Recent advances in the branch-and-cut approach for solving stochastic integer programs

Jim Luedtke

Dept. of Industrial and Systems Engineering
Wisconsin Institute for Discovery
University of Wisconsin-Madison, USA
jim.luedtke@wisc.edu

XVII Conference on Stochastic Programming
July 30, 2025

General Model

Two-Stage Stochastic Mixed Integer Program (SMIP)

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

where $\xi = (q, h, T, W)$ and

$$\begin{aligned} Q(x,\xi) = \min\ & q^\top y \\ \text{s.t.}\ & Wy = h - Tx \\ & y \in \mathbb{R}_+^{n_2} \times \mathbb{Z}_+^{p_2} \end{aligned}$$

- x: first-stage decision variables
- y: second-stage decision variables
- Sometimes assume n_1, p_1, n_2, or p_2 are zero

Agenda

1. Example SMIP applications
   - Power grid investment planning (x2)
   - Data center management
2. Progress in the branch-and-cut approach for solving SMIP
   - Overview of the general approach
   - Some recent progress
3. Outlook: What next for stochastic integer programming?

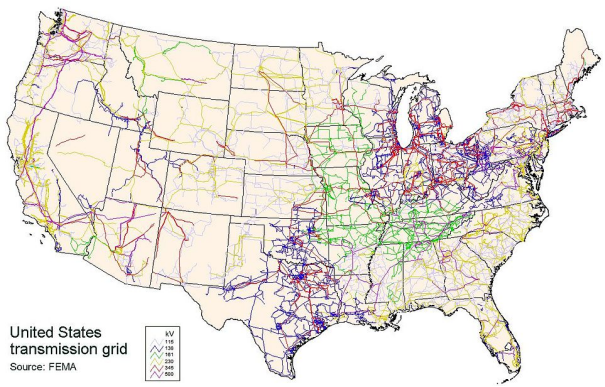
Dedicated to Shabbir and Maarten

Application 1: Generation Expansion Planning

Collaborators: Ramsey Rossmann, Line Roald (UW-Madison), Mihai Anitescu, Julie Bessac, Mitchell Krock (Argonne)

Given candidate generators (sizes/locations): Which to open/what sizes to...

- Minimize capacity expansion and average operating costs to meet demands?
- Minimize risk of load shed in extreme temperature events?



Generation Expansion Planning (cont'd)

First-stage decisions

- For each candidate generator location and type: Install generator or not (binary)? Size (continuous)?

Uncertainty

- Temperatures (spatially correlated), generator outages/capacity reductions (temperature dependent!), load, solar/wind outputs

(Overly) Simplified recourse model

- Just require total generation $\geq$ total load
- Ignores: transmission network/power flow, intertemporal operating constraints, intertemporal uncertainty correlations

Explicit bi-objective model: Minimize operational costs (on "normal" scenarios) and risk (load shed on "extreme" scenarios)

- Claim: nearly *all* stochastic programs should be considered as bi-objective

Generation Expansion: Takeaways

Models nontrivial to solve even with simplistic operational model

- Large-scale: Over 2000 first-stage variables, and many scenarios needed for approximating expected cost

Application-driven pre-selection of "extreme" scenarios (low/high temperatures) enhanced solvability

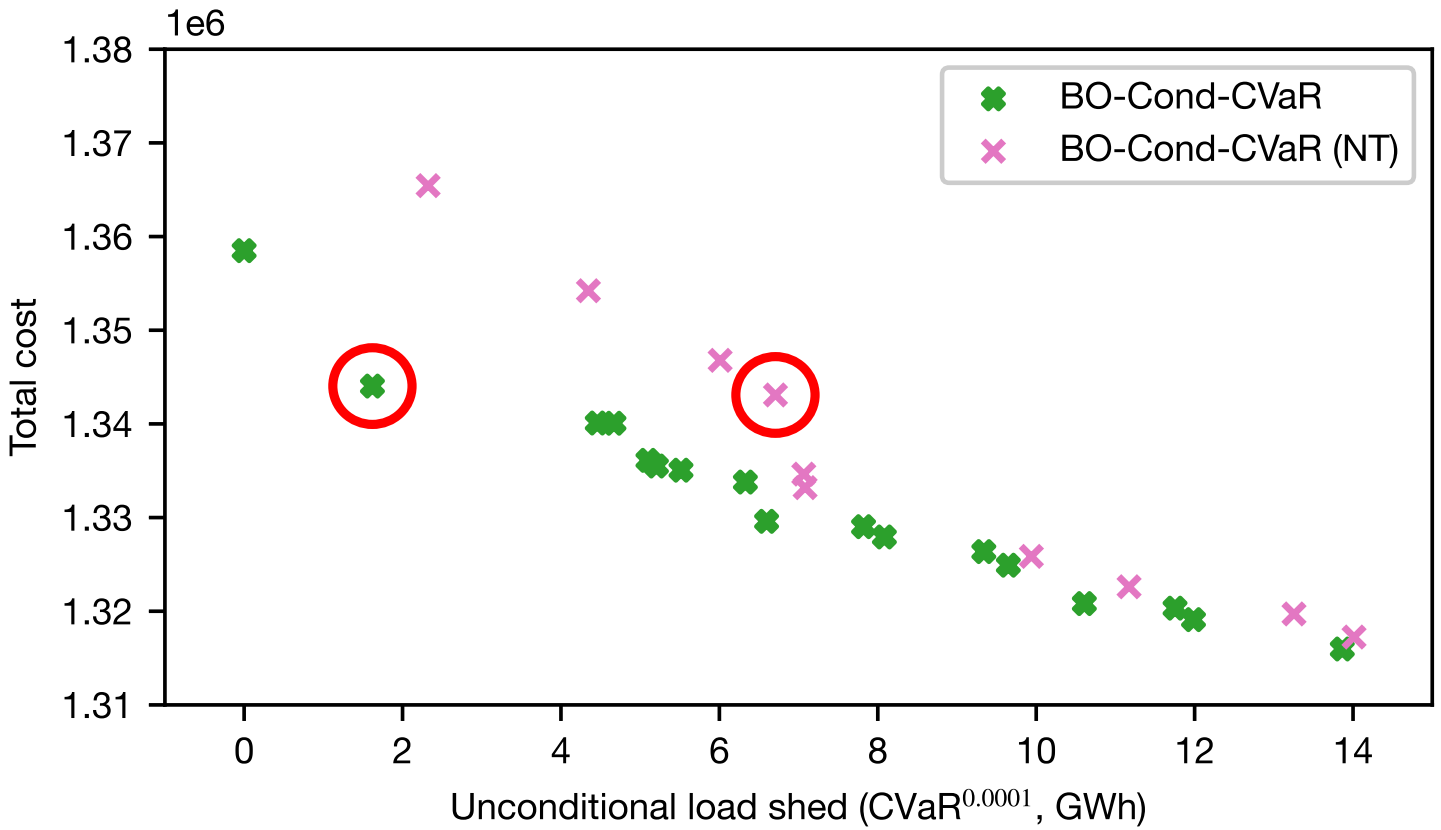
- Enabled generating low-risk solutions with modest number of scenarios in risk objective

Proper stochastic modeling of spatial dependence of random variables: challenging and important

- Incorrectly assume locational independence $\Rightarrow$ Impossible to find low-risk solutions

Generation Expansion: Temperature Impact

Comparing Pareto frontiers of solutions generated by model considering temperature dependence of failures and not (NT)



Application 2: Distributed Energy Resources

Collaborators: Ramsey Rossmann, Line Roald (UW-Madison)

Idea: Install distributed energy resources in the *distribution grid* to enable microgrid operation during outages

- Switches: Enable isolating faults to smaller part of grid
- Thermal generators: Emergency power generation
- Batteries: Meet power via stored energy
- Solar: Emergency and normal energy distribution

Problem statement

Given resilience budget and solar budget, what should be installed and where to minimize average load shed *when an outage occurs*

Distributed Energy Resources: Two-stage Model

First-stage decisions

- Where to install switches, generators, batteries, etc.? Sizes? (binary/continuous)

Uncertainty

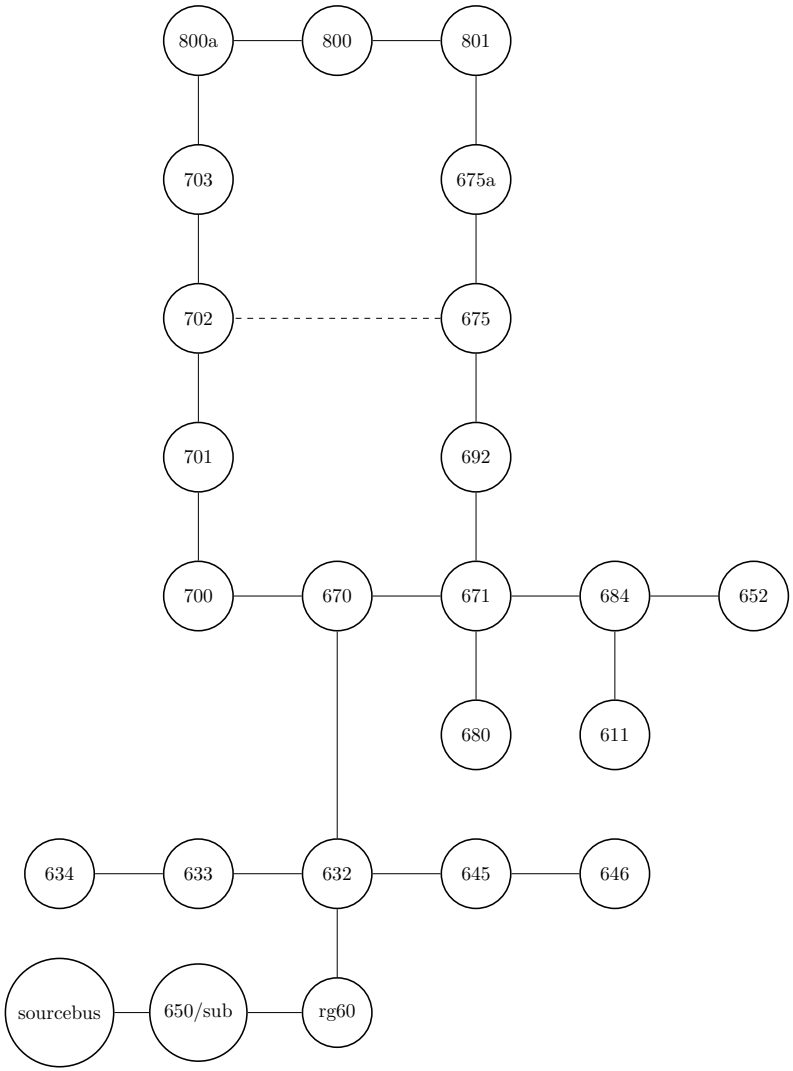
- Line failures (at least one), solar outputs, demands, battery status, outage duration

Recourse model

- Minimize load shed
- Determine switch positions (binary), generator outputs (continuous), and power flows (continuous)
- Key constraints: isolate outages, power flow, line and generator capacity
- Two variations: dispatchable and nondispatchable load

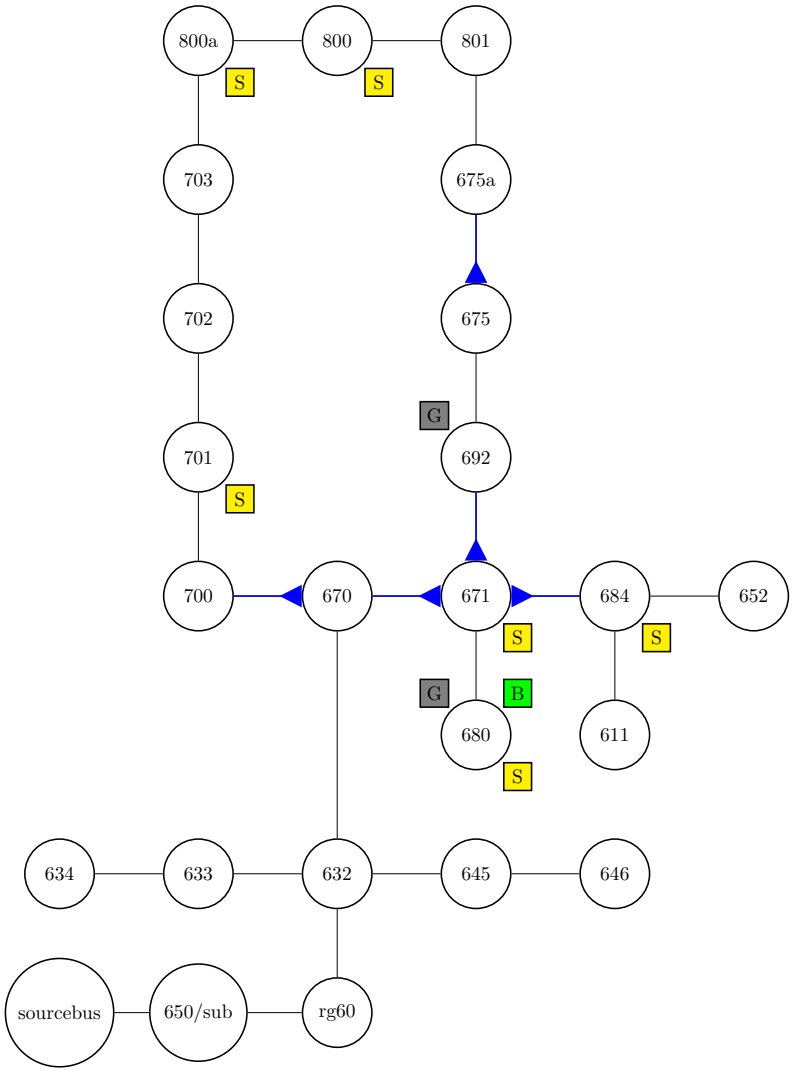
Distribution Energy Resources Illustration

Distribution feeder



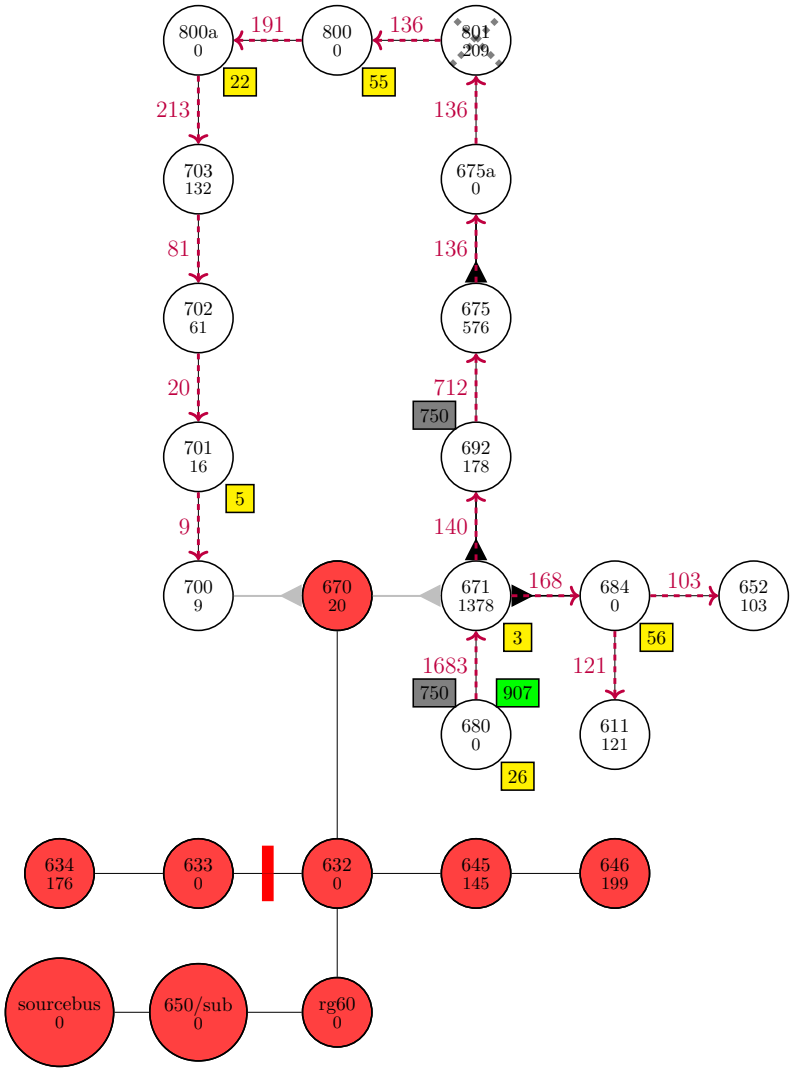
Distribution Energy Resources Illustration

Investments (first-stage decisions)



Distribution Energy Resources Illustration

Outage and recourse decisions



Distributed Energy Resources: Takeaways

Full model very difficult to solve for modest size instances

- Approximate via LP relaxation of second stage?
- For LP 2nd-stage model, Benders decomposition outperforms extensive form even with small number of scenarios

Integer recourse important for nondispatchable load model

- 2nd-stage LP relaxation $\Rightarrow$ 7% lower bound gap
- Solution from LP relaxation $\Rightarrow$ 11% gap to optimal value

Integer recourse less important for dispatchable load model

- 2nd-stage LP relaxation $\Rightarrow$ 1.4% lower bound gap
- Solution from LP relaxation $\Rightarrow$ 1.2% gap to optimal value

Application 3: Managing Data Centers

Collaborators: Harshit Kothari (UW-Madison), Luke Marshall, Konstantina Mellou (Microsoft Research)



Problem: Assign and re-assign jobs (e.g., databases) to servers

- Given a set of servers with some jobs currently running on them and a new set of jobs
- Assign the new jobs to machines, and potentially re-assign current jobs to make them fit
- Costs: Moving jobs between machines and rejecting jobs
- True problem is dynamic problem with this process repeated over time

Managing Server Clusters

Two-stage approximation

- First stage: (Re-)assignment of jobs to machines (binary)
- Uncertainty: Job sizes change, some leave, new jobs arrive
- Second stage: (Re-)assignment of jobs to machines (binary)
  - Second stage has a longer horizon ($\Rightarrow$ more jobs/periods) to better capture impact of current assignments

Simulation study

- Solving two-stage model reduces costs by 20% compared to myopic policy
- LP relaxation of second stage insufficient yields policy 10% worse than solving with IP second stage

What Makes SMIP Hard?

Stochastic integer programming combines challenges from Stochastic programming and Integer Programming

Stochastic programming challenges

- Evaluating expectation
- Huge size even with finite scenario approximation

Integer programming challenges

- Huge number of discrete options
- Weak relaxations can lead to huge branch-and-bound trees

Method 1: Extensive Form

Introduce variables to represent recourse actions for each scenario

$$\begin{aligned} \min\ & c^\top x + \sum_{s=1}^{S} q_s^\top y_s \\ \text{s.t. } & T_s x + W_s y_s = h_s, \qquad s = 1, \ldots, S \\ & y_s \in \mathbb{R}_+^{n_2} \times \mathbb{Z}_+^{p_2}, \qquad s = 1, \ldots, S \\ & Ax \geq b, x \in \mathbb{R}_+^{n_1} \times \mathbb{Z}_+^{p_1} \end{aligned}$$

Advantages

- Easy to model
- Solvable by general-purpose MIP solvers $\Rightarrow$ Increasingly often most efficient available option

Key reason for success of MIP solvers: Well-developed techniques for generating strong valid inequalities

- But, eventually hits memory/time limits due to size

Method 2: Cut-Based Decomposition

Key reformulation idea:

$$\begin{aligned} \min\ & c^\top x + \sum_{s=1}^{S} p_s \theta_s \\ \text{s.t. } & Ax \geq b \\ & (x, \theta_s) \in P_s, \quad s = 1, \ldots, S \\ & x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+ \end{aligned}$$

where for $s = 1, \ldots, S$

$$P_s = \left\{ (x, \theta_s) : \theta_s \geq Q_s(x), Ax \geq b, x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+ \right\}$$

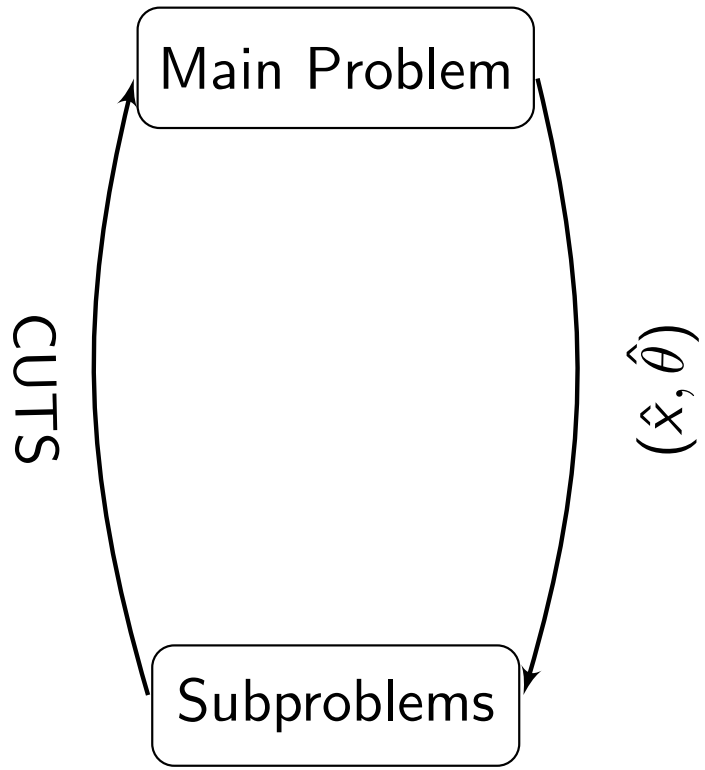
and

$$\begin{aligned} Q_s(x) = \min\ & q_s^\top y \\ \text{s.t. } & W_s y = h_s - T_s x \\ & y \in \mathbb{R}^{n_2}_+ \times \mathbb{Z}^{p_2}_+ \end{aligned}$$

Method 2: Cutting Plane Framework

Relaxed reformulation

$$\begin{aligned} \min\ & c^\top x + \sum_{s=1}^{S} p_s \theta_s \\ \text{s.t. } & Ax \geq b \\ & (x, \theta_s) \in \hat{P}_s, \quad s = 1, \ldots, S \\ & x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+ \end{aligned}$$



Algorithm template: Repeat

- Solve relaxed main problem
- For each s, solve subproblem to *separate* $(\hat{x}, \hat{\theta}_s)$ from P_s

Branch-and-cut (single tree) implementation: Do this with integer restrictions on x relaxed, repeat within branch-and-bound method

Deriving Cuts

Two important purposes of the cuts

1. **Validity:** Ensure the relaxed constraints $(x, \theta_s) \in P_s$ are satisfied
   - Only *necessary* for solutions in which x satisfy the integrality constraints
   - Not always possible to do (e.g., mixed-integer first and second stage) – then purpose is to yield good approximation
2. **Performance:** Improve the LP relaxation of the main problem
   - Since it is being solved via branch-and-bound, this is critical for keeping tree size small enough
   - Important to find inequalities that cut off *fractional* $\hat{x}$ solutions

Benders/L-shaped Cuts

Given relaxed main solution $(\hat{x}, \hat{\theta}_s)$, we seek valid inequalities for:

$$P_s = \left\{(x, \theta_s) : Ax \geq b, \theta_s \geq Q_s(x), x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+\right\}$$

Classic Benders subproblem (of LP relaxation):

$$\begin{aligned} Q_s(\hat{x}) = \min\ & q_s^\top y \\ \text{s.t. } & W_s y \geq h_s - T_s \hat{x} \\ & y \in \mathbb{R}^{n_2}_+ \times \mathbb{R}^{p_2}_+ \end{aligned}$$

Suppose π is a dual feasible solution $(\pi^\top W_s \leq q_s, \pi \geq 0)$:

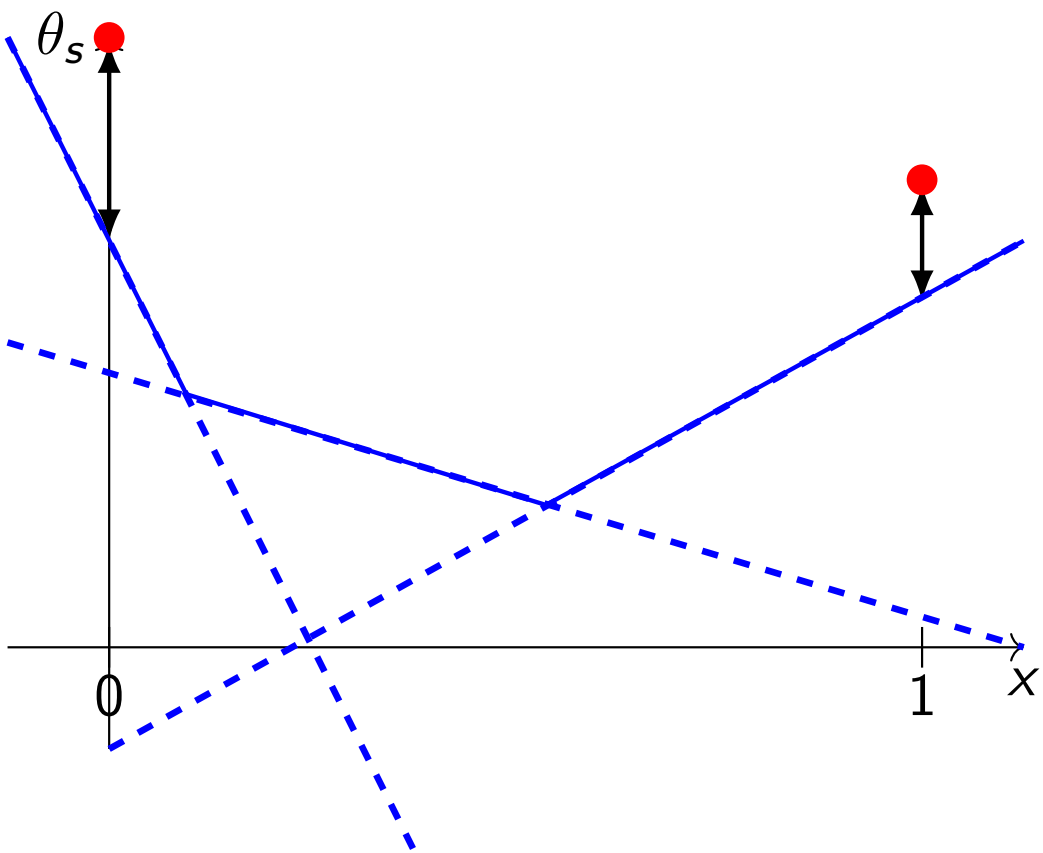
- Yields Benders cut, valid for P_s

$$\theta_s \geq \pi^\top (h_s - T_s x)$$

- NB: Did not use integrality of x or y variables

Dual solution/Benders cut can be found by solving classic subproblem, or other variants: [Magnanti et al., 1986], [Fischetti et al., 2010], etc.

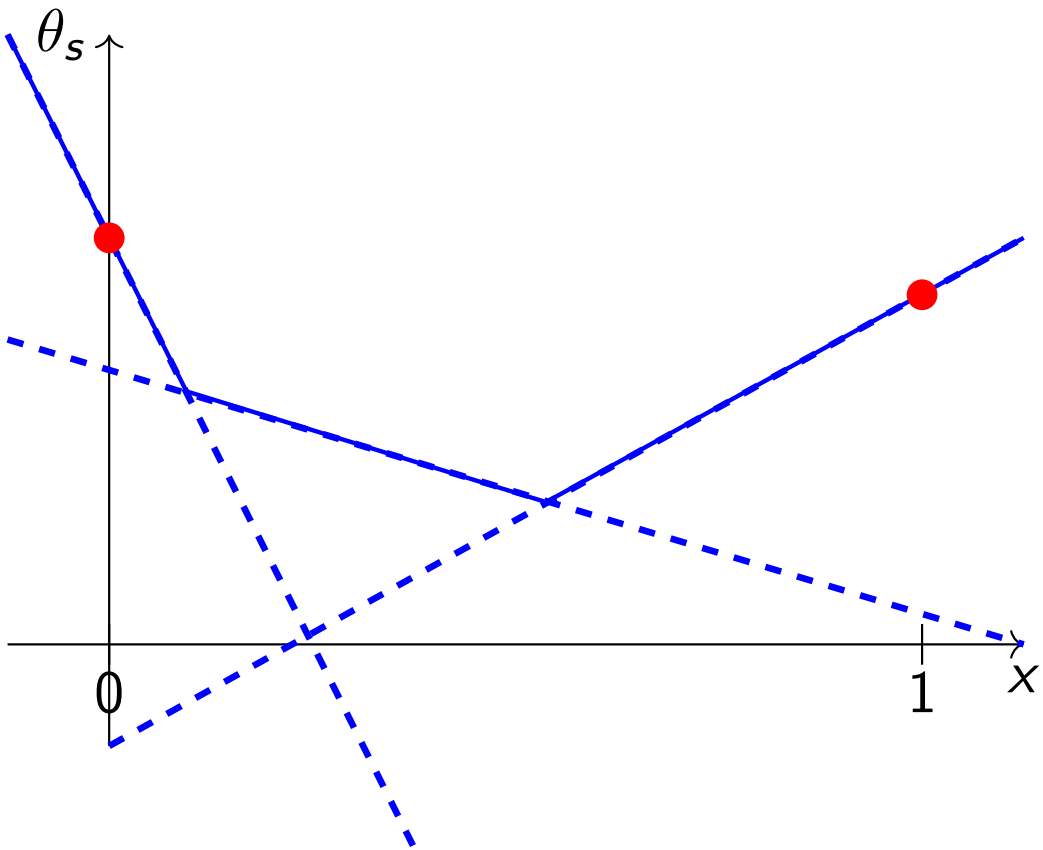
Visualization: Mixed-Integer Recourse



- Red dots: True values $(x, Q_s(x))$ at integer solutions
- Blue lines: Benders cuts

- Gap is due to LP relaxation of second-stage
- Approximation at fractional solutions $x \in (0, 1)$ matters when doing branch-and-bound

Visualization: Continuous Recourse



- Red dots: True values $(x, Q_s(x))$ at integer solutions
- Blue lines: Benders cuts

- No gap at integer solutions $\Rightarrow$ Benders cuts sufficient for *validity* in case of continuous recourse
- Still room for improvement in *performance*: Equivalent to using basic LP relaxation of extensive form (i.e., *very bad*)

How to Achieve Validity Via Cuts?

Easy cases

- Continuous recourse: Benders cuts are enough
- Pure binary first-stage: Integer L-shaped cuts [Laporte and Louveaux, 1993]

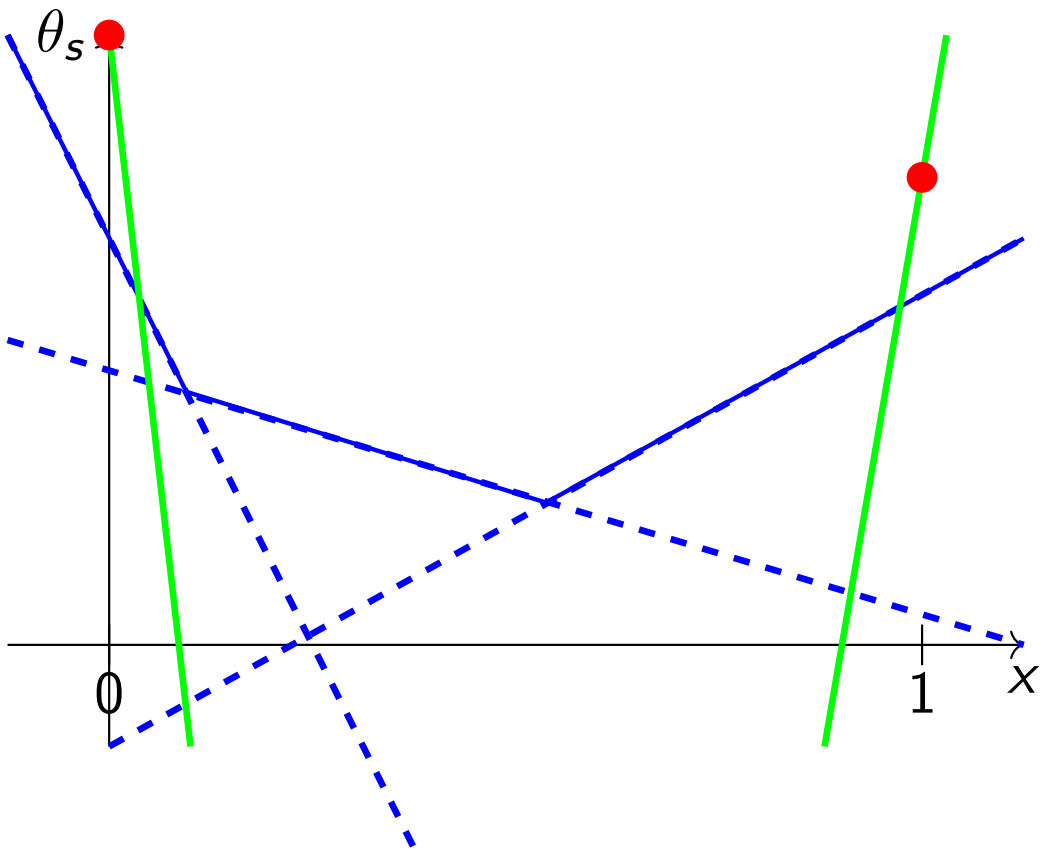
Integer L-shaped cuts

Assume model includes constraints $x \in \{0,1\}^{n_1}$ and $Q_s(x) \geq L$ for all x. Let $\hat{x} \in \{0,1\}^{n_1}$. Then the following inequality is valid:

$$\theta_s \geq Q_s(\hat{x}) - (Q_s(\hat{x}) - L)\left(\sum_{j:\hat{x}_j=1}(1 - x_j) + \sum_{j:\hat{x}_j=0} x_j\right)$$

- Integer L-shaped cuts are sufficient to define $Q_s(x)$ at integer points
- Calculating an integer L-shaped cut requires solving recourse (MIP) to evaluate $Q_s(\hat{x})$

Achieves Validity, But Weak!



- Red dots: True values $(x, Q_s(x))$ at integer solutions
- Blue lines: Benders cuts
- Green lines: Integer L-shaped cut

Integer *L*-shaped cuts are very weak away from defining point

- Must be combined with cuts that improve *performance* – at least Benders cuts, e.g., [Angulo et al., 2016]

Validity Beyond These Cases?

Approaches

- Cuts that share information across scenarios
  [van der Laan and Romeijnders, 2024]
- Nonlinear/nonconvex cuts: [Carøe and Tind, 1997,
  Philpott et al., 2020, Deng and Xie, 2024, Ahmed et al., 2022]
- Specialized branching:
  [Ahmed et al., 2004, Alonso-Ayuso et al., 2003]
- Dynamically add *binary variables and cuts*: [Yan and Yang, 2025]

Practical approximation

- Enforce integrality in first-stage, relax in second
- Improve via valid inequalities that use second-stage integrality

Improving Benders Cuts

Key Idea

Use valid inequalities to obtain stronger LP relaxation of each scenario mixed-integer set:

$$X_s := \{(x, y) : Ax \geq b,\ T_s x + W_s y = h_s \\ x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+, y \in \mathbb{R}^{n_2}_+ \times \mathbb{Z}^{p_2}_+\}$$

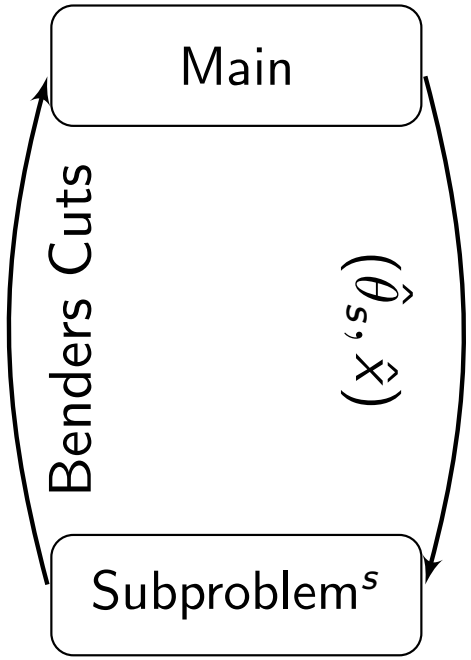
Cuts generated for a single scenario $\Rightarrow$ Can still apply Benders decomposition

Idea has been explored for many classes of cuts

- Split cuts: [Carøe and Tind, 1997, Bodur et al., 2017, Tanner and Ntaimo, 2008, Sen and Higle, 2005]
- Gomory cuts: [Gade et al., 2014, Zhang and Küçükyavuz, 2014]
- Fenchel cuts: [Ntaimo, 2013]
- Multi-term disjunctive cuts:[Sen and Sherali, 2006]
- Reformulation linearization technique: [Sherali and Zhu, 2007]

Implementing Cuts in Subproblem

$$(\text{MP})_t^{LP} : \min_{\theta,x} \; c^T x + \sum_{s=1}^{S} p_s \theta_s$$
$$\text{s.t. } Ax \geq b, x \in \mathbb{R}_+^{n_1} \times \mathbb{R}_+^{p_1}$$
$$(\theta_s, x) \in \hat{P}_s, \quad s = 1, \ldots, S$$



$$(\text{SP})^s : Q_s(\hat{x}) := \min_{y_s} \; q_s^T y_s$$
$$\text{s.t. } W_s y_s \geq h_s - T_s \hat{x}$$
$$y \in \mathbb{R}_+^{n_2} \times \mathbb{R}_+^{p_2}$$

Work required: Solving modified Benders *LP* subproblems and separating cuts from single scenario sets

Implementing Cuts in Subproblem

$$(\text{MP})_t^{LP} : \min_{\theta,x} \; c^T x + \sum_{s=1}^{S} p_s \theta_s$$

$$\text{s.t. } Ax \geq b, x \in \mathbb{R}_+^{n_1} \times \mathbb{R}_+^{p_1}$$

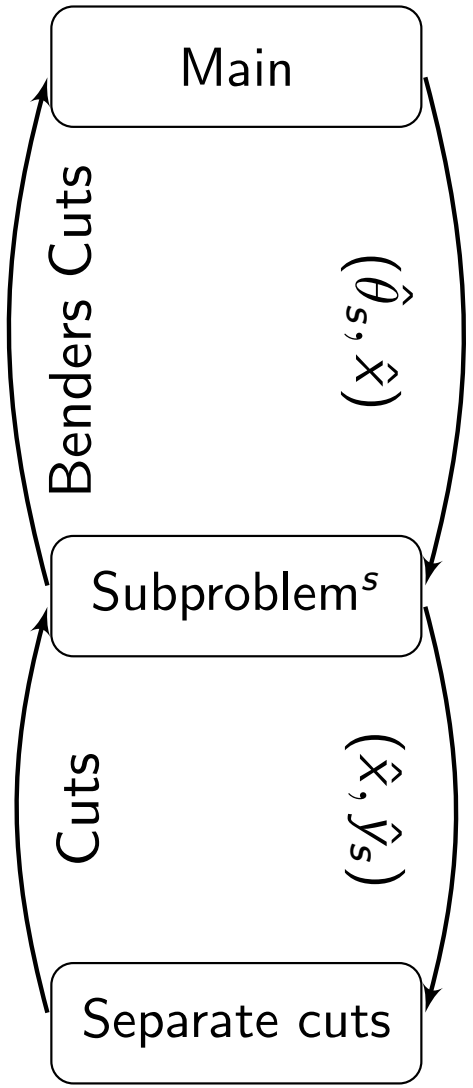
$$(\theta_s, x) \in \hat{P}_s, \quad s = 1, \ldots, S$$

$$(\text{SP})^s : Q_s(\hat{x}) := \min_{y_s} \; q_s^T y_s$$

$$\text{s.t. } W_s y_s \geq h_s - T_s \hat{x}$$

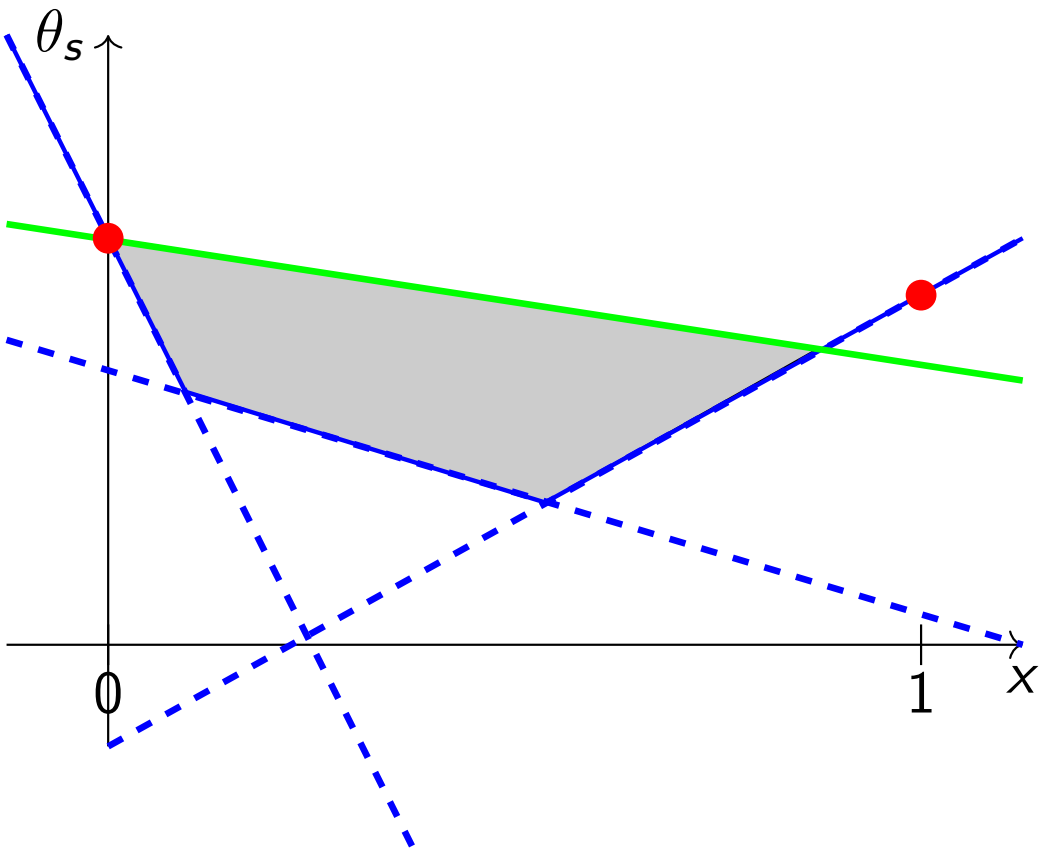
$$\boxed{C_s y_s \geq g_s - D_s \hat{x}}$$

$$y \in \mathbb{R}_+^{n_2} \times \mathbb{R}_+^{p_2}$$



Work required: Solving modified Benders *LP* subproblems and separating cuts from single scenario sets

Potential Impact: Continuous Recourse

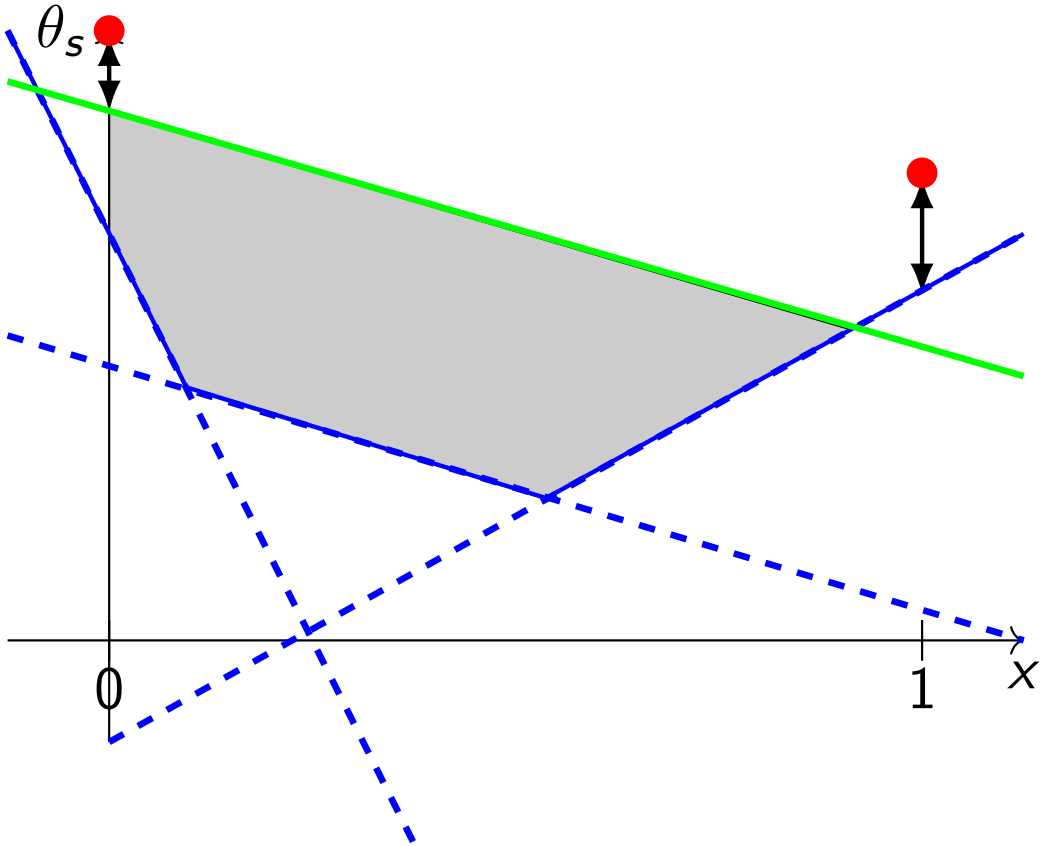


- Red dots: True values $(x, Q_s(x))$ at integer solutions
- Blue lines: LP-Benders cuts
- Green line: Cut-strengthened Benders cut

Better LP relaxations may have (very) significant *performance* impact

- Potential to recover the power MIP solvers have in extensive form

Potential Impact: Mixed-Integer Recourse



- Red dots: True values $(x, Q_s(x))$ at integer solutions
- Blue lines: Benders cuts
- Green line: Cut-strengthened Benders cut

May impact both:

- Validity: Better approximation of value at integer solutions (but still not exact)
- Performance: Better LP relaxation bounds

Cuts in the Subproblems

Question

Given $(\hat{x}, \hat{y}_s)$, how to generate valid inequalities for single scenario mixed-integer set?

$$X_s := \{(x, y) : Ax \geq b,\ T_s x + W_s y \geq h_s \\ x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+, y \in \mathbb{R}^{n_2}_+ \times \mathbb{Z}^{p_2}_+\}$$

General purpose cuts

- Split cuts, Gomory mixed-integer cuts, Chvátal-Gomory cuts,...
- Implemented *very well* in good MIP solvers, but not easy! $\Rightarrow$ Advantage to extensive form...

Problem-specific cuts

- Can add valid inequalities that are derived for particular problem you are solving
- Combinatorial problems, etc.

Lagrangian Cuts

Lagrangian cuts: valid for convex hull of single-scenario projected problem

$$P_s = \left\{(x, \theta_s) : \theta_s \geq Q_s(x), Ax \geq b, x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+\right\}$$

- First introduced for SDDiP [Zou et al., 2019]
- Heuristic use in two-stage stochastic IP: [Rahmaniani et al., 2020]
- Recent work: [Deng and Xie, 2024, Bansal and Küçükyavuz, 2024, Füllner et al., 2024, Yan and Yang, 2025]

Relationship to *dual decomposition* [Carøe and Schultz, 1999]

- All Lagrangian cuts $\Leftrightarrow$ Nonanticipative Lagrangian dual

Highlight joint work with Rui Chen [2023]

- PhD@UW-Madison. Postdoc@Cornell Tech
- Assistant Prof., CUHK Shenzhen

Separating Lagrangian Cuts

For $\pi \in \mathbb{R}^{n_1+p_1}$, $\pi_0 \geq 0$, define:

$$\begin{aligned} Q_s^*(\pi, \pi_0) &= \min_x \left\{ \pi^\top x + \pi_0 Q_s(x) : Ax \geq b, x \in X \right\} \\ &= \min_{x,y} \left\{ \pi^\top x + \pi_0 (q^s)^\top y : (x, y) \in X_s \right\} \end{aligned}$$

- Evaluation: Single-scenario mixed-integer program

Valid inequality for $\text{conv}(P_s)$:

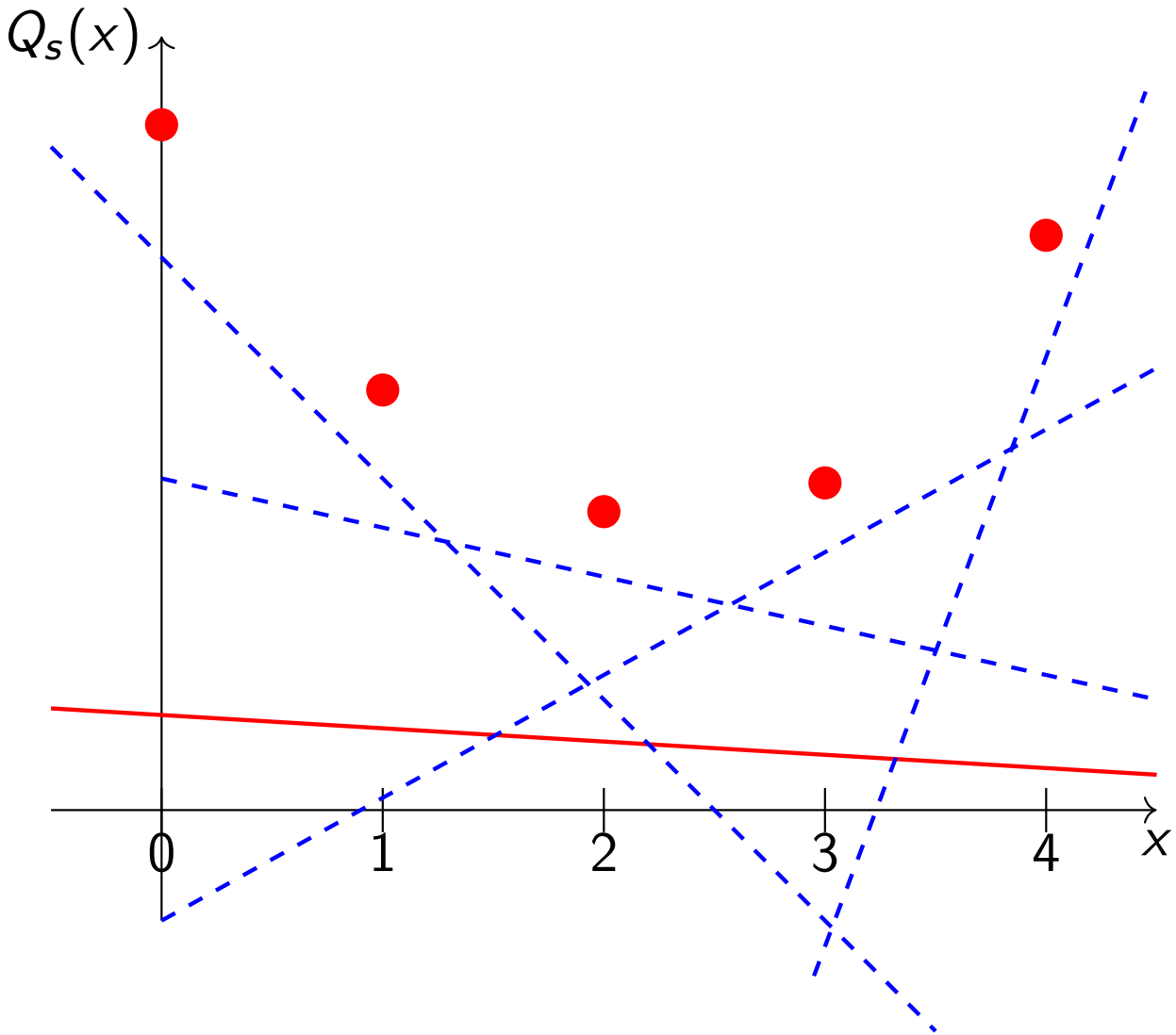
$$\pi^\top x + \pi_0 \theta_s \geq Q_s^*(\pi, \pi_0)$$

- Invariant to scaling by positive value, so can choose a normalization: E.g., $\pi_0 = 1$ or $\|\pi\|_1 \leq 1$

Given relaxation solution $(\hat{x}, \hat{\theta}_s)$, a Lagrangian cut is found by solving

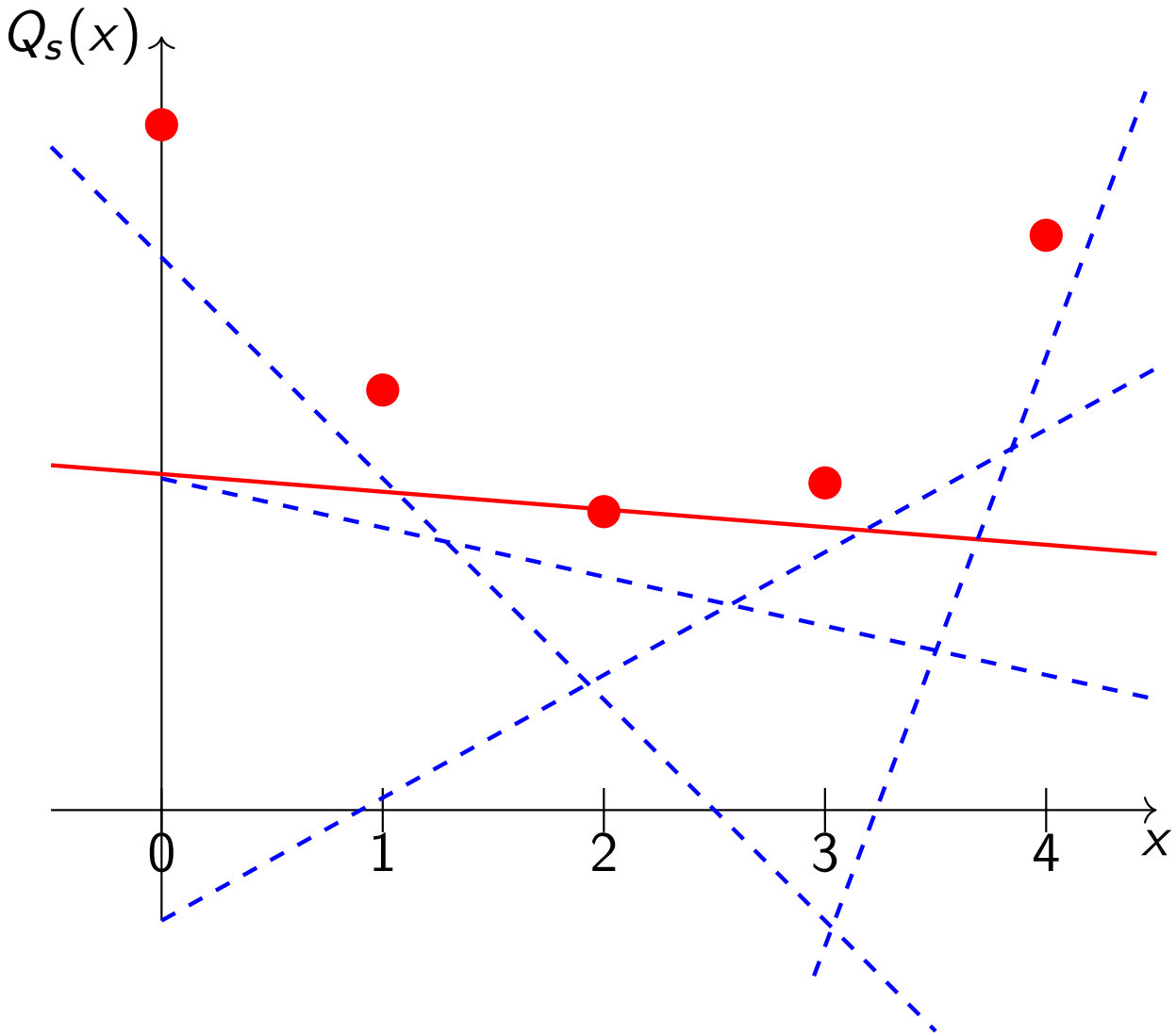
$$\max_{\pi,\pi_0} \{ Q_s^*(\pi, \pi_0) - \pi^\top \hat{x} - \pi_0^\top \hat{\theta}_s : \|\pi\|_1 \leq 1 \}$$

Example



(π, π_0) defines a slope of a candidate cut

Example



$Q_s^*(\pi, \pi_0)$ determines highest it can be to be valid

Separating Lagrangian Cuts

For $\pi \in \mathbb{R}^{n_1+p_1}$, $\pi \geq 0$, define:

$$\begin{aligned} Q_s^*(\pi, \pi_0) &= \min_x \left\{ \pi^\top x + \pi_0 Q_s(x) : Ax \geq b, x \in X \right\} \\ &= \min_{x,y} \left\{ \pi^\top x + \pi_0 (q^s)^\top y : (x, y) \in X_s \right\} \end{aligned}$$

- Evaluation: Single-scenario mixed-integer program

Valid inequality for $\text{conv}(P_s)$:

$$\pi^\top x + \pi_0 \theta_s \geq Q_s^*(\pi, \pi_0)$$

- Invariant to scaling by positive value, so can choose a normalization: E.g., $\pi_0 = 1$ or $\|\pi\|_1 \leq 1$

Given relaxation solution $(\hat{x}, \hat{\theta}_s)$, a Lagrangian cut is found by solving

$$\max_{\pi,\pi_0} \{ Q_s^*(\pi, \pi_0) - \pi^\top \hat{x} - \pi_0^\top \hat{\theta}_s : \|\pi\|_1 \leq 1 \}$$

Separating Lagrangian Cuts (cont'd)

How to solve the cut separation problem?

$$\max_{\pi,\pi_0}\{Q_s^*(\pi,\pi_0) - \pi^\top \hat{x} - \pi_0^\top \hat{\theta}_s : \|\pi\|_1 \leq 1\}$$

Cutting plane algorithm!

- $Q_s^*()$ is piecewise-linear concave $\Rightarrow$ Iteratively approximate via cuts, similar to Benders
- Evaluation and supergradient obtained by solving single-scenario MIP
- Upper bound allows early termination when "approximately most violated" cut found

Lagrangian Cuts (cont'd)

Cutting plane algorithm challenge

- Convergence can be slow (solve many IP subproblems to generate a single cut)

Idea: Heavily restrict cut coefficient space

- Require π to be a linear combination of a small number of previously generated Benders cut coefficients
- Low-dimensional space $\Rightarrow$ Significantly faster convergence (empirically)
- With good choice of restriction, loss in cut quality not too drastic
- Further improvement possible by using a MIP to *select* the previous Benders cuts to use in restriction

LP Relaxation Results

Test problems

1. Stochastic server location problem (SSLP, Ntaimo and Sen 2005), but *larger*
   - 20-50 first-stage binary variables, 2000 mixed-integer second-stage variables, 50-200 scenarios
2. Stochastic network interdiction problem (SNIP, Pan and Mortaon 2008)
   - 320 first-stage binary variables, 2856 continuous second-stage variables, 456 scenarios

Experiment

- Add cuts to LP relaxation until "tails off"
- Then solve via branch-and-cut adding Benders or Integer L-shaped cuts as needed in tree

Branch-and-Cut Results (SSLP)

	On $\|S\| = 50$ instances			On $\|S\| = 200$ instances		
	LBC	BBC	DSP	LBC	BBC	DSP
# solved	12/12	0/12	12/12	12/12	0/12	7/12
Avg soln time	1496	≥ 7200	1259	4348	≥ 7200	≥ 5365
Avg gap (%)	0.0	18.1	—	0.0	27.0	—
Avg B&C time	25	≥ 7200	—	163	≥ 7200	—
Avg # nodes	775	13808	—	1411	3954	—

- LBC: Lagrangian cuts integrated with branch-and-cut
- BBC: Benders cuts integrated with branch-and-cut
- DSP: a general-purpose SMIP solver based on Lagrangian dual (Kim and Zavala 2018)

Branch-and-Cut Results (SNIP)

Option-(Gurobi cuts): Disabled Gurobi cuts in main problem in branch-and-cut method

	LBC-(Gurobi cuts)	BBC	BBC-(Gurobi cuts)
# solved	20/20	20/20	9/20
Avg soln time	2385	163	$\geq$ 4981
Avg gap (%)	0.0	0.0	2.9
Avg B&C time	71	116	$\geq$ 4534
Avg # nodes	4236	3348	$\geq$ 2923914

- Gurobi cuts in main problem are very effective when added on top of Benders cuts (added to root formulation) for this problem
- They are unnecessary when using restricted Lagrangian cuts

Sparse Multi-term Disjunctive Cuts

Summary of results from [Chen and L. 2024] for binary special case:

$$P_s = \{(x, \theta_s) \in \{0,1\}^{n_1} \times \mathbb{R} : Ax \geq b, \theta_s \geq Q_s(x)\}$$

Sparse cuts

- A cut $\theta \geq \alpha^\top x + \beta$ is sparse if the number of nonzeros in the coefficient vector α is small
- Improves speed of re-solving of linear programs after adding cuts
- [Dey et al., 2018]: Provide conditions when sparse cuts approximate relaxation obtainable by dense cuts

Multi-term disjunctions

- Integer programming technique for generating valid inequalities based on union of polyhedra [Balas, 1979]
- In practice, restricted to at most 2-4 polyhedra (e.g., consider $x_i = 0$ or $x_i = 1$)

Sparse Multi-term Disjunctive Cuts (cont'd)

Idea

Use a small subset of variables I *both* for defining a multi-term disjunction by partial enumeration *and* as restricted support for cut

Key results:

- Generating such a valid inequality can be accomplished by solving $2^{|I|}$ small linear programs *once*, then solving a "cut generating linear program" with $2^{|I|}$ constraints
- Makes it feasible to scale to $|I| \approx 10 - 12$ (i.e., 1,000–4,000 polyhedra)
- Yields significant gap reduction especially for inherently sparse problems

Expanding SMIP's Impact

While applications of SMIP are growing, usage is *far* less than deterministic integer programming

What's missing?

- Hypothesis: User-friendly, reliable, and efficient solver for two-stage SMIP

How to get there?

- Researcher-written code?
  - Integration and fine-tuning of cut generation tools requires significant development time and expertise
- Interest a solver company?
  - To do so, will require interest from customers $\Rightarrow$ Importance of application work

Future Direction: Approximation Quality

Approximations in SMIP

- Cases where nonconvex value function can be *provably* approximated by convex function:
  [Romeijnders et al., 2016, van der Laan and Romeijnders, 2021, van Beesten and Romeijnders, 2022]

Lagrangian dual (e.g., via cuts) as a provable approximation?

- Empirical observation: Lagrangian dual bound is *very* close to optimal value

Questions

- Can we provide theory to justify this?
- How to reliably recover primal solution?
- Are there conditions when we can prove the bound is within a factor?

Future Direction: Interpretable Policies

Multi-stage problems with integer decisions

- Extremely challenging to solve
- Decision-makers may be reluctant to implement decisions provided by "black box" algorithm

Interpretable policies

- Design custom *parameterized decision rules* for mapping states to decisions
- Old idea: trade-off between solution quality/tractability to optimize
- [Hakizimana, 2019] $\Rightarrow$ Conditions that assure objective is *differentiable* $\Rightarrow$ Can apply stochastic gradient algorithm

Example: [Sherkat-Masoumi, Bodur, and L., 2025+]

- Use to optimize vaccine ordering and administration policies
- Optimized policies compared to information relaxation bound computed via two-stage stochastic integer program
- Interpretable policies remarkably close to this bound

Parting Thoughts

- There are many important and impactful applications of SMIP
- Making steady progress on solution methods, but still work to do
- Lots of interesting new directions

Thank you!

- `jim.luedtke@wisc.edu`


Ahmed, S., Cabral, F. G., and Freitas Paulo da Costa, B. (2022).
Stochastic lipschitz dynamic programming.
Mathematical Programming, 191(2):755–793.


Ahmed, S., Tawarmalani, M., and Sahinidis, N. (2004).
A finite branch-and-bound algorithm for two-stage stochastic integer programs.
Mathematical Programming, 100(2):355–377.


Alonso-Ayuso, A., Escudero, L. F., and no, M. T. O. (2003).
Bfc, a branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed 0–1 programs.
European Journal of Operational Research, 151(3):503 – 519.


Angulo, G., Ahmed, S., and Dey, S. (2016).
Improving the integer L-shaped method.
INFORMS Journal on Computing, 28:483–499.


Balas, E. (1979).
Disjunctive programming.

In *Annals of Discrete Mathematics*, volume 5, pages 3–51. Elsevier.


Bansal, A. and Küçükyavuz, S. (2024).
A computational study of cutting-plane methods for multi-stage stochastic integer programs.
arXiv preprint arXiv:2405.02533.


Bodur, M., Dash, S., Günlük, O., and Luedtke, J. (2017).
Strengthened Benders cuts for stochastic integer programs with continuous recourse.
INFORMS Journal on Computing, 29:77–91.


Carøe, C. C. and Schultz, R. (1999).
Dual decomposition in stochastic integer programming.
Operations Research Letters, 24(1-2):37–45.


Carøe, C. C. and Tind, J. (1997).
A cutting-plane approach to mixed 0–1 stochastic integer programs.
European Journal of Operational Research, 101(2):306–316.


Deng, H. and Xie, W. (2024).

On the ReLU lagrangian cuts for stochastic mixed integer programming.
arXiv preprint arXiv:2411.01229.


Dey, S. S., Molinaro, M., and Wang, Q. (2018).
Analysis of sparse cutting planes for sparse MILPs with applications to stochastic MILPs.
Mathematics of Operations Research, 43(1):304–332.


Fischetti, M., Salvagnin, D., and Zanette, A. (2010).
A note on selection of benders' cuts.
Mathematical Programming, 124:175–182.


Füllner, C., Sun, X. A., and Rebennack, S. (2024).
On lipschitz regularization and lagrangian cuts in multistage stochastic mixed-integer linear programming.
Available at Optimization Online.


Gade, D., Küçükyavuz, S., and Sen, S. (2014).

Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs.
Mathematical Programming, 144(1-2):39–64.


Hakizimana, A. (2019).
Novel Optimization Approaches for Integrated Design and Operation of Smart Manufacturing and Energy Systems.
PhD thesis, Clemson University.


Laporte, G. and Louveaux, F. (1993).
The integer L-shaped method for stochastic integer programs with complete recourse.
Operations Research Letters, 13(3):133–142.


Magnanti, T. L., Wong, R. T., et al. (1986).
Decomposition methods for facility location problems.


Ntaimo, L. (2013).
Fenchel decomposition for stochastic mixed-integer programming.
Journal of Global Optimization, 55:141–163.


Philpott, A. B., Wahid, F., and Bonnans, J. F. (2020).
Midas: A mixed integer dynamic approximation scheme.
Mathematical Programming, 181(1):19–50.


Rahmaniani, R., Ahmed, S., Crainic, T. G., Gendreau, M., and Rei, W. (2020).
The Benders dual decomposition method.
Oper. Res., 68(3):878–895.


Romeijnders, W., Schultz, R., van der Vlerk, M. H., and Klein Haneveld, W. K. (2016).
A convex approximation for two-stage mixed-integer recourse models with a uniform error bound.
SIAM Journal on Optimization, 26(1):426–447.


Sen, S. and Higle, J. L. (2005).
The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: set convexification.
Mathematical Programming, 104:1–20.

Sen, S. and Sherali, H. (2006).
Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming.
Mathematical Programming, 106:203–223.

Sherali, H. and Zhu, X. (2007).
On solving discrete two-stage stochastic programs having mixed-integer first- and second-stage variables.
Mathematical Programming, 108:597–616.

Tanner, M. and Ntaimo, L. (2008).
Computations with disjunctive cuts for two-stage stochastic mixed 0-1 integer programs.
Journal of Global Optimization, 58:365–384.

van Beesten, E. R. and Romeijnders, W. (2022).
Parametric error bounds for convex approximations of two-stage mixed-integer recourse models with a random second-stage cost vector.

Operations Research Letters, 50(5):541–547.


van der Laan, N. and Romeijnders, W. (2021).
A loose benders decomposition algorithm for approximating two-stage mixed-integer recourse models.
Mathematical Programming, 190(1):761–794.


van der Laan, N. and Romeijnders, W. (2024).
A converging benders' decomposition algorithm for two-stage mixed-integer recourse models.
Operations Research, 72(5):2190–2214.


Yan, H. and Yang, H. (2025).
Globally converging algorithm for multistage stochastic mixed-integer programs via enhanced lagrangian cuts.
OptimizationOnline.


Zhang, M. and Küçükyavuz, S. (2014).
Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs.

SIAM Journal on Optimization, 24:1933–1951.

Zou, J., Ahmed, S., and Sun, X. A. (2019).
Stochastic dual dynamic integer programming.
Mathematical Programming, 175:461–502.