Split Cuts for Two-Stage Stochastic Integer Programs

Merve Bodur[1] Sanjeeb Dash[2] Oktay Günlük[2] Jim Luedtke[1]

[1]University of Wisconsin-Madison

[2]IBM T.J. Watson Research Center

Aussois 2014

Two-stage stochastic programs

- There are uncertainties in some parameters (ω)
- Make first stage decisions (x)

  $\searrow$

  Observe realizations

  $\searrow$

  Make second stage (recourse) decisions (y)
- Objective: min (1st stage cost) + $\mathbb{E}$ [2nd stage cost]
- Assumptions:
  - Finitely many scenarios ($\mathcal{K}$)
  - First stage: **integer**
  - Second stage: **continuous**

Extensive form

A (very) large scale mixed integer program.

$$
\begin{aligned}
\min\ & c^T x + \sum_{k \in \mathcal{K}} q_k^T y_k \\
\text{s.t. } & T_k x + W_k y_k \geq h_k \ , \quad \forall k \in \mathcal{K} \\
& Ax \geq b \\
& x \in \mathbb{Z}_+^n, y \in \mathbb{R}_+^{tK}
\end{aligned}
$$

Extensive form

Introduce variables to represent objective in each scenario

$$
\begin{aligned}
\min\ & c^T x + \sum_{k \in \mathcal{K}} z_k \\
\text{s.t. } & z_k \geq q_k^T y_k \ , \qquad\qquad \forall k \in \mathcal{K} \\
& T_k x + W_k y_k \geq h_k \ , \quad \forall k \in \mathcal{K} \\
& Ax \geq b \\
& x \in \mathbb{Z}_+^n, y \in \mathbb{R}_+^{tK}, z \in \mathbb{R}^K
\end{aligned}
$$

$Q_k^{LP} := \{(z_k, x, y_k) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}_+^t : T_k x + W_k y_k \geq h_k, z_k \geq q_k^T y_k\}$
$X := \{x \in \mathbb{Z}_+^n : Ax \geq b\}$

Extensive form

$$\boxed{\begin{aligned} \min\ & c^T x + \sum_{k \in \mathcal{K}} z_k \\ \text{s.t. }\ & (z_k, x, y_k) \in Q_k^{LP}, \ \forall k \in \mathcal{K} \\ & x \in X \end{aligned}}$$

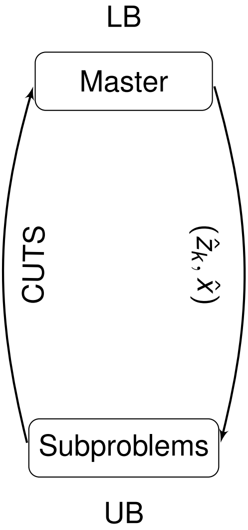
$$Q_k^{LP} := \{(z_k, x, y_k) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}_+^t : T_k x + W_k y_k \geq h_k, z_k \geq q_k^T y_k\}$$
$$X := \{x \in \mathbb{Z}_+^n : Ax \geq b\}$$

Benders Decomposition (L-Shaped Method)

$$
\begin{aligned}
(\mathrm{MP})^{LP} : \min_{z,x} \; & c^T x + \sum_{k \in \mathcal{K}} z_k \\
\text{s.t. } & x \in X^{LP} \\
& \text{Cuts to enforce } z_k \geq f_k(x), \; \forall k \in \mathcal{K} \\
& z \in \mathbb{R}^K
\end{aligned}
$$

$$
\begin{aligned}
(\mathrm{SP})^k : f_k(\hat{x}) := \min_{y_k} \; & q_k^T y_k \\
\text{s.t. } & W_k y_k \geq h_k - T_k \hat{x} \\
& y_k \in \mathbb{R}_+^t
\end{aligned}
$$

- Decomposes by scenario
- Subproblems are small LP's
- Embed within branch-and-cut



Dual decomposition (Carøe and Schultz, 1999)

Copy the first-stage decision-variables

$$\begin{aligned} \min\ & c^T x + \sum_{k \in \mathcal{K}} q_k^T y_k \\ \text{s.t. } & x_k = x, \quad \forall k \in \mathcal{K} \\ & T_k x_k + W_k y_k \geq h_k\ , \quad \forall k \in \mathcal{K} \\ & A x_k \geq b \\ & x_k \in \mathbb{Z}_+^n, y \in \mathbb{R}_+^{tK} \end{aligned}$$

Solve Lagrangian dual relaxing constraints $x_k = x$

- Decomposes problem by scenario
- Subproblems are *mixed-integer programs*

Something in between?

Benders

- Fast solution of LP relaxation (LP subproblems)
- Potentially weak bounds

Dual decomposition

- Expensive relaxation (many MIP subproblems)
- Potentially strong bounds

(Obvious) Idea

Strengthen Benders with integrality-based cuts

Integrality-based cuts in stochastic programming

- Carøe (1998), Sen and Higle (2005), Sen and Sherali (2006), Gade et al. (2012), Zhang and Küçükyavuz (2013), Ntaimo (2013)
- Focus has been on harder case of *second-stage* integer variables
- Common goal: Handle second-stage integrality with cuts only

Benders as a projection

Extensive Form

$$\begin{aligned}\min_{z,x,y}\ & c^T x + \sum_{k \in \mathcal{K}} z_k \\ \text{s.t. } & (z_k, x, y_k) \in Q_k^{LP}\ ,\ \forall k \in \mathcal{K} \\ & x \in X\end{aligned}$$

Benders reformulates the problem in the space of first stage variables.

$$\begin{aligned}\min_{z,x}\ & c^T x + \sum_{k \in \mathcal{K}} z_k \\ \text{s.t. } & (z_k, x) \in \text{Proj}_{z_k, x}(Q_k^{LP})\ ,\ \forall k \in \mathcal{K} \\ & x \in X\end{aligned}$$

Two options for using integrality-based cuts

Project-and-cut:

- Generate Benders cuts to obtain polyhedron B_k with

$$\text{Proj}_{z_k,x}(Q_k^{LP}) \subseteq B_k$$

- Use integrality information in X to derive cuts for the set $B_k \cap X$

Cut-and-project:

- Use integrality information to derive cuts valid for $Q_k^{IP} := Q_k^{LP} \cap X$ to obtain polyhedron Q_k^S with:

$$Q_k^{IP} \subseteq Q_k^S \subseteq Q_k^{LP}$$

- Project resulting polyhedra into (x, z_k) space via Benders

Cut-and-project

$$(\mathrm{MP})^{LP}: \min_{z,x} c^T x + \sum_{k \in \mathcal{K}} z_k$$

$$\text{s.t. } x \in X^{LP}$$

$$\text{Cuts to enforce } z_k \geq f_k(x),\ \forall k \in \mathcal{K}$$

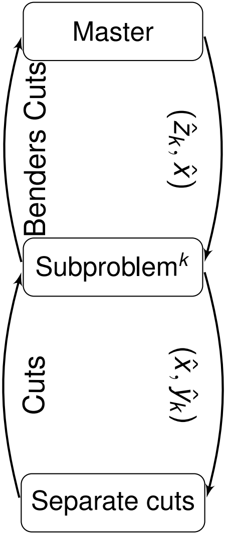
$$z \in \mathbb{R}^K$$

$$(\mathrm{SP})^k: f_k(\hat{x}) := \min_{y_k} q_k^T y_k$$

$$\text{s.t. } W_k y_k \geq h_k - T_k \hat{x}$$

$$y_k \in \mathbb{R}^t_+$$

Note: Add cuts to subproblem, even though it's an LP



Which is better?

Project-and-cut

- \+ Work in more compact space
- \+ Straightforward to generate cuts using multiple scenarios
- \- Lose structure
- \- Working with B_k may lead to weaker cuts
  - Can be overcome

Cut-and-project

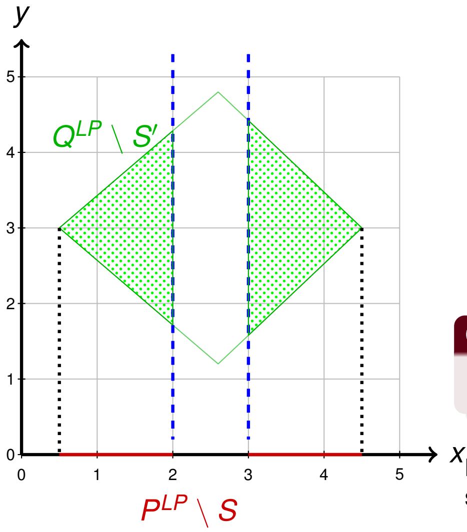
- \+ Keep formulation structure $\Rightarrow$ Can use problem-specific cuts
- \+ Easy to incorporate known cut separation routines
- \- May be memory-intensive

Question

Does one approach yield *stronger* relaxations than the other when using a given class of cuts?

We investigate this for *split cuts*

First Step : Single Split



$Q^{LP} \subseteq \mathbb{R}^{q+n+t}$ (a polyhedron)

$P^{LP} = \text{Proj}_{(z,x)}(Q^{LP})$

$S = \{(z,x) : \gamma + 1 > \pi x > \gamma\}$

$S' = \{(z,x,y) : \gamma + 1 > \pi x > \gamma\}$

Claim

$P^{LP} \setminus S = \text{proj}_{(z,x)}(Q^{LP} \setminus S')$

Modaresi et al. (MIP 2012) showed similar result for conic MIR

Intersectiong multiple splits

Question: $\mathrm{Proj}_{z,x}\left(\bigcap_{i\in I}\mathrm{conv}(Q^{LP}\setminus S_i')\right) \stackrel{?}{=} \bigcap_{i\in I}\mathrm{conv}(P^{LP}\setminus S_i)$

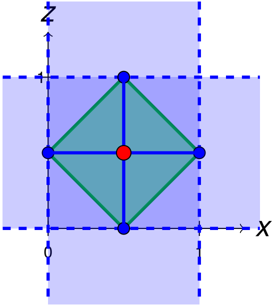
Answer: ($\subseteq$) is easy to show. ($\supseteq$) is false.

Counterexample: (z,x,y) where $z,x\in\mathbb{Z}$, $y\in\mathbb{R}$

$Q^{LP} = \mathrm{conv}(\{(1/2,0,0),(1/2,1,0),(0,1/2,1),(1,1/2,1)\}) \subseteq \mathbb{R}^3$

$P^{LP} = \mathrm{Proj}_{(z,x)}(Q^{LP})$

$SC(P^{LP}) = \{(1/2,1/2)\}$



Again related to Modaresi et al. (MIP 2012)

Intersecting multiple splits

Question: $\text{Proj}_{z,x}\left(\bigcap_{i\in I}\text{conv}(Q^{LP}\setminus S_i')\right) \stackrel{?}{=} \bigcap_{i\in I}\text{conv}(P^{LP}\setminus S_i)$

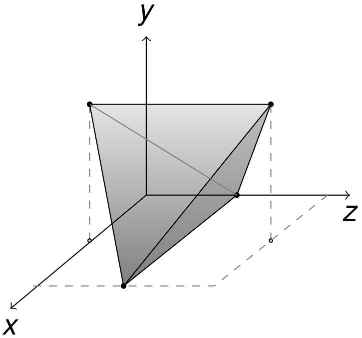
Counterexample: (z, x, y) where $z, x \in \mathbb{Z}$, $y \in \mathbb{R}$

$Q^{LP} = \text{conv}(\{(1/2, 0, 0), (1/2, 1, 0), (0, 1/2, 1), (1, 1/2, 1)\}) \subseteq \mathbb{R}^3$

$P^{LP} = \text{Proj}_{(z,x)}(Q^{LP})$

$SC(P^{LP}) = \{(1/2, 1/2)\}$

$SC(Q^{LP}) = \emptyset$



Can weakness in projected space be overcome?

Let

- $Q^{LP} = \{(z, x, y) \in \mathbb{R}^{q+n+t} \; : \; Hx + Kz + Ly \; \geq \; g, \; x, y, z \geq 0\}$.
- $S_i' = \{(x, z, y) \in \mathbb{R}^{n+q+t} \; : \; \gamma^i + 1 > \pi^i x > \gamma^i\}$ for $i \in I$
- $$\hat{P} \; = \; \mathrm{proj}_{(x,z)} \left(\bigcap_{i \in I} \mathrm{conv}(Q^{LP} \setminus S_i') \right)$$

Is it possible to efficiently separate cuts for $\hat{P}$?

- I.e., simultaneously consider all splits in set I, and perform the projection

Valid inequality characterization

Theorem

The inequality

$$cx + dz \geq f$$

is valid for the set $\hat{P}$ *if and only if there exists a solution to :*

$$c = \sum_{i \in I} c^i, \quad d = \sum_{i \in I} d^i, \quad 0 = \sum_{i \in I} h^i, \quad f = \sum_{i \in I} f^i$$

$$\left.\begin{array}{rclrcl} c^i & \geq & \lambda_1^i H - \mu_1^i \pi^i & c^i & \geq & \lambda_2^i H + \mu_2^i \pi^i \\ d^i & \geq & \lambda_1^i K & d^i & \geq & \lambda_2^i K \\ h^i & \geq & \lambda_1^i L & h^i & \geq & \lambda_2^i L \\ f^i & \leq & \lambda_1^i g - \mu_1^i \gamma^i & f^i & \leq & \lambda_2^i g + \mu_2^i (\gamma^i + 1) \\ \lambda_1^i, \lambda_2^i & \geq & 0 & \mu_1^i, \mu_2^i & \geq & 0, \qquad c^i, d^i, f^i \text{ free} \end{array}\right\} \forall i \in I$$

- Yields cut-generating LP that is $2|I|$ times larger than original formulation
- For one split, overcomes limitation of working with Benders-cut based approximation

Is the difference significant?

Two experiments:

- Compare lift-and-project closure (simple splits) using cut-and-project and project-and-cut
- Investigate impact of using split cuts in the cut-and-project approach to solve to optimality

Test problems:

- CAP: Stochastic version of capacitated warehouse location problem
- SNIP: Stochastic network interdiction problem from Pan and Morton (2008)

When solving to optimality, we use the Rank-1 GMI heuristic separation routine of Dash and Goycoolea (2010)

- Not restricted to simple disjunctions

Closure results: CAP (250 scenarios)

Integrality gap

CAP#	Benders	P&C-S	C&P-S	C&P-GMI
101	17.5	13.6	0.00	0.03
102	21.9	17.5	0.00	0.11
111	9.0	8.1	0.05	0.04
112	9.0	8.0	0.20	0.33
121	15.9	13.7	0.03	0.18
122	19.0	16.5	0.60	0.78
131	20.6	17.4	0.01	0.02
132	25.2	21.7	0.94	0.22

Benders: LP relaxation with no split cuts
P&C-S: Lift-and-project closure in projected space (project-and-cut)
C&P-S: Lift-and-project closure in extended space (cut-and-project)
C&P-GMI: Cut-and-project using GMI heuristic

Closure results: SNIP

Integrality gap

Budget	Benders	P&C-S	C&P-GMI
30	22.3	5.7	8.0
40	25.3	8.4	11.0
50	26.9	9.8	12.0
60	26.9	9.6	11.4
70	29.0	16.5	12.7
80	28.9	12.9	12.1
90	31.0	15.8	15.0

- P&C-S: Stopped early after 2-3 days (true gap may be smaller)
- C&P-S: Too slow
- C&P-GMI: Obtained in minutes

Branch-and-cut results: CAP

		Avg Time (# unsolved)		Avg Gap (%)		
K	CAP #	Ext	+GMI	Ext	Ben	+GMI
250	101-104	258.1	56.2	0.0	14.1	0
	111-114	2359.0	644.0	0.0	6.9	0
	121-124	3252.3 (2)	1223.4	0.4	15.6	0
	131-134	4150.8 (1)	294.8	0.2	22.2	0
500	101-104	1170.5	113.4	0.0	15.7	0
	111-114	10787.1 (3)	1994.2 (1)	2.0	7.5	*0
	121-124	10935.4 (3)	2420.0	3.1	15.9	0
	131-134	9512.0 (3)	737.3	1.4	23.1	0

- C++ , Cplex 12.4 , 2.7 GHz Intel Core i7 CPU , 16GB RAM (used only 1 thread)
- Pure Benders did not solve any in the four hour time limit

Branch-and-cut results: SNIP

		Avg Time (# unsolved)		Avg Gap (%)		
no	B	Ben	+GMI	Ext	Ben	+GMI
3	30	1139	426	18.0	0.0	0.0
	50	11838 (3)	2158	26.1	2.0	0.0
	70	14400 (5)	8242 (1)	27.1	3.3	0.1
	90	14400 (5)	13425 (3)	30.7	6.4	1.5
4	30	695	412	22.4	0.0	0.0
	50	4966 (1)	1107	29.1	0.9	0.0
	70	9554 (2)	1597	39.0	0.3	0.0
	90	9641 (3)	1475	58.7	0.8	0.0

- No instances were solved in time limit using Extensive form
- Ben+GMI is comparable to results obtained in Pan and Morton using problem-specific cuts

Summary

Two options for using integrality information to enhance Benders decomposition

- Project-and-cut: Add cuts based on master formulation directly
- Cut-and-project: Add cuts to subproblems, then project

Given a cut class (e.g., splits cuts), adding cuts in extended space can yield stronger relaxation

Cut-and-project with GMI heuristic yields very competitive algorithm for stochastic IP with first-stage integer variables

Questions for general MIP

Benders master formulation	$\Leftrightarrow$	"Natural" MIP formulation
Extensive Form	$\Leftrightarrow$	Lifted formulation

- Can we introduce lifted formulations for the explicit purpose of improving the power of known classes of cuts?
- Is it possible that a problem with exponential extension complexity could have a polynomial lifted formulation such that the *closure* of that formulation is an extended formulation? (e.g., like matching)