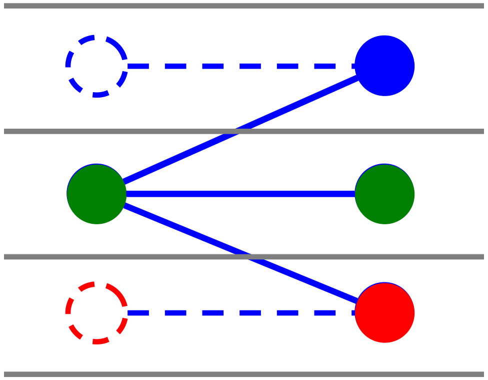
DSP: D̲ecomposition methods for S̲tructured P̲rogramming

- DSP: An ***parallel*** open-source software package
  - ***Decomposition*** methods for ***structured programming***
  - Solving stochastic mixed-integer programs
  - Using parallel computing





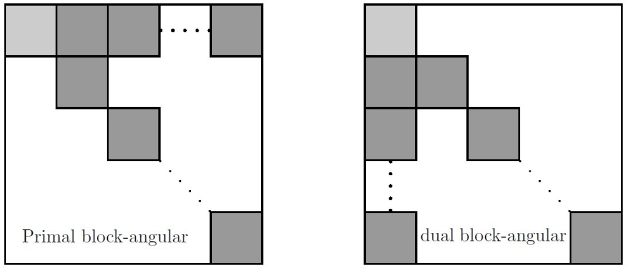
- Decompositions for Stochastic Mixed-Integer Program



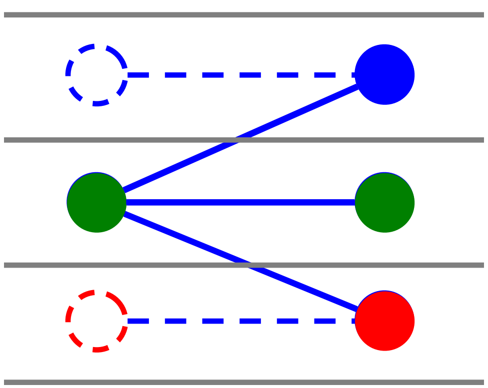
Scenario Decomposition

DSP: Decomposition methods for Structured Programming

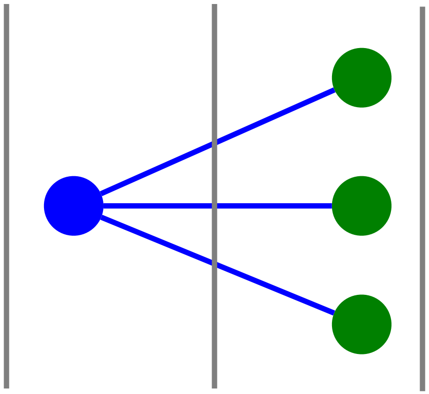
- DSP: An ***parallel*** open-source software package
  - ***Decomposition*** methods for ***structured programming***
  - Solving stochastic mixed-integer programs
  - Using parallel computing



- Decompositions for Stochastic Mixed-Integer Program



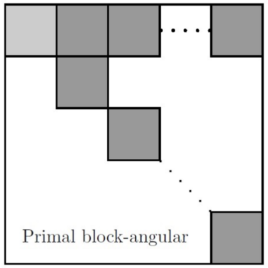
Scenario Decomposition

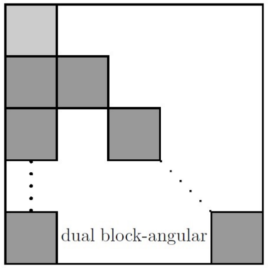


Stage Decomposition

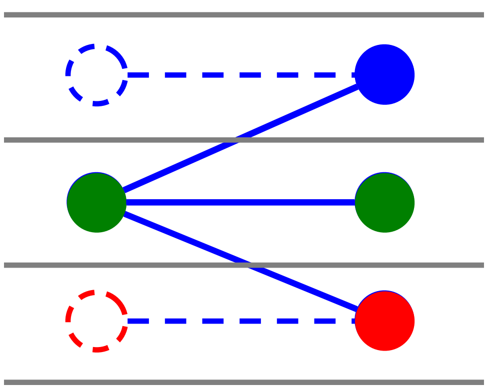
DSP: Decomposition methods for Structured Programming

- DSP: An ***parallel*** open-source software package
  - ***Decomposition*** methods for ***structured programming***
  - Solving stochastic mixed-integer programs
  - Using parallel computing

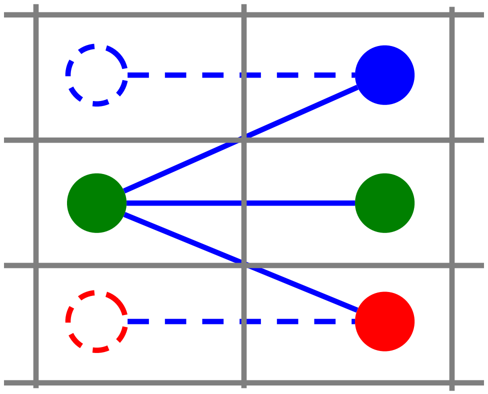




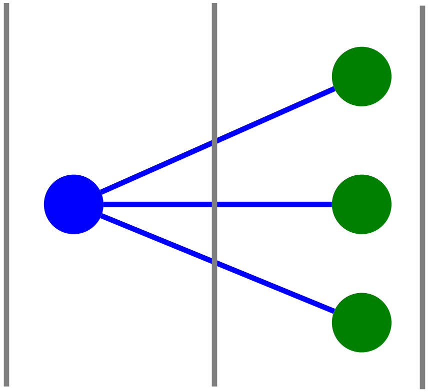
- Decompositions for Stochastic Mixed-Integer Program



Scenario Decomposition

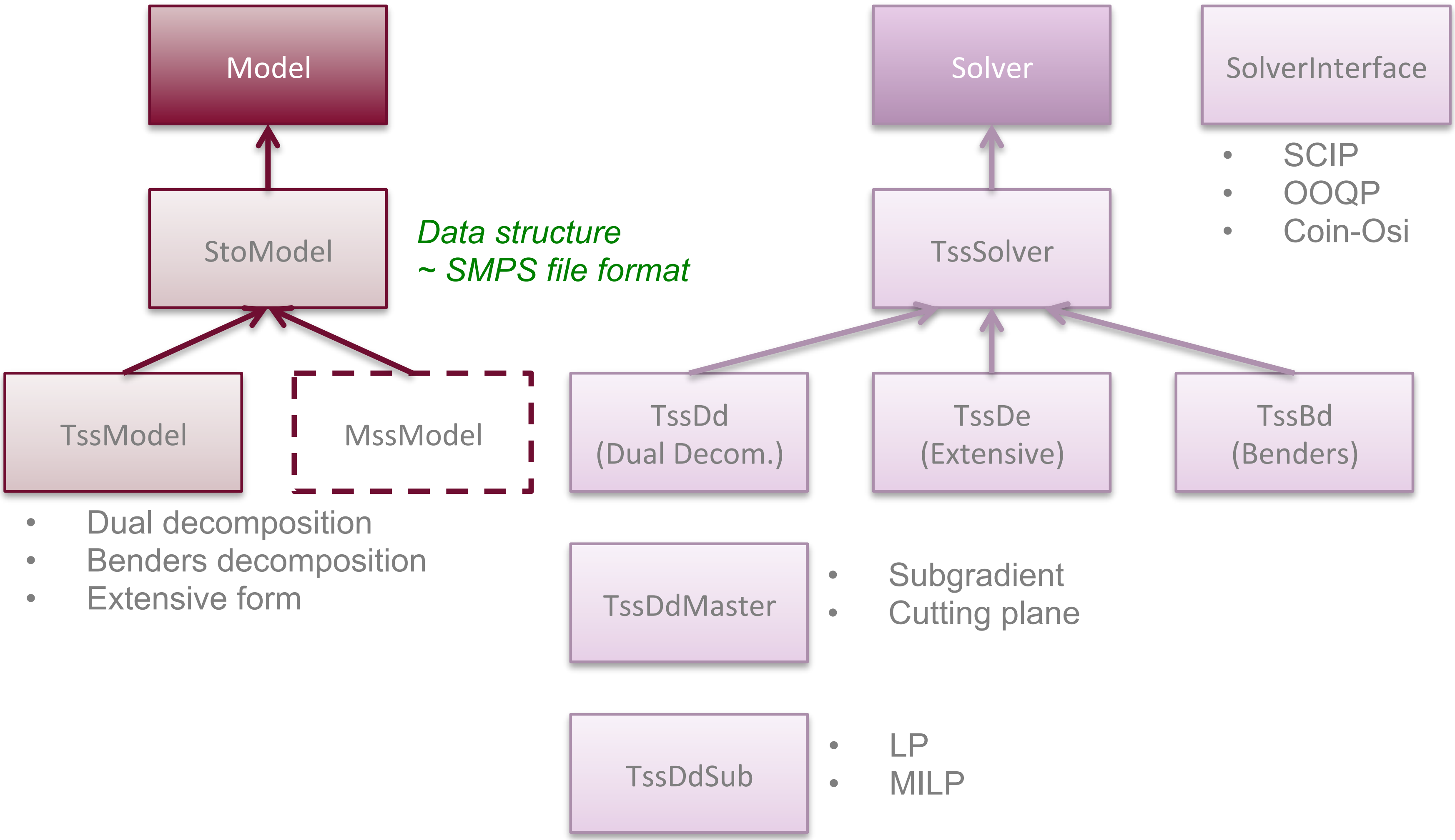


Scenario+Stage Decomposition



Stage Decomposition

Software Design of DSP



Interfaces for C, SMPS and *StochJuMP*

- Dynamic programming language (script language) developed by MIT
- Computational speed like "C"
- Familiar syntax like MATLAB, python and R
- ***StochJuMP*** package for modeling stochastic programs [2]
  - *Faster than any* other modeling tools, and as fast as AMPL
  - Supports distributed memory computing via **MPI** communication – *Scalable!*
  - Users: ***PIPS*** and ***DSP***

	Fortran gcc 4.8.2	Julia 0.3.7	Python 2.7.9	R 3.1.3	Matlab R2014a	Octave 3.8.1	Mathematica 10.0	JavaScript V8 3.14.5.9	Go go1.2.1	LuaJIT gsl-shell 2.3.1	Java 1.7.0_75
fib	0.57	2.14	95.45	528.85	4258.12	9211.59	166.64	3.68	2.20	2.02	0.96
parse_int	4.67	1.57	20.48	54.30	1525.88	7568.38	17.70	2.29	3.78	6.09	5.43
quicksort	1.10	1.21	46.70	248.28	55.87	1532.54	48.47	2.91	1.09	2.00	1.65
mandel	0.87	0.87	18.83	58.97	60.09	393.91	6.12	1.86	1.17	0.71	0.68
pi_sum	0.83	1.00	21.07	14.45	1.28	260.28	1.27	2.15	1.23	1.00	1.00
rand_mat_stat	0.99	1.74	22.29	16.88	9.82	30.44	6.20	2.81	8.23	3.71	4.01
rand_mat_mul	4.05	1.09	1.08	1.63	1.12	1.06	1.13	14.58	8.45	1.23	2.35

Benchmark times relative to C (smaller is better, C performance = 1.0) [1]

1. **Julia**, http://julialang.org
2. **Parallel algebraic modeling for stochastic optimization**. Joey Huchette, Miles Lubin, Cosmin Petra (2014), *HPTCDL'14 Proceedings of the 1st Workshop on High Performance Technical Computing in Dynamic Languages*, 29–35, doi:10.1109/HPTCDL.2014.6

Example: *DSP-StochJuMP*

1. Carøe, Claus C., and Rüdiger Schultz. "Dual decomposition in stochastic integer programming." *Operations Research Letters* 24.1 (1999): 37-45.

$$\min \left\{ -1.5\,x_1 - 4\,x_2 + \sum_{s=1}^{3} p_s Q(x_1, x_2, \xi_1^s, \xi_2^s) : x_1, x_2 \in \{0, \ldots, 5\} \right\},$$

where

$$\begin{aligned} Q(x_1, x_2, \xi_1^s, \xi_2^s) = \min_{y_1, y_2, y_3, y_4} \quad & -16y_1 + 19y_2 + 23y_3 + 28y_4 \\ \text{s.t.} \quad & 2y_1 + 3y_2 + 4y_3 + 5y_4 \le \xi_1^s - x_1 \\ & 6y_1 + y_2 + 3y_3 + 2y_4 \le \xi_2^s - x_2 \\ & y_1, y_2, y_3, y_4 \in \{0, 1\} \end{aligned}$$

and $(\xi_1^s, \xi_2^s) \in \{(7,7), (11,11), (13,13)\}$ with probability $1/3$.

Example: *DSP-StochJuMP*

$$\min \left\{ -1.5\,x_1 - 4\,x_2 + \sum_{s=1}^{3} p_s Q(x_1, x_2, \xi_1^s, \xi_2^s) : x_1, x_2 \in \{0, \dots, 5\} \right\},$$

where

$$\begin{aligned} Q(x_1, x_2, \xi_1^s, \xi_2^s) = \min_{y_1, y_2, y_3, y_4} \quad & -16y_1 + 19y_2 + 23y_3 + 28y_4 \\ \text{s.t.} \quad & 2y_1 + 3y_2 + 4y_3 + 5y_4 \leq \xi_1^s - x_1 \\ & 6y_1 + y_2 + 3y_3 + 2y_4 \leq \xi_2^s - x_2 \\ & y_1, y_2, y_3, y_4 \in \{0, 1\} \end{aligned}$$

and $(\xi_1^s, \xi_2^s) \in \{(7,7), (11,11), (13,13)\}$ with probability 1/3.

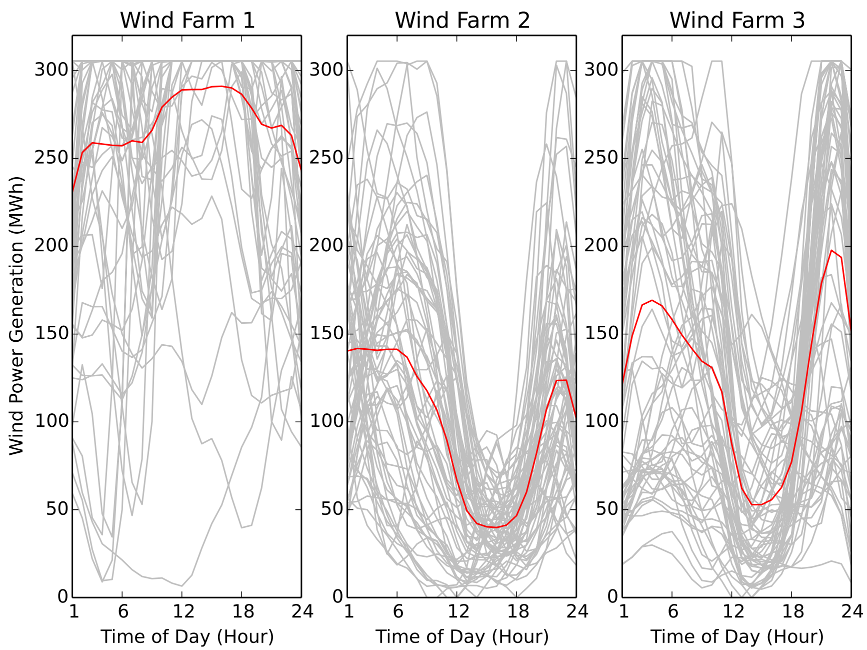
Only 16 lines of Julia script!

```
using DSPsolver, StochJuMP, MPI; # Load packages
MPI.Init(); # Initialize MPI
m = StochasticModel(3); # Create a Model object with three scenarios
xi = [[7,7] [11,11] [13,13]]; # random parameter
@defVar(m, 0 <= x[i=1:2] <= 5, Int);
@setObjective(m, Min, -1.5*x[1]-4*x[2]);
@second_stage m s begin
    q = StochasticBlock(m, 1/3);
    @defVal(q, y[j=1:4], Bin);
    @setObjective(q, Min, -16*y[1]+19*y[2]+23*y[3]+28*y[4]);
    @addConstraint(q, 2*y[1]+3*y[2]+4*y[3]+5*y[4]<=xi[1,s]-x[1]);
    @addConstraint(q, 6*y[1]+1*y[2]+3*y[3]+2*y[4]<=xi[2,s]-x[2]);
end
DSPsolver.loadProblem(m); # Load model m to DSP
DSPsolver.solve(DSP_SOLVER_DD); # Solve problem using dual decomposition
MPI.Finalize(); # Finalize MPI
```

Computational Experiment: Stochastic Unit Commitment

- Unit Commitment with Uncertain Wind Power Generation [1]
  - IEEE 118-bus system (54 generators and 186 transmission lines)
  - 17 generators have quick-start capabilities on demand.
  - Demand load: 3,095 MW on average with a peak of 3,733 MW
  - 3 identical wind farms
  - Wind power: 494 MW on average with a peak of 916 MW (from 64 scenarios)
  - With security constraints:
    - Network flow constraints
    - Transmission line constraints
    - Ramping constraints
    - Minimum up/down constraints
    - Spinning reserve constraints

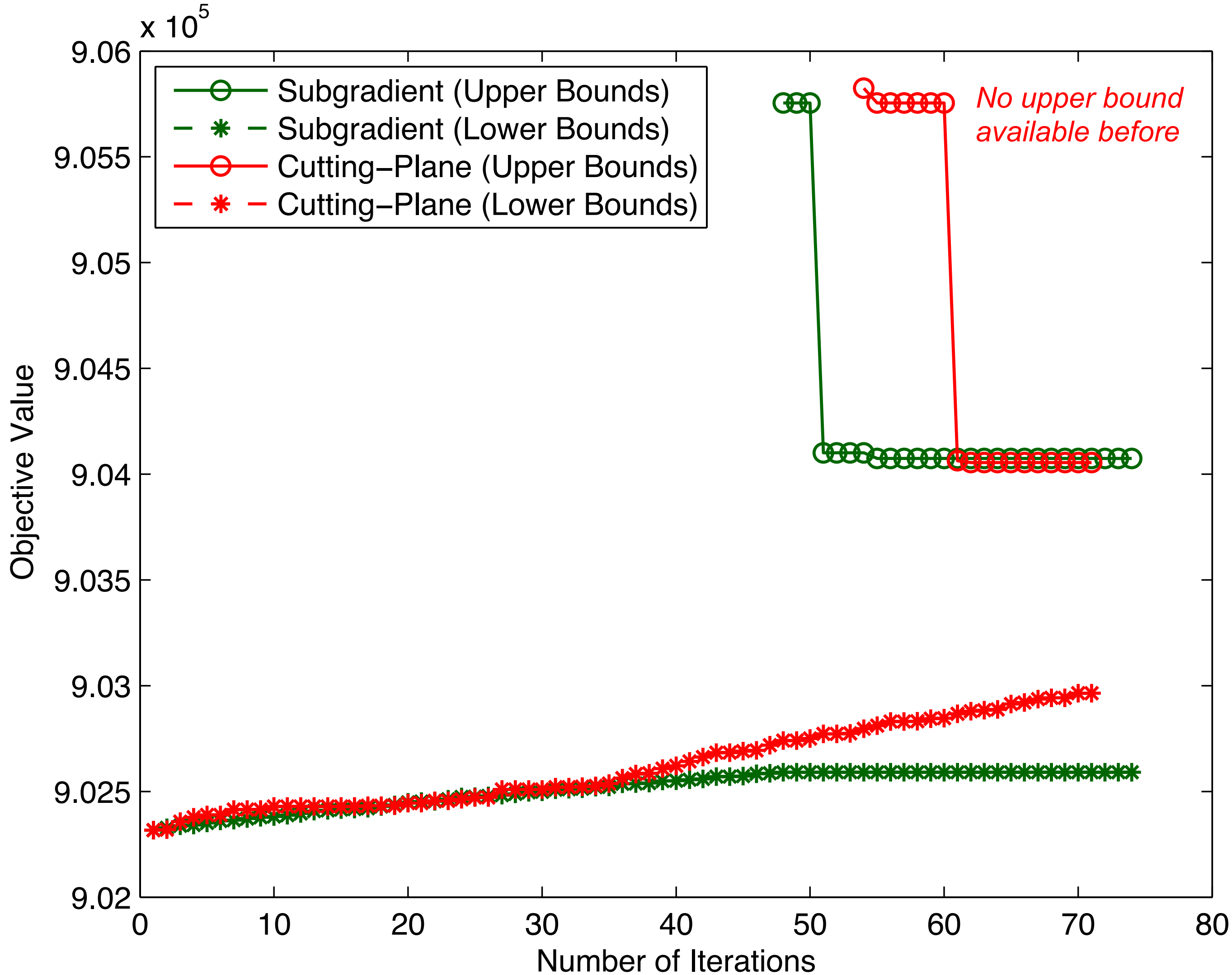
Scenarios	# Rows	# Columns	# Integers
4	120,015	38,880	2,592
8	229,303	75,168	4,320
16	447,879	147,744	7,776
32	885,031	292,896	14,688
64	1,759,335	583,200	28,512



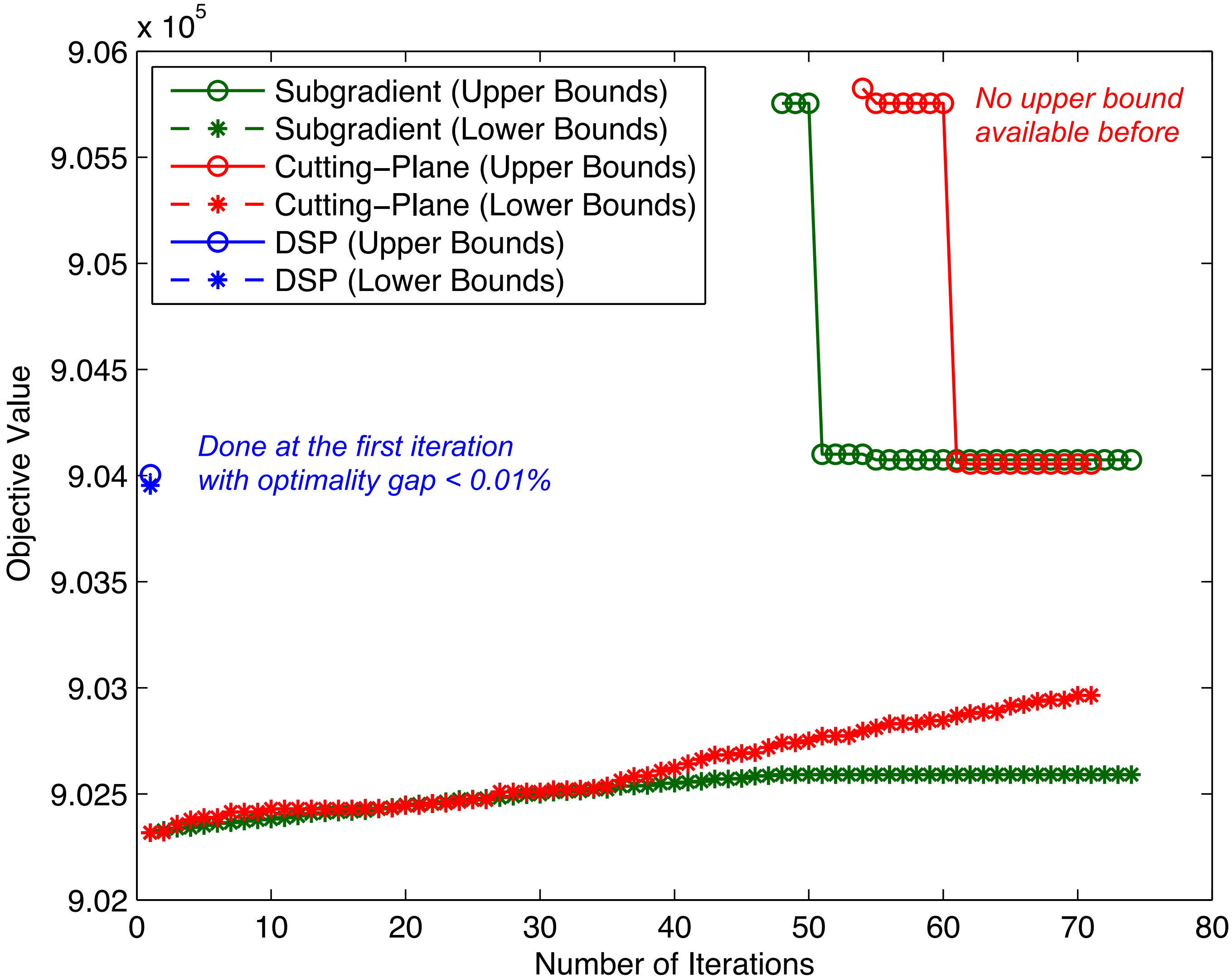
[Scenarios generated from Illinois data by Bessac]

1. Lee, C., Liu, C., Mehrotra, S., Shahidehpour, M. (2014). Modeling Transmission Line Constraints in Two-Stage Robust Unit Commitment Problem. IEEE Transactions on Power Systems, 29(3), 1221–1231. doi:10.1109/TPWRS.2013.2291498

Computational Experiment: Stochastic Unit Commitment



Computational Experiment:
Stochastic Unit Commitment



Computational Experiment: Stochastic Unit Commitment

Numerical results from solving Extensive forms with SCIP

Scenarios	Number of B&C nodes	Upper Bound	Lower Bound	Gap (%)	Wall time (sec.)
4	88831	907035.3	906089.9	0.01	6632
8	58235	904068.1	903567.8	0.05	> 21600
16	3505	900806.1	900200.3	0.07	> 21600
32	9	907536.0	901759.8	0.64	> 21600
64	1	∞	33605.4	∞	> 21600

Numerical results from DSP

Scenarios	Iter	Upper Bound	Lower Bound	Gap (%)	Wall time (sec.)
4	1	907046.1	906979.1	< 0.01	590
8	1	904006.6	903953.5	< 0.01	785
16	1	900706.3	900650.7	< 0.01	1293
32	18	903227.7	903149.9	< 0.01	19547
64	5	895118.0	894756.5	0.04	> 21600

Dual Decomposition

- Dualizing non-anticipativity constraints:

$$D(\lambda) := \min_{x_s, y_s} \left\{ \sum_{s \in \mathcal{S}} L_s(x_s, y_s, \lambda) : (x_s, y_x) \in G_s, \ s \in \mathcal{S} \right\}$$

$$L_s(x_s, y_s, \lambda) = p_s \left(c^T x_s + q_s^T y_s \right) + \lambda^T \left(H_s x_s \right)$$

- Scenario *decomposition*:

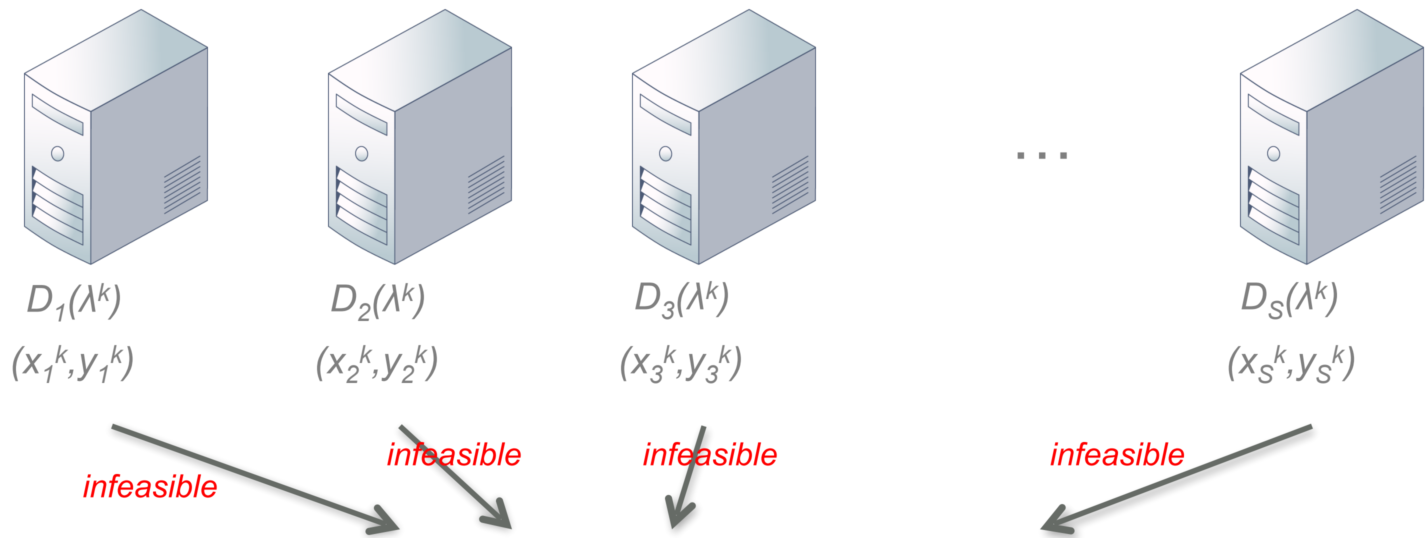
$$D(\lambda) = \sum_{s \in \mathcal{S}} D_s(\lambda) \qquad D_s(\lambda) := \min_{x_s, y_s} \left\{ L_s(x_s, y_s, \lambda) : (x_s, y_x) \in G_s \right\}$$

- *Lower bound* of SMIP for any given λ
- Seek to obtain the best lower bound by solving:

$$z_{\text{UB}} \geq z \geq z_{\text{LD}} := \max_{\lambda} \sum_{s \in \mathcal{S}} D_s(\lambda)$$

Algorithmic Innovation 1: Tightening Inequalities for the Subproblems

- Case 1: A feasible subproblem solution that is infeasible with respect to SMIP.

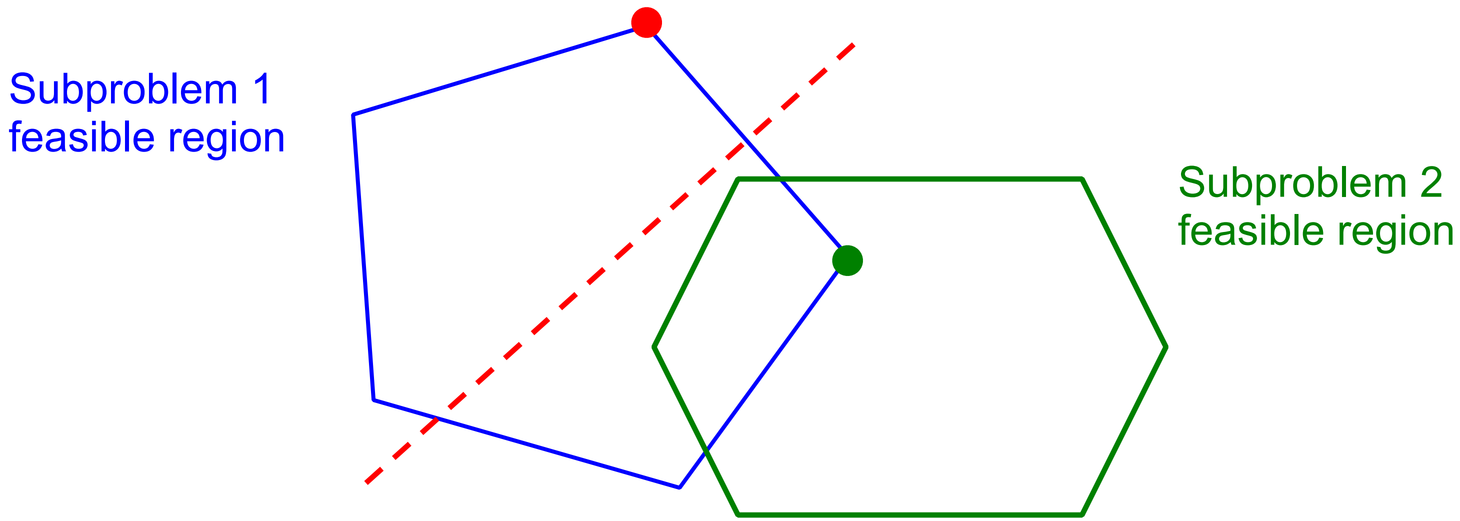


$$
\begin{aligned}
z_{\mathrm{UB}}(x) = c^T x + \min_{y_s} \quad & \sum_{s \in \mathcal{S}} p_s q_s^T y_s \\
\text{s.t.} \quad & W_s y_s = h_s - T_s x, \ \ s \in \mathcal{S}, \\
& y_s \in Y, \ \ s \in \mathcal{S}
\end{aligned}
$$

NOTE: can also be decomposable in s

Algorithmic Innovation 1:
Tightening Inequalities for the Subproblems

- Case 1: A feasible subproblem solution that is infeasible with respect to SMIP.

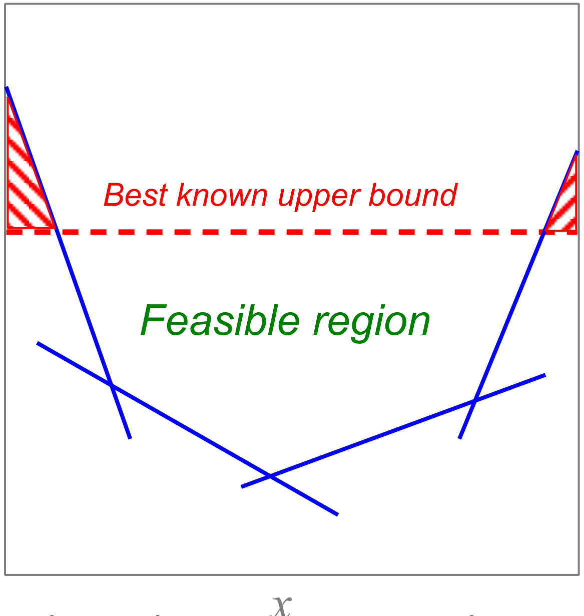


- **IDEA:** Finding a separating hyperplane by Farkas' lemma to exclude the infeasible solution
- **Finiteness:** There are only a finite number of optimal bases to the problem.

Algorithmic Innovation 1: Tightening Inequalities for the Subproblems

- Case 2: A feasible subproblem solution that is also feasible with respect to SMIP.

$$c^T x + \min_{y_s} \sum_{s \in \mathcal{S}} p_s q_s^T y_s$$



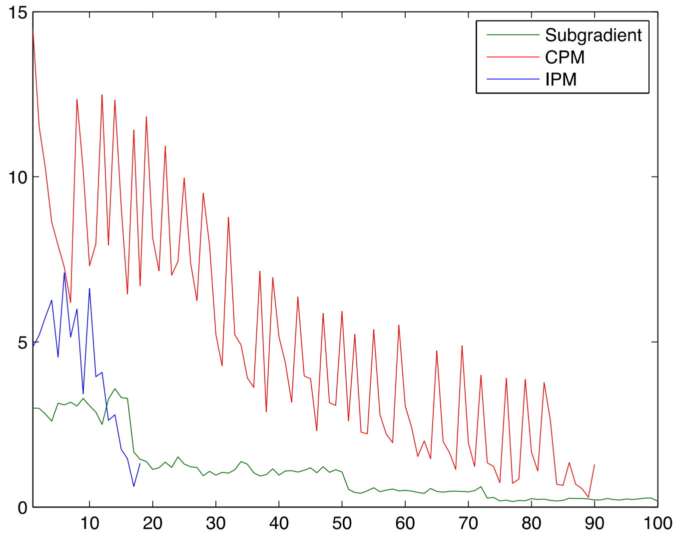
- **IDEA:** Finding a supporting hyperplane that represents a lower bound of the objective function of SMIP
- ***Parametric cut*:** The inequality is parameterized by upper bound z_{UB}.
- **Finiteness:** There are only a finite number of optimal bases to *each* problem.

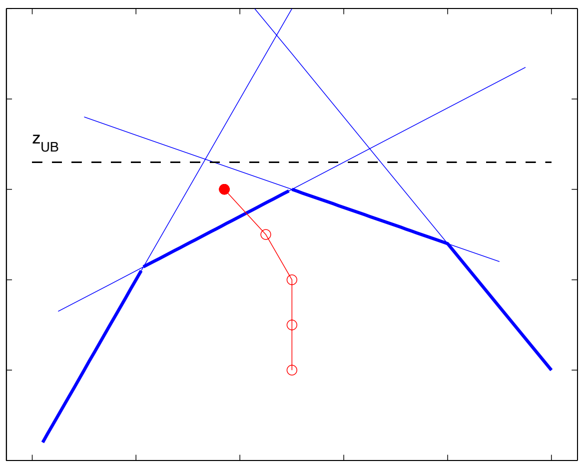
Algorithmic Innovation 2: Interior-Point Cutting-Plane Method

- The solution of the Master problem can oscillate significantly and suffer from degeneracy.
- **Recipe:** Interior-Point Cutting-Plane Method
  - Enabling us to use interior points of the polyhedral set of feasible solutions
    - as compared to an extreme point
  - NOTE: The master has inequalities only.
- ***Prioritized Termination Criteria:***

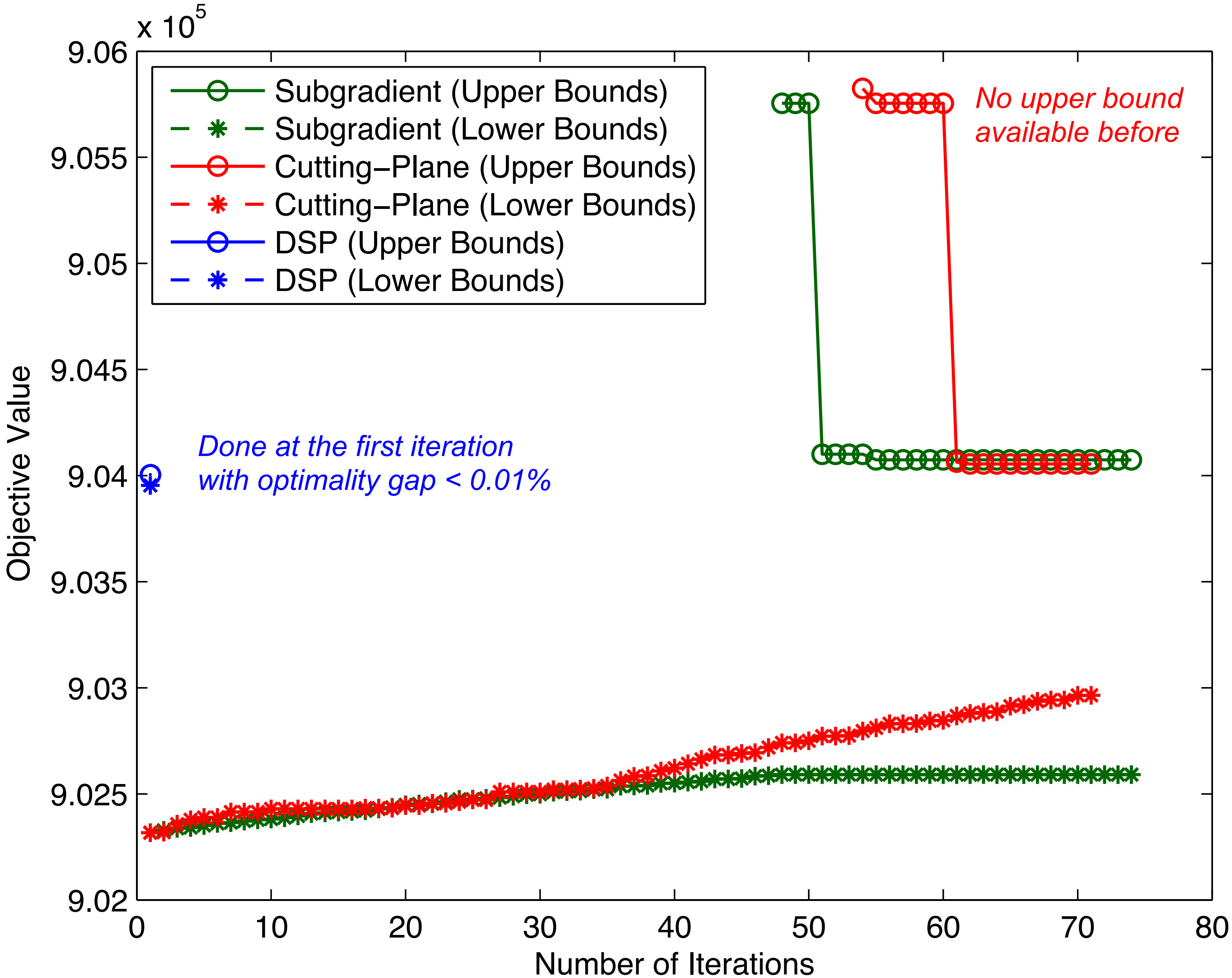
1. $\sum_{s \in \mathcal{S}} \theta_s^k \geq z_{\text{UB}}$
2. $\text{Duality gap} < \epsilon_{\text{IPM}}^k$

Finite convergence: by making sure the contraction of the termination gap

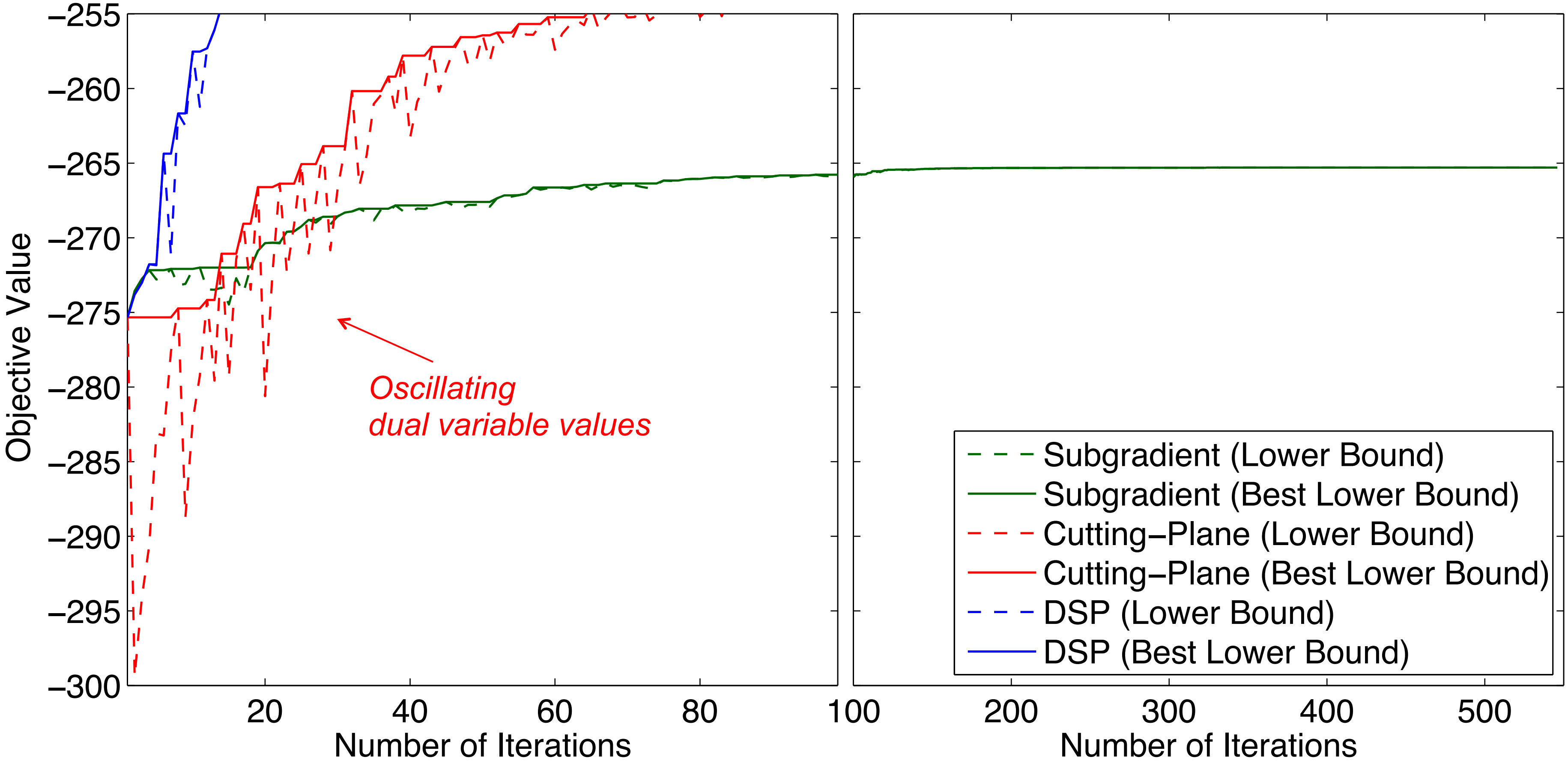




Computational Experiment: Stochastic Unit Commitment



Computational Experiment 1:
SIPLIB (sslp_15_45_15) - Solution Progress



Concluding Remarks

- DSP: an open-source software implementation
  - Dual decomposition and Benders decomposition
  - Interface for Julia-StochJuMP (algebraic modeling language for stochastic programming)
  - Significant improvements in the quality of the bounds, the number of iterations and solution time
- Algorithmic Innovations for Dual Decomposition
  - Tightening inequalities for the subproblem
  - Interior-point cutting-plane method for solving the master problem
  - Finite convergence
  - Parallelization
- Future research work
  - Parallelizing the master by PIPS
  - Asynchronous parallelism
  - Other decomposition structures

Thank You

- DSP software package is available from
  - https://github.com/kibaekkim/DSP
  - Including *StochJuMP* script for stochastic unit commitment

- Research paper is available from
  - http://www.optimization-online.org/DB_FILE/2015/06/4960.pdf