Stochastic Integer Programming

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

XIV International Conference on Stochastic Programming
Buzios, Brazil
June 25, 2016
Outline

1. Introduction
2. Integer Programming Background
3. Cut Based Methods (Branch-and-Cut/Benders)
4. Lagrangian Relaxation Based Methods (Dual Decomposition)
5. Odds and Ends
**Stochastic Mixed Integer Program (SMIP)**

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

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

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

- \( x \): first-stage decision variables
- \( y \): second-stage decision variables
- Sometimes assume \( n_1, p_1, n_2, \) or \( p_2 \) are zero
Example applications

Stochastic service system design (facility location)
- Random customer demands
- First-stage decisions: which service centers to open (binary)
- Second-stage decisions: which customers served by which servers (binary or continuous)

Stochastic vehicle routing (a priori routing)
- Random customer demands
- First-stage decisions: planned vehicle routes (binary)
- Second-stage decisions: recourse actions when capacity violated (binary or continuous)

Stochastic unit commitment
- Random electricity loads and wind/solar production
- First-stage decisions: units to commit and when (binary)
- Second-stage decisions: production amounts, line switching (binary or continuous)
We focus only on the two-stage SMIP problem

Many interesting topics we won’t cover, e.g.,

- Multistage
- Risk-averse
- Chance constraints
Finite scenario model

We assume the random data $\xi$ is represented by a finite set of scenarios:

- $(q^s, h^s, T^s, W^s), s = 1, \ldots, S$
- Scenario $s$ occurs with probability $p_s$
- $S$ should not be too large!

Sample Average Approximation

[McK et al., 1999, Kleywegt et al., 2001, Ahmed and Shapiro, 2002]

- Medium accuracy approximation can be obtained by Monte Carlo sampling
- Required sample size grows linearly with number of first-stage variables
- Replicating SAA problems yields statistical estimates of optimality
- Key challenge: Solving the SAA problem for “large enough” $S$
Example: Stochastic facility location

Problem setup

A firm is deciding which facilities to open to serve customers with random demands. Goal is to minimize total (expected) cost.

Notation:

- $I$: Set of possible facilities to open
- $J$: Set of customers
- $f_i$: Fixed cost for opening facility $i$
- $C_i$: Capacity of facility $i$
- $c_{ij}$: Unit cost for serving customer $j$ demand at facility $i$
- $q_j$: Penalty per unit of unmet demand of customer $j$
- $p_s$: Probability of scenario $s$, $s = 1, \ldots, S$
- $d^s_j$: Demand of customer demand $j$ in scenario $s$
Example: Stochastic facility location

First-stage integer variables: \( x_i = 1 \) if facility \( i \) is open, 0 otherwise

\[
\min_x \sum_{i \in I} f_i x_i + \sum_{s \in S} p_s Q_s(x) \\
\text{s.t. } x_i \in \{0, 1\}, \quad i \in I
\]

where

\[
Q_s(x) = \min_{x,y} \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} + \sum_{j \in J} q_j z_j \\
\text{s.t. } \sum_{i \in I} y_{ij} + z_j \geq d_j^s, \quad j \in J \\
\sum_{j \in J} y_{ij} \leq C_i x_i, \quad i \in I \\
y_{ij} \geq 0, z_j \geq 0, \quad i \in I, j \in J
\]
First option for solving a SMIP with finite scenarios

**Extensive form (deterministic equivalent) of an SMIP**

\[
\begin{align*}
\min & \quad c^T x + \sum_{s=1}^{S} p_s q_s^T y_s \\
\text{s.t.} & \quad Ax \geq b \\
& \quad T_s x + W_s y_s = h_s \\
& \quad x \in \mathbb{R}_{+}^{n_1} \times \mathbb{Z}_{+}^{p_1} \\
& \quad y_s \in \mathbb{R}_{+}^{n_2} \times \mathbb{Z}_{+}^{p_2}, \quad s = 1, \ldots, S
\end{align*}
\]
Example: Stochastic facility location

\[
\begin{align*}
\min_{x,y,z} & \quad \sum_{i \in I} f_i x_i + \sum_{s \in S} p_s \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} s + \sum_{s \in S} p_s \sum_{j \in J} q_j z_{js} \\
\text{s.t.} & \quad \sum_{i \in I} y_{ij} s + z_{js} \geq d_j^s, \quad j \in J, s \in S \\
& \quad \sum_{j \in J} y_{ij} s \leq C_i x_i, \quad i \in I, s \in S \\
& \quad x_i \in \{0, 1\}, \quad i \in I \\
& \quad z_{js} \geq 0, y_{ij} \geq 0, \quad i \in I, j \in J, s \in S
\end{align*}
\]
What makes SMIP hard?

Stochastic integer programming combines challenges from Integer programming and Stochastic Programming

**Integer programming challenges**
- Huge number of discrete options
- Weak relaxations can lead to huge enumeration trees

**Stochastic programming challenges**
- Evaluating expectation
- Huge size even with finite scenario approximation

**Key questions**
- Approximate expected value (SAA)
- Obtain strong relaxations
- Decompose large problem into smaller subproblems
- Preserve relaxation strength when doing decomposition
- Converge to optimal solution
Basic idea behind most algorithms for solving integer programming problems

- Solve a *relaxation* of the problem
  - Some constraints are ignored or replaced with less stringent constraints
- Gives a lower *bound* on the true optimal value
- If the relaxation solution is feasible, it is optimal
- Otherwise, divide the feasible region (*branch*) and repeat
Linear programming relaxation

Mixed-integer program:

\[ z_{IP} = \min \, c^\top x \]
\[ Ax \geq b \]
\[ x \in \mathbb{R}_+^n \times \mathbb{Z}_+^p \]

Linear programming relaxation:

\[ z_{LP} = \min \, c^\top x \]
\[ Ax \geq b \]
\[ x \geq 0 \]

Simple observation

\[ z_{LP} \leq z_{IP} \]
Branching: The “divide” in “Divide-and-conquer”

Generic optimization problem:

$$z^* = \min \{ c^T x : x \in S \}$$

Consider subsets $S_1, \ldots, S_k$ of $S$ which cover $S$: $S = \bigcup_i S_i$. Then

$$\min \{ c^T x : x \in S \} = \min_{1 \leq i \leq k} \left\{ \min \{ c^T x : x \in S_i \} \right\}$$

In other words, we can optimize over each subset separately.

- Usually want $S_i$ sets to be disjoint ($S_i \cap S_j = \emptyset$ for all $i \neq j$)

Dividing the original problem into subproblems is called branching
Bounding: The “conquer” in “Divide-and-conquer”

Any feasible solution to the problem provides an upper bound $U$ on the optimal solution value. ($\hat{x} \in S \Rightarrow z^* \leq c^\top \hat{x}$).

- We can use heuristics to find a feasible solution $\hat{x}$

After branching, for each subproblem $i$ we solve a relaxation yielding a lower bound $\ell(S_i)$ on the optimal solution value for the subproblem.

- Overall Bound: $L = \min_i \ell(S_i)$

**Key:** If $\ell(S_i) \geq U$, then we don’t need to consider subproblem $i$.

In deterministic MIP, we usually get the lower bound by solving the LP relaxation, but there are other ways too.
Let $z_{IP}$ be the optimal value of the MIP

Solve the relaxation of the original problem:

1. If unbounded $\Rightarrow$ the MIP is unbounded or infeasible.
2. If infeasible $\Rightarrow$ MIP is infeasible.
3. If we obtain a feasible solution for the MIP $\Rightarrow$ it is an optimal solution to MIP. ($L = z_{IP} = U$)
4. Else $\Rightarrow$ Lower Bound. ($L = z_{Rel}$).

In the first three cases, we are finished.

In the final case, we must **branch** and recursively solve the resulting subproblems.
Terminology

- Subproblems (nodes) form a search tree
- Eliminating a problem from further consideration is called pruning
- The act of bounding and then branching is called processing
- A subproblem that has not yet been processed is called a candidate
- The set of candidates is the candidate list
Branch and bound algorithm

1. Derive an bound $U$ using a heuristic method (if possible).
2. Put the original problem on the candidate list.
3. Select a problem $S$ from the candidate list and solve the relaxation to obtain the bound $\ell(S)$
   - Relaxation infeasible $\Rightarrow$ node can be pruned.
   - $\ell(S) < U$ and the solution is feasible for the MIP $\Rightarrow$ set $U \leftarrow \ell(S)$.
   - $\ell(S) \geq U$ $\Rightarrow$ node can be pruned.
   - Otherwise, branch. Add the new subproblems to the list.
4. If the candidate list is nonempty, go to Step 3. Otherwise, the algorithm is completed.
How long does branch-and-bound take?

Simple approximation:

Total time = (Time to process a node) × (Number of nodes)

Both can be very important:

- For **very** large instances (as in stochastic programming), solving a single relaxation can be too time-consuming
- Number of nodes can grow exponentially in number of decision variables if do not prune often enough

**Keys to success**

- Solve relaxations fast (enough)
- Obtain **strong relaxations** so that can prune high in tree

Also important (but less so):

- Make good branching decisions (limits size of tree)
- Obtain good feasible solutions early (e.g., good heuristics)
Formulations in MIP

Integer programs can often be formulated in multiple ways

- E.g., facility location problem
  
  \( x_i = 1 \) if facility \( i \) is open, \( y_{ij} = \) customer \( j \) demand served from facility \( i \)

- Formulation we used earlier:

  \[
  \sum_{j \in J} y_{ij} \leq C_i x_i, \quad \forall i \in I
  \]

- Redundant constraints:

  \[
  y_{ij} \leq \min\{d_j^s, C_i\} x_i, \quad \forall i \in I, j \in J
  \]

- Set of integer feasible points satisfying these are the same

- But many fractional points that satisfy original formulation do not satisfy the redundant constraints
Given two formulations, which is better?

- **Option 1**: Fewer constraints ⇒ Faster LP relaxation solution

\[
\sum_{j \in J} y_{ij} \leq C_i x_i, \quad \forall i \in I
\]

- **Option 2**: Better LP relaxations ⇒ Prune more often

\[
\sum_{j \in J} y_{ij} \leq C_i x_i, \quad \forall i \in I, \quad y_{ij} \leq \min\{d_j^s, C_i\} x_i, \quad \forall i \in I, j \in J
\]

Using the formulation with **better bound** is almost always (much) better.

- May need to use specialized techniques to solve larger relaxation
Example: Stochastic facility location revisited

\[
\begin{align*}
\min_{x, y, z} & \quad \sum_{i \in I} f_i x_i + \sum_{s \in S} p_s \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ijs} + \sum_{s \in S} \sum_{j \in J} p_s q_j z_{js} \\
\text{s.t.} & \quad \sum_{i \in I} y_{ijs} + z_{js} \geq d^s_j, \quad j \in J, s \in S \\
& \quad \sum_{j \in J} y_{ijs} \leq C_i x_i, \quad i \in I, s \in S \\
& \quad y_{ijs} \leq \min\{d^s_j, C_i\} x_i, \quad i \in I, j \in J, s \in S \\
& \quad x_i \in \{0, 1\}, \quad i \in I \\
& \quad z_{js} \geq 0, y_{ijs} \geq 0, \quad i \in I, j \in J, s \in S
\end{align*}
\]
Let $X = \{ x \in \mathbb{R}^n_+ : Ax \leq b, x_j \in \mathbb{Z}, j \in J \}$

**Definition**

An inequality $\pi x \leq \pi_0$ is a **valid inequality** for $X$ if $\pi x \leq \pi_0$ for all $x \in X$.

$(\pi \in \mathbb{R}^n, \pi_0 \in \mathbb{R})$

- Valid inequalities are also called “cutting planes” or “cuts”
- Goal of adding valid inequalities to a formulation: improve relaxation bound ⇒ explore fewer branch-and-bound nodes

**Key questions**

- How to find valid inequalities?
- How to use them in a branch-and-bound algorithm?
Using valid inequalities

1. Add them to the initial formulation
   - Creates a formulation with better LP relaxation
   - Feasible only when you have a “small” set of valid inequalities
   - Easy to implement

2. Add them only as needed to cut off fractional solutions
   - Solve LP relaxation, cut off solution with valid inequalities, repeat
   - **Cut-and-branch**: Do this *only* with the initial LP relaxation (root node)
   - **Branch-and-cut**: Do this at all nodes in the branch-and-bound tree

Trade-off with increasing effort generating cuts

+ Fewer nodes from better bounds
- More time finding cuts and solving LP
Branch-and-cut

At each node in branch-and-bound tree

1. Solve current LP relaxation $\Rightarrow \hat{x}$
2. Attempt to generate valid inequalities that cut off $\hat{x}$
3. If cuts found, add to LP relaxation and go to step 1

Why branch-and-cut?

- Reduce number of nodes to explore with improved relaxation bounds
- Add inequalities required to define feasible region

This approach is the heart of all modern MIP solvers
Deriving valid inequalities

General purpose valid inequalities

- Assume only that you have a (mixed or pure) integer set described with inequalities

\[ X = \{ x \in \mathbb{R}^n_+ : Ax \leq b, x_j \in \mathbb{Z}, j \in J \} \]

- Critical to success of deterministic MIP solvers

Structure-specific valid inequalities

- Rely on particular structure that appears in a problem
- Combinatorial optimization problems: matching, traveling salesman problem, set packing, ...
- Knapsack constraints: \( \{ x \in \mathbb{Z}^n_+ : ax \leq b \} \)
- Flow balance with variable upper bounds, etc.
Let \( P = \{ x \in \mathbb{R}^n_+ : Ax \leq b \} \) and 
\[ X = \{ x \in P : x_j \in \mathbb{Z}, \text{ for } j \in J \} \]

Let \( \pi \in \mathbb{Z}^n \) be such that \( \pi_j = 0 \) for all \( j \notin J \), and \( \pi_0 \in \mathbb{Z} \)

\( \pi x \) is integer for all \( x \in X \)

\( x \in X \Rightarrow \) either \( \pi x \leq \pi_0 \) or \( \pi x \geq \pi_0 + 1 \)

So \( X \subset P_0 \cup P_1 \) where

\[ P_0 = \{ x \in P : \pi x \leq \pi_0 \} \]
\[ P_1 = \{ x \in P : \pi x \geq \pi_0 + 1 \} \]

An inequality valid for \( P_0 \cup P_1 \) is called a split cut
Finding violated split cuts

**Separation problem**

Given an LP relaxation solution \( \hat{x} \), find a valid inequality that “cuts off” \( \hat{x} \), or prove none exists.

When the split disjunction \( \pi x \leq \pi_0, \pi x \geq \pi_0 + 1 \) is fixed

- A split cut, if one exists, can be obtained by solving a linear program

How to choose the split disjunction?

- Difficult problem in general
- Lift-and-project cuts: Restrict to \( x_j \leq 0 \) and \( x_j \geq 1 \) for binary variables \( x_j \)
- Many other heuristics

Split cuts closely related to Gomory mixed-integer cuts and mixed-integer rounding cuts
LP relaxation of SMIP

Stochastic MIP

\[
\begin{align*}
\min & \quad c^T x + \sum_{s=1}^{S} p_s \theta_s \\
\text{s.t.} & \quad A x \geq b \\
& \quad \theta_s \geq Q_s(x), \quad s = 1, \ldots, S \\
& \quad x \in \mathbb{R}_{+}^{n_1} \times \mathbb{Z}_{+}^{p_1}
\end{align*}
\]

where for \( s = 1, \ldots, S \)

\[
Q_s(x) = \min q_s^T y \\
\text{s.t.} \quad W_s y = h_s - T_s x \\
& \quad y \in \mathbb{R}_{+}^{n_2} \times \mathbb{Z}_{+}^{p_2}
\]

LP Relaxation

\[
\begin{align*}
\min & \quad c^T x + \sum_{s=1}^{S} p_s \theta_s \\
\text{s.t.} & \quad A x \geq b \\
& \quad \theta_s \geq Q_s^{LP}(x), \quad s = 1, \ldots, S \\
& \quad x \in \mathbb{R}_{+}^{n_1} \times \mathbb{R}_{+}^{p_1}
\end{align*}
\]

where for \( s = 1, \ldots, S \)

\[
Q_s^{LP}(x) = \min q_s^T y \\
\text{s.t.} \quad W_s y = h_s - T_s x \\
& \quad y \in \mathbb{R}_{+}^{n_2} \times \mathbb{R}_{+}^{p_2}
\]
Benders Decomposition (L-Shaped Method)

\[(\text{MP})^{LP}_{t}: \min_{\theta,x} \ c^T x + \sum_{s=1}^{S} p_s \theta_s \]
\[\text{s.t. } Ax \geq b, x \in \mathbb{R}_{+}^{n_1} \times \mathbb{R}_{+}^{p_1} \]
\[e \theta_s \geq d_{s,t} + B_{s,t} x, \ s = 1, \ldots, S,\]

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

- Converges after finitely many iterations
- Improvements: trust region/level stabilization, modified subproblem
- Variant: Single cut aggregated over scenarios
Simplest case: Continuous recourse

\[
\min \ c^T x + \sum_{s=1}^{S} p_s \theta_s \\
\text{s.t. } Ax \geq b \\
\theta_s \geq Q_s(x), \quad s = 1, \ldots, S \\
x \in \mathbb{R}_{+}^{n_1} \times \mathbb{Z}_{+}^{p_1}
\]

where for \( s = 1, \ldots, S \)

\[
Q_s(x) = \min \ q_s^T y \\
\text{s.t. } W_s y = h_s - T_s x \\
y \in \mathbb{R}_{+}^{n_2}
\]
Method 1: Basic Benders decomposition

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

\[\text{s.t. } Ax \geq b, x \in \mathbb{R}^{n_1}_{+} \times \mathbb{Z}^{p_1}_{+} \]
\[e \theta_s \geq d_{s,t} + B_{s,t} x, \quad s = 1, \ldots, S, \]

\[(SP)^s : Q^{LP}_{s}(\hat{x}) := \min_{y_s} q_s^T y_s \]

\[\text{s.t. } W_s y_s \geq h_s - T_s \hat{x} \]
\[y \in \mathbb{R}^{n_2}_{+} \]

- Converges after finitely many iterations
- Master problem is a mixed-integer program
Example: Facility location

Example data:

- Three possible facilities and four customers
- Fixed costs: $f = [120, 100, 90]$  
- Capacity: $C = [26, 25, 18]$  
- Two equally likely scenarios: $d^1 = [12, 8, 6, 11], \quad d^2 = [8, 11, 7, 6]$  
- Penalty for unmet demand: $q_j = 20$

---

Iteration 1: Master problem (no $\theta$ variable yet)

$$\min 120x_1 + 100x_2 + 90x_3$$

s.t. $x_i \in \{0, 1\}, \quad i = 1, 2, 3$

Optimal solution: $\hat{x} = (0, 0, 0)$

Optimal value (lower bound on SMIP): 0
Example: Iteration 1

Subproblems with $\hat{x} = (0, 0, 0)$:

\[
\begin{align*}
\min & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30 z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^1, \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 0 \\
& \quad y_{ij} \geq 0, z_j \geq 0
\end{align*}
\]

Yields Benders cut:

$\theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3$

Upper bound: $\sum_i f_i \hat{x}_i + \sum_s p_s Q_s(\hat{x}) = 0 + 1/2(1140 + 990) = 1065$

\[
\begin{align*}
\min & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30 z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^2, \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 0 \\
& \quad y_{ij} \geq 0, z_j \geq 0
\end{align*}
\]

Yields Benders cut:

$\theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3$
Example: Iteration 2

Updated master problem

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad x_i \in \{0, 1\}, \, i = 1, 2, 3
\end{align*}
\]

Optimal solution: \(\hat{x} = (0, 1, 1), \, \hat{\theta} = (0, 0)\)
Optimal value (lower bound on SMIP): 190
Subproblems with $\hat{x} = (0, 1, 1)$:

$$\begin{align*}
\text{min} & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30 z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^1, \quad \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 1 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 1 \\
& \quad y_{ij} \geq 0, \quad z_j \geq 0
\end{align*}$$

Yields Benders cut:

$$\theta_1 \geq 200 - 130 x_1 - 18 x_3$$

Upper bound: $\sum_i f_i \hat{x}_i + \sum_s p_s Q_s(\hat{x}) = 190 + 1/2(182 + 142) = 352$

Yields Benders cut:

$$\theta_2 \geq 142 - 104 x_1$$
Example: Iteration 3

Updated master problem

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + \frac{1}{2}(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_1 \geq 200 - 130x_1 - 18x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 142 - 104x_1 \\
& \quad x_i \in \{0, 1\}, \ i = 1, 2, 3
\end{align*}
\]

Optimal solution: \( \hat{x} = (1, 0, 1), \ \hat{\theta} = (52, 38) \)
Optimal value (lower bound on SMIP): 255
Example: Iteration 3

Subproblems with \( \hat{x} = (1, 0, 1) \):

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30 z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^1, \quad \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 1 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 1 \\
& \quad y_{ij} \geq 0, \ z_j \geq 0
\end{align*}
\]

Yields Benders cut:

\[
\theta_1 \geq 237 - 26x_1 - 125x_2
\]

Upper bound: \( \sum_i f_i \hat{x}_i + \sum_s p_s Q_s(\hat{x}) = 210 + 1/2(211 + 182) = 406.5 \)

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30 z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^2, \quad \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 1 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 1 \\
& \quad y_{ij} \geq 0, \ z_j \geq 0
\end{align*}
\]

Yields Benders cut:

\[
\theta_2 \geq 208 - 26x_1 - 125x_2
\]
Example: Iteration 5 (skipped one!)

Updated master problem

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_1 \geq 200 - 130x_1 - 18x_3 \\
& \quad \theta_1 \geq 237 - 26x_1 - 125x_2 \\
& \quad \theta_1 \geq 141 - 36x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 142 - 104x_1 \\
& \quad \theta_2 \geq 208 - 26x_1 - 125x_2 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad x_i \in \{0, 1\}, i = 1, 2, 3
\end{align*}
\]

Optimal solution: \( \hat{x} = (0, 1, 1), \hat{\theta} = (182, 142) \)

Optimal value (lower bound on SMIP): 352

- Matches upper bound from Iteration 2 ⇒ Optimal
- Subproblems yield no violated cuts
Recap: Basic Benders algorithm

Basic Benders for SMIP with continuous recourse

Repeat until no cuts found:

1. Solve Benders master mixed-integer program
2. Solve scenario LP subproblems, generate cuts, and add to Benders master.

Limitation

Solving the MIP in step 1 can become very time-consuming

- Tends to become more difficult as more cuts are added
- Unlike an LP, MIP master cannot be effectively warm-started ⇒ Significant “redundant” work

Alternative

Add Benders cuts as needed during a single branch-and-cut process.
Method 2: Branch-and-cut with Benders cuts

Initialize Benders master problem with Benders cuts
- E.g., solve the LP relaxation via Benders and keep cuts

Begin branch-and-cut algorithm. At each node in the search tree:
- Solve LP relaxation $\Rightarrow (\hat{x}, \hat{\theta})$
- If LP bound exceeds known incumbent, prune.
- If $\hat{x}$ is integer feasible: $(\hat{x}, \hat{\theta})$ might not be feasible!
  - Solve scenario subproblems to generate Benders cuts
  - If $(\hat{\theta}_s, \hat{x})$ violates any Benders cut, add cut to LP relaxation and re-solve.
- If $\hat{x}$ not integer feasible:
  - Optional: Solve scenario subproblems and add Benders cuts if violated
  - Else: Branch to create new nodes

Cuts added when $\hat{x}$ is integer feasible are known as lazy cuts in MIP solvers (add via cut callback routine).
Example revisited: Initialization

First, solve the LP relaxation via Benders

Iteration 1: Master linear problem (no $\theta$ variable yet)

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 \\
\text{s.t.} & \quad 0 \leq x_i \leq 1, \quad i = 1, 2, 3
\end{align*}
\]

Optimal solution: $\hat{x} = (0, 0, 0)$

Optimal value (lower bound on SMIP): 0
Example: Iteration 1

Subproblems with $\hat{x} = (0, 0, 0)$:

<table>
<thead>
<tr>
<th>Min</th>
<th>$\sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30 z_j$</th>
<th>Min</th>
<th>$\sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30 z_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S.t.</td>
<td>$\sum_{i=1}^{4} y_{ij} + z_j = d^1_j$, $\forall j$</td>
<td>S.t.</td>
<td>$\sum_{i=1}^{4} y_{ij} + z_j = d^2_j$, $\forall j$</td>
</tr>
<tr>
<td></td>
<td>$\sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0$</td>
<td></td>
<td>$\sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0$</td>
</tr>
<tr>
<td></td>
<td>$\sum_{j=1}^{4} y_{ij} \leq 25 \cdot 0$</td>
<td></td>
<td>$\sum_{j=1}^{4} y_{ij} \leq 25 \cdot 0$</td>
</tr>
<tr>
<td></td>
<td>$\sum_{j=1}^{4} y_{ij} \leq 18 \cdot 0$</td>
<td></td>
<td>$\sum_{j=1}^{4} y_{ij} \leq 18 \cdot 0$</td>
</tr>
<tr>
<td></td>
<td>$y_{ij} \geq 0$, $z_j \geq 0$</td>
<td></td>
<td>$y_{ij} \geq 0$, $z_j \geq 0$</td>
</tr>
</tbody>
</table>

Yields Benders cut:

$\theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3$

Upper bound (because $\hat{x}$ is integer feasible!):

$\sum_i f_i \hat{x}_i + \sum_s p_s Q_s(\hat{x}) = 0 + 1/2(1140 + 990) = 1065$
Example: Iteration 2

Updated master linear program

$$\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad 0 \leq x_i \leq 1, i = 1, 2, 3
\end{align*}$$

Optimal solution: $\hat{x} = (0.639, 1, 0), \hat{\theta} = (0, 0)$
Optimal value (lower bound on SMIP): 176.6
Example: Iteration 2

Subproblems with $\hat{x} = (0.639, 1, 0)$:

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30 z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^1, \ \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0.639 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 1 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 0 \\
& \quad y_{ij} \geq 0, z_j \geq 0
\end{align*}
\]

Yields Benders cut:

\[
\theta_1 \geq 179 - 52x_1 - 72x_3
\]

No upper bound because $\hat{x}$ is not integer feasible

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30 z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^2, \ \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0.639 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 1 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 0 \\
& \quad y_{ij} \geq 0, z_j \geq 0
\end{align*}
\]

Yields Benders cut:

\[
\theta_2 \geq 124 - 36x_3
\]
Example: Iteration 6 (skipped several steps)

Updated master linear problem

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_1 \geq 179 - 52x_1 - 72x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad 0 \leq x_i \leq 1, i = 1, 2, 3
\end{align*}
\]

Optimal solution: \(\hat{x} = (0.5, 0.76, 0.33), \hat{\theta} = (129, 112)\)

Optimal value (lower bound on SMIP): 286.5

- Subproblems yield no more violated Benders cuts
- Solution is optimal to the LP relaxation
Example: Branch-and-cut phase

Current master problem

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_1 \geq 179 - 52x_1 - 72x_3 \\
& \quad \theta_1 \geq 179 - 52x_1 - 72x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad 0 \leq x_i \leq 1, i = 1, 2, 3 \\
x_i \in \{0, 1\}, & \quad i = 1, 2, 3
\end{align*}
\]

Load this (partial) formulation to the MIP solver and start solution process

- Let’s first suppose MIP solver adds no cuts of its own
- What will it do?
Example: Branch-and-cut phase

Initial master linear program relaxation

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + \frac{1}{2}(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_1 \geq 179 - 52x_1 - 72x_3 \\
& \quad \hdots \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad \hdots \\
& \quad 0 \leq x_i \leq 1, \ i = 1, 2, 3
\end{align*}
\]

Optimal solution: \( \hat{x} = (0.5, 0.76, 0.33), \hat{\theta} = (129, 112) \)

Branch!

Let’s branch on \( x_1 \):
Example: First two nodes

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_1 \geq 179 - 52x_1 - 72x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad 0 \leq x_i \leq 1, \quad i = 1, 2, 3
\end{align*}
\]

Node 1: Fix \( x_1 = 0 \)
Optimal solution: \( \hat{x} = (0, 1, 0.66) \), \( \hat{z} = 326.3 \)

Node 2: Fix \( x_1 = 1 \)
Optimal solution: \( \hat{x} = (1, 0.46, 0) \), \( \hat{z} = 304.9 \)

Neither can be pruned. Neither yields an upper bound

⇒ Further subdivide each. Start with node 2.
Example: More nodes

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_1 \geq 179 - 52x_1 - 72x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad 0 \leq x_i \leq 1, i = 1, 2, 3
\end{align*}
\]

Node 3: Fix \( x_1 = 1, x_2 = 0 \)
Optimal solution: \( \hat{x} = (1, 0, 0.42), \hat{z} = 355.2 \)

Node 4: Fix \( x_1 = 1, x_2 = 1 \)
Optimal solution: \( \hat{x} = (1, 1, 0), \hat{z} = 345.5 \)

Node 4 yields integer feasible solution!

- But \( (\hat{x}, \hat{\theta}) \) is not necessarily feasible! (if \( \hat{\theta}_s < Q_s(\hat{x}) \) for some \( s \))
- We MUST check if there are any violated Benders cuts
Scenario subproblems at Node 4

Subproblems with $\hat{x} = (1, 1, 0)$:

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^1, \quad \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 1 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 1 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 0 \\
& \quad y_{ij} \geq 0, z_j \geq 0
\end{align*}
\]

Yields violated Benders cut:

\[\theta_1 \geq 141 - 36x_3\]

Upper bound (because $\hat{x}$ is integer feasible!):

\[\sum_i f_i \hat{x}_i + \sum_s p_s Q_s(\hat{x}) = 220 + 1/2(141 + 124) = 352.5\]
Example: Updated master LP relaxation

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_1 \geq 141 - 36x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad 0 \leq x_i \leq 1, \ i = 1, 2, 3
\end{align*}
\]

Node 3: Fix \( x_1 = 1, x_2 = 0 \)
Optimal solution: \( \hat{x} = (1, 0, 0.42), \hat{z} = 355.2 \)

Node 4: Fix \( x_1 = 1, x_2 = 1 \)
New optimal solution: \( \hat{x} = (1, 1, 0), \hat{z} = 352.5 \)

Re-solve master problem at Node 4: Again integer feasible
- No more violated Benders cut \( \Rightarrow \) 352.5 is a valid upper bound
- Both Nodes 3 and 4 can be pruned!
Example: Back to node 1

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \ldots \\
& \quad \theta_1 \geq 141 - 36x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad \ldots \\
& \quad 0 \leq x_i \leq 1, i = 1, 2, 3
\end{align*}
\]

Node 1: Fix \( x_1 = 0 \)

Optimal solution: \( \hat{x} = (0, 1, 0.66) \), \( \hat{z} = 326.3 \)

Subdivide again: \( x_3 = 0 \) or \( x_3 = 1 \)
Example: More nodes

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_1 \geq 141 - 36x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad 0 \leq x_i \leq 1, \ i = 1, 2, 3
\end{align*}
\]

Node 5: Fix \(x_1 = 0, x_3 = 0\)

Optimal solution: \(\hat{x} = (0, 1, 0), \hat{z} = 507\)

Node 6: Fix \(x_1 = 0, x_3 = 1\)

Optimal solution: \(\hat{x} = (0, 0.75, 1), \hat{z} = 326.3\)

- Node 5 can be pruned (bound is worse than upper bound of 352.5)
- Node 6 must be subdivided again: \(x_2 = 0\) or \(x_3 = 1\)
Example: More nodes

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + 1/2(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \ldots \\
& \quad \theta_1 \geq 141 - 36x_3 \\
& \quad \theta_2 \geq 990 - 728x_1 - 675x_2 - 468x_3 \\
& \quad \theta_2 \geq 124 - 36x_3 \\
& \quad \ldots \\
& \quad 0 \leq x_i \leq 1, i = 1, 2, 3
\end{align*}
\]

Node 7: Fix \( x_1 = 0, x_3 = 1, x_2 = 0 \)
Optimal solution: \( \hat{x} = (0, 0, 1) \), \( \hat{z} = 687 \)

Node 8: Fix \( x_1 = 0, x_3 = 1, x_2 = 1 \)
Optimal solution: \( \hat{x} = (0, 1, 1) \), \( \hat{z} = 350.5 \)

- Node 7 can be pruned (bound is worse than upper bound of 352.5)
- Node 8 is integer feasible: Must check for Benders cuts
Scenario subproblems at Node 8

Subproblems with $\hat{x} = (0, 1, 1)$:

\[
\begin{align*}
\min \quad & \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij}y_{ij} + \sum_{j=1}^{4} 30z_{j} \\
\text{s.t.} \quad & \sum_{i=1}^{4} y_{ij} + z_{j} = d_{j}^{1}, \quad \forall j \\
& \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0 \\
& \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 1 \\
& \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 1 \\
& y_{ij} \geq 0, \quad z_{j} \geq 0
\end{align*}
\]

Yields Benders cut:

\[
\theta_{1} \geq 200 - 130x_{1} - 18x_{3}
\]

Upper bound (because $\hat{x}$ is integer feasible):

\[
\sum_{i} f_{i}\hat{x}_{i} + \sum_{s} p_{s} Q_{s}(\hat{x}) = 190 + 1/2(182 + 142) = 352
\]

After re-solving, node 8 can be pruned $\Rightarrow$ We are done!
That was not very efficient!

What went wrong?
- Poor LP relaxations!

What to do?
- Add Benders cuts at fractional LP solutions
- Use integrality to add stronger cuts (not implied by LP relaxation)

Two options for using integrality to add stronger cuts
- Generate cuts directly in the master problem
- Generate cuts in the subproblems
Master problem cuts

Idea

Derive valid inequalities for the mixed-integer set:

\[ \{(x, \theta) : Ax \geq b, \]
\[ e\theta_s \geq d_{s,t} + B_{s,t}x, \ s = 1, \ldots, S, \]
\[ x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+, \theta \in \mathbb{R}^S \}\]

where the constraints in second row are some Benders cuts

- E.g., split cuts, mixed-integer rounding [Bodur and Luedtke, 2016],
  Gomory mixed-integer cuts, . . . ,
- Ideally, would have all Benders cuts defining \( \{(x, \theta) : \theta_s \geq Q_s(x)\} \) but in general too many to enumerate
Adding cuts in master problem

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

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

\[e \theta_s \geq d_{s,t} + B_{s,t} x, \forall s, \]

\[C_t \theta + D_t x \geq g_t \]

\[(SP)^s : Q_s^{LP}(\hat{x}) := \min_{y_s} q_s^T y_s \]

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

\[y \in \mathbb{R}_{+}^{n_2} \times \mathbb{R}_{+}^{p_2} \]
Master problem cuts: Help the MIP solver help you

Master Problem Cuts

Derive valid inequalities for the mixed-integer set:

\[
\{(x, \theta) : Ax \geq b, \\
e \theta_s \geq d_{s,t} + B_{s,t}x, \ s = 1, \ldots, S, \\
x \in \mathbb{R}_+^{n_1} \times \mathbb{Z}_+^{p_1}, \theta \in \mathbb{R}^S\}
\]

where the constraints in second row are some Benders cuts

- MIP solvers will (try to) do this for you if Benders cuts are given to the solver as part of the formulation!
- Facility location example: Gurobi improves root node relaxation from 286.5 to 326.8 (compared to 352 opt)

MIP solvers do not derive cuts based on cuts you add in a callback
**Master Problem Cuts**

Derive valid inequalities for the mixed-integer set:

\[ \{(x, \theta) : Ax \geq b, \]
\[ e\theta_s \geq d_{s,t} + B_{s,t}x, \quad s = 1, \ldots, S, \]
\[ x \in \mathbb{R}_+^{n_1} \times \mathbb{Z}_+^{p_1}, \quad \theta \in \mathbb{R}^S \}\]

where the constraints in second row are some Benders cuts

---

**Takeaway**

Phase 0 (solve LP relaxation with Benders, include cuts in formulation) can be very important for effective branch-and-cut implementation

- Don’t just add cuts in a callback
Cuts in the subproblems: Help yourself!

**Key Idea**

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

\[ X_s := \{(x, y) : Ax \geq b, \; T_s x + W_s y = h_s \} \]

\[ x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+, \; y \in \mathbb{R}^{n_2}_+ \times \mathbb{Z}^{p_2}_+ \]

- NB: So far, we have only seen a convergent algorithm for the case \( y \) is continuous
- But subproblem approach for generating cuts is valid and useful for \( y \) mixed-integer

Cuts generated for a single scenario \( \Rightarrow \) Can still apply Benders decomposition
Strength of split cuts: Master vs. subproblem

\[ E_s := \{(x, y, \theta) : Ax \geq b, \ T_s x + W_s y = h_s, \ \theta = q_s^\top y \] 
\[ x \in \mathbb{R}^{n_1}_+ \times \mathbb{R}^{p_1}_+, \ y \in \mathbb{R}^{n_2}_+ \times \mathbb{R}^{p_2}_+ \} \]

\[ P_s := \{(x, \theta) : \ \exists y \text{ with } (x, y, \theta) \in Q_s \} \]

- \( E_s \) is the subproblem formulation, \( P_s \) is projection based on all Benders cuts from one scenario

**Theorem [Bodur et al., 2016]**

Relaxation obtained using all split cuts on \( E_s \) is at least as good as that using all split cuts on \( P_s \), and difference can be large.

Adding cuts in subproblems can be much more effective:

- Power of extended variable space
- Can also use second-stage integrality

Also implies that split cuts in master problem can be more effective when using multi-cut \( (\theta_s \geq \ldots) \) vs. single-cut \( (\theta \geq \sum_s \ldots) \)
Cuts in subproblem

\[(\text{MP})_{t}^{LP} : \min_{\theta, x} c^T x + \sum_{s=1}^{S} p_s \theta_s\]

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

\(e\theta_s \geq d_{s,t} + B_{s,t}x, \forall s,\)

\(\theta \in \mathbb{R}^S\)

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

s.t. \(W_s y_s \geq h_s - T_s \hat{x}\)

\(y \in \mathbb{R}_{+}^{n_2} \times \mathbb{R}_{+}^{p_2}\)
Cuts in subproblem

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

s.t. \[Ax \geq b, x \in \mathbb{R}^{n_1}_+ \times \mathbb{R}^{p_1}_+\]
\[e \theta_s \geq d_{s,t} + B_{s,t} x, \ \forall s,\]
\[\theta \in \mathbb{R}^S\]

\[(SP)^s : Q_s(\hat{x}) := \min_{y_s} q_s^T y_s \]

s.t. \[W_s y_s \geq h_s - T_s \hat{x}\]
\[C_s y_s \geq g_s - D_s \hat{x}\]
\[y \in \mathbb{R}^{n_2}_+ \times \mathbb{R}^{p_2}_+\]

Add cuts to \((SP)^s\), even if it’s originally an LP \((p_2 = 0)\)
Cuts in the subproblems: Help yourself!

Question

How to generate valid inequalities for each scenario mixed-integer set?

\[ X_s := \{(x, y) : Ax \geq b, \ T_s x + W_s y = h_s \] 
\[ x \in \mathbb{R}^{n_1}_1 \times \mathbb{Z}^{p_1}_1, y \in \mathbb{R}^{n_2}_1 \times \mathbb{Z}^{p_2}_1 \} \]

Generating such cuts might require expertise in general integer programming cuts

- Split cuts, Gomory mixed-integer cuts, Chvátal-Gomory cuts, ...

But it also might not...

- Use problem-specific cuts, or a better formulation
- E.g., facility location problem
Facility location: Subproblem cuts

Feasible region for a scenario $s$:

$$\{(x, y, z) : \sum_{i \in I} y_{ij} + z_j \geq d^s_j, \quad j \in J \}
\sum_{j \in J} y_{ij} \leq C_i x_i, \quad i \in I
y_{ij} \geq 0, z_j \geq 0, \quad i \in I, j \in J
x_i \in \{0, 1\}, \quad i \in I \}$$

Recall: Valid inequalities

$$y_{ij} \leq \min\{d^s_j, C_i\} x_i, \quad i \in I, j \in J$$

Two options for using them (because there is a “small” number of them)

- Directly add them to the scenario subproblem formulations ✓
- Add them as cuts when solving scenario subproblems
Cut Based Methods (Branch-and-Cut/Benders)

Continuous Recourse

Branch-and-cut again

Initialization phase: Solve new LP relaxation via Benders

Iteration 1: Master linear problem (no $\theta$ variable yet)

\[
\begin{align*}
\min \ & 120x_1 + 100x_2 + 90x_3 \\
\text{s.t.} \ & 0 \leq x_i \leq 1, \ i = 1, 2, 3
\end{align*}
\]

Optimal solution: $\hat{x} = (0, 0, 0)$
Optimal value (lower bound on SMIP): 0
Subproblems with $\hat{x} = (0, 0, 0)$:

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^1, \quad \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 0 \\
& \quad y_{11} \leq 12 \cdot 0 \\
& \quad \ldots \\
& \quad y_{34} \leq 11 \cdot 0 \\
& \quad y_{ij} \geq 0, z_j \geq 0
\end{align*}
\]

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} y_{ij} + \sum_{j=1}^{4} 30z_j \\
\text{s.t.} & \quad \sum_{i=1}^{4} y_{ij} + z_j = d_j^2, \quad \forall j \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 26 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 25 \cdot 0 \\
& \quad \sum_{j=1}^{4} y_{ij} \leq 18 \cdot 0 \\
& \quad y_{11} \leq 8 \cdot 0 \\
& \quad \ldots \\
& \quad y_{34} \leq 6 \cdot 0 \\
& \quad y_{ij} \geq 0, z_j \geq 0
\end{align*}
\]

Each yields a Benders cut which is added to master problem...
Example: Iteration 7 (skipped many steps)

Updated master linear program

\[
\begin{align*}
\text{min} & \quad 120x_1 + 100x_2 + 90x_3 + \frac{1}{2}(\theta_1 + \theta_2) \\
\text{s.t.} & \quad \theta_1 \geq 1140 - 923x_1 - 922x_2 - 864x_3 \\
& \quad \ldots \\
& \quad \theta_2 \geq 990 - 794x_1 - 812x_2 - 758x_3 \\
& \quad \ldots \\
& \quad 0 \leq x_i \leq 1, \ i = 1, 2, 3
\end{align*}
\]

Optimal solution: \( \hat{x} = (0.56, 0.93, 0) \), \( \hat{\theta} = (190.3, 152.4) \)

Recall: Original LP relaxation bound = 286.5

- Bound using this formulation: 332.4 (recall opt = 352)
- After Gurobi master cuts on this: 350.5
Recap: SMIP with continuous recourse

Two general approaches:

1. Benders with MIP master problem
2. Branch-and-cut adding Benders cuts (and others) in tree

Which is better?

1. **Sequence of MIPs**
   - Easy to implement
   - Tends to solve fewer scenario subproblems
   - Master MIP problems may become bottleneck
   - Takes full advantage of MIP solver

2. **Branch-and-cut**
   - More difficult to implement
   - Single tree eliminates redundant work
   - Allows exploiting subproblem cuts, e.g., based on problem structure

My advice: Try simpler option first!
Mixed-integer recourse

What goes wrong with Benders approach with mixed-integer recourse?

**Stochastic MIP**

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

- \( Q_s(x) \): Value function of a mixed-integer program
- Benders cuts (including strengthened with subproblem cuts) still valid
- But master problem constraints \( \theta_s \geq Q_s(x) \) cannot be enforced with Benders cuts alone!

**Where for** \( s = 1, \ldots, S \)

\[
Q_s(x) = \min q_s^\top y \\
\text{s.t.} \quad W_s y = h_s - T_s x \\
\quad y \in \mathbb{R}_{+}^{n_2} \times \mathbb{Z}_{+}^{p_2}
\]
Pure binary first-stage

Suppose all first-stage variables are binary

\[
\begin{align*}
\min & \quad c^\top x + \sum_{s=1}^{S} p_s \theta_s \\
\text{s.t.} & \quad Ax \geq b \\
& \quad \theta_s \geq Q_s(x), \quad s = 1, \ldots, S \\
& \quad x \in \{0, 1\}^{n_1}
\end{align*}
\]

Integer \(L\)-shaped cuts

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

\[
\theta_s \geq Q_s(\hat{x}) - (Q_s(\hat{x}) - L) \left( \sum_{j: \hat{x}_j = 1} (1 - x_j) + \sum_{j: \hat{x}_j = 0} x_j \right)
\]
Pure binary first-stage

Suppose all first-stage variables are binary

**Integer $L$-shaped cuts**

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

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

Proof:
- If $x = \hat{x}$: Inequality becomes $\theta_s \geq Q_s(\hat{x})$ \checkmark
- If $x \neq \hat{x}$: Summation term $\geq 1$

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

$$\leq Q_s(\hat{x}) - (Q_s(\hat{x}) - L) = L \leq \theta_s$$
Pure binary first-stage

Suppose all first-stage variables are binary

**Integer L-shaped cuts**

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

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

- Proof also demonstrates that integer L-shaped cuts are sufficient to define $Q_s(x)$ at integer points
- NB: Calculating an integer L-shaped cut requires solving MIP to evaluate $Q_s(\hat{x})$
Pure binary first-stage

Integer $L$-shaped cuts can be used exactly as Benders cuts were used in case of continuous recourse

- Solve sequence of MIP master problems
- Use them as cuts within branch-and-cut algorithm

But integer $L$-shaped should not be used alone!

- VERY WEAK cuts. Cut defined by $\hat{x}$ probably only useful for $x = \hat{x}$
- $\Rightarrow$ May require one cut for every feasible solution

Instead, combine with all the techniques we discussed for continuous recourse

- Benders cuts from LP relaxation: see, e.g., [Angulo et al., 2016]
- Master problem cuts based on integrality of first-stage variables and subset of Benders cuts
- Subproblem cuts based on integrality of first and second-stage variables
Many other cases

Key ingredient in each case

Use cuts/branching to enforce constraint $\theta_s \geq Q_s(x)$ for $x$ feasible to first-stage problem

Pure binary first stage:
- Split cuts, transfer from one scenario to another: [Sen and Higle, 2005]
- Gomory cuts: [Gade et al., 2014]
- Fenchel cuts: [Ntaimo, 2013]
- Coordination branching: [Alonso-Ayuso et al., 2003]
- Lagrangian cuts: [Zou et al., 2016] (Ahmed’s semiplenary on Friday)

Pure integer first and second-stage:
- Gomory cuts [Zhang and Küçükyavuz, 2014]
Other cases (cont’d)

Mixed binary in first and second-stage:
- Lift-and-project (split) cuts: [Carøe, 1998, Tanner and Ntaimo, 2008]
- Reformulation linearization technique: [Sherali and Zhu, 2007]
- Disjunctive cuts from branch-and-cut tree: [Sen and Sherali, 2006]

Pure integer second-stage:
- Reformulation, integer subproblems, specialized branching: [Ahmed et al., 2004]

Simple integer second-stage recourse:
- Cuts and reformulation: [Louveaux and van der Vlerk, 1993]

General mixed-integer first and second-stage: ???
Variable Splitting

Idea

- Create copies of the first-stage decision variables for each scenario

Recall: Extensive form

\[
z^{SMIP} = \min c^\top x + \sum_{s=1}^{S} p_s q_s^\top y_s
\]

s.t. \(Ax \geq b\)

\[
T_s x + W_s y_s = h_s \quad s = 1, \ldots, S
\]

\[
x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+
\]

\[
y_s \in \mathbb{R}^{n_2}_+ \times \mathbb{Z}^{p_2}_+, \quad s = 1, \ldots, S
\]
Variable Splitting

Idea

- Create copies of the first-stage decision variables for each scenario

Copy first-stage variables

\[ z^{SMIP} = \min \sum_{s=1}^{S} p_s (c^T x_s + q_s^T y_s) \]

\[ Ax_s \geq b \quad s = 1, \ldots, S \]

\[ T_s x_s + W_s y_s = h_s \quad s = 1, \ldots, S \]

\[ x_s = \sum_{s'=1}^{S} p_{s'} x_{s'} \quad s = 1, \ldots, S \]

\[ x_s \in \mathbb{R}_{+}^{n_1} \times \mathbb{Z}_{+}^{p_1}, \quad s = 1, \ldots, S \]

\[ y_s \in \mathbb{R}_{+}^{n_2} \times \mathbb{Z}_{+}^{p_2}, \quad s = 1, \ldots, S \]
Relax nonanticipativity

- The constraints \( x_s = \sum_{s' = 1}^S p_{s'} x_{s'} \) are called *nonanticipativity constraints*.
- Relax these constraints using *Lagrangian Relaxation* with dual vectors \( \lambda = (\lambda_1, \ldots, \lambda_S) \):

\[
\mathcal{L}(\lambda) := \min \sum_{s=1}^S p_s (c^T x_s + q_s^T y_s) + \sum_{s=1}^S p_s \lambda_s^T (x_s - \sum_{s'=1}^S p_{s'} x_{s'})
\]

\[
Ax_s \geq b \quad s = 1, \ldots, S
\]

\[
T_s y_s + W_s y_s = h_s \quad s = 1, \ldots, S
\]

\[
x_s \in \mathbb{R}^{n_1} \times \mathbb{Z}^{p_1}_+, \quad s = 1, \ldots, S
\]

\[
y_s \in \mathbb{R}^{n_2} \times \mathbb{Z}^{p_2}_+, \quad s = 1, \ldots, S
\]

- Rewrite the objective (\( \bar{\lambda} = \sum_{s=1}^S p_s \lambda_s \)):

\[
\sum_{s=1}^S p_s \left( (c + (\lambda_s - \bar{\lambda}))^T x_s + q_s^T y_s \right)
\]
Relax nonanticipativity

- Rewritten objective ($\bar{\lambda} = \sum_{s=1}^{S} p_s \lambda_s$):

  $$\sum_{s=1}^{S} p_s \left( (c + (\lambda_s - \bar{\lambda}))^T x_s + q_s^T y_s \right)$$

- Normalize $\lambda_s$ so that $\bar{\lambda} = 0$

- Lagrangian relaxation problem decomposes: $\mathcal{L}(\lambda) = \sum_s p_s D_s(\lambda_s)$
  where

  $$D_s(\lambda_s) := \min \ (c + \lambda_s)^T x + q_s^T y_s$$

  s.t. $Ax \geq b$, $T_s x + W_s y = h_s$

  $x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+$, $y \in \mathbb{R}^{n_2}_+ \times \mathbb{Z}^{p_2}_+$

- Each subproblem is a deterministic mixed-integer program
For any $\lambda = (\lambda_1, \ldots, \lambda_S)$ with $\sum_s p_s \lambda_s = 0$,

\[ \mathcal{L}(\lambda) \leq z^{SMIP} \]

**Lagrangian dual**

Find best lower bound:

\[ w^{LD} := \max \left\{ \mathcal{L}(\lambda) : \sum_{s=1}^{S} p_s \lambda_s = 0 \right\} \]
Strength of Lagrangian dual

**Theorem**

The Lagrangian dual bound satisfies

\[
    w^{LD} = \min \left\{ c^\top x + \sum_{s=1}^S p_s y_s : (x, y_s) \in \text{conv}(X_s), s = 1, \ldots, S \right\}
\]

where for \( s = 1, \ldots, S \)

\[
    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}_+ \}
\]

- In general \( w^{LD} < z^{SMIP} \)
- But \( w^{LD} \geq z^{SLP} \) (the usual LP relaxation)
- \( w^{LD} \) at least as good as any bound obtained using cuts in single scenario subproblems
- In many test instances, \( w^{LD} \) is very close to \( z^{SMIP} \)
Solving the Lagrangian dual

**Lagrangian dual**

Find best lower bound:

\[ w^{LD} := \max \left\{ \mathcal{L}(\lambda) : \sum_{s=1}^{S} p_s \lambda_s = 0 \right\} \]

- \( \lambda \): High-dimensional \((S \times n)\)
- \( \mathcal{L}(\lambda) \): Non-smooth, concave function of \( \lambda \)
- Subgradients of \( \mathcal{L}(\lambda) \): Solve \( S \) deterministic MIP problems

Convex program, but challenging!
Subgradient algorithm

Iteration $k$

- Solve scenario MIP problems to evaluate $D_s(\lambda_s^k)$ and obtain solution $x_s^k$, $s = 1, \ldots, S$
- Calculate $\bar{x}^k = \sum_s p_s x_s^k$
- Update $\lambda_s$, $s = 1, \ldots, S$ ($\rho_k$ is a predetermined step size)

$$\lambda_{s}^{k+1} \leftarrow \lambda_{s}^{k} + \rho_k \left( x_s^k - \bar{x}^k \right)$$

subgradient at $\lambda^k$

Properties

- Converges for proper choice of step sizes
- But slow in practice and very sensitive to step-size choices
Cutting plane algorithm

Idea (similar to Benders!)

- Formulate a master problem

\[
\max_{\lambda, \theta} \left\{ \sum_s p_s \theta_s : \theta_s \leq D_s(\lambda_s), \sum_s p_s \lambda_s = 0 \right\}
\]

- Use subgradient cuts to approximate the constraints \( \theta_s \leq D_s(\lambda_s) \) in a relaxed master problem (RMP)

\[
\max \sum_s p_s \theta_s \\
\text{s.t. } \theta_s \leq D_s(\lambda_s^i) + (x_s^i - \bar{x}^i)^\top (\lambda_s - \lambda_s^i), \quad i = 1, \ldots, k, \ s = 1, \ldots, S \\
\sum_s p_s \lambda_s = 0
\]

- Solve RMP, solve MIP subproblems to find cuts, repeat
Improvements to cutting plane algorithm

Bundle-Regularization techniques can be applied to improve performance

- General idea: Add objective term or constraint to encourage/require RMP solutions to not move “too far” in consecutive iterations
- [Ruszczyński, 1986]: Proximal bundle applied in SLP
- [Lemaréchal et al., 1995]: Bundle-level method
- [Linderoth and Wright, 2003]: Trust region applied in SLP
- [Zverovich et al., 2012]: Numerical comparison of SLP
- [Lubin et al., 2013]: Numerical comparison for Lagrangian dual, parallel implementation
Progressive hedging

Progressive hedging: [Rockafellar and Wets, 1991]
- Elegant algorithm for solving \textit{primal and dual} for convex stochastic programs
- Equivalent to alternating direction method of multipliers

Iteration $k$
- Solve augmented (convex!) scenario problems and obtain solution $x_s^k$, $s = 1, \ldots, S$:

$$D_s^\rho(\lambda_s; \bar{x}^{k-1}) := \min \ (c + \lambda_s)^\top x + q_s^\top y + (\rho/2)\|x - \bar{x}^{k-1}\|_2^2$$
  \[ \text{s.t. } Ax \geq b, \ T_s x + W_s y = h_s \]
  \[ x \in \mathbb{R}^{n_1}_+, \ y \in \mathbb{R}^{n_2}_+ \]
- Calculate $\bar{x}^k = \sum_s p_s x_s^k$
- Update $\lambda_s$, $s = 1, \ldots, S$ ($\rho$ is a fixed step size)

$$\lambda_s^{k+1} \leftarrow \lambda_s^k + \rho(x_s^k - \bar{x}^k)$$
Progressive hedging for SMIP?

What if we just solve quadratic MIP subproblems?

\[ D^\rho_s(\lambda_s; \bar{x}^{k-1}) := \min (c + \lambda_s)^T x + q_s^T y + (\rho/2)\|x - \bar{x}^{k-1}\|_2^2 \]

s.t. \((x, y) \in X_s\)

where (recall):

\[ X_s = \{(x, y) : Ax \geq b, T_s x + W_s y = h_s \} \]

\[ x \in \mathbb{R}^{n_1}_+ \times \mathbb{Z}^{p_1}_+, y \in \mathbb{R}^{n_2}_+ \times \mathbb{Z}^{p_2}_+ \}

No convergence theory, but:

- [Watson et al., 2010] Basis of effective heuristic for primal feasible solutions
- [Gade et al., 2016] Heuristic search for good dual solutions \(\lambda_s\) (sensitive to choice of \(\rho\))
Progressive hedging for SMIP?

Suppose we could solve this subproblem:

$$\min \left\{ (c + \lambda_s)^\top x + q^\top_s y + (\rho/2) \|x - \bar{x}^{k-1}\|_2^2 : (x, y) \in \text{conv}(X_s) \right\}$$

Problem is convex $\Rightarrow$ Progressive hedging works!

- Dual solution $\lambda_s$ converges to optimum of Lagrangian dual
- Challenge: $\text{conv}(X_s)$ not known explicitly

[Boland et al., 2016]: Use inner approximation of $\text{conv}(X_s)$

- Update inner approximation by solving standard MILP subproblems

$$D_s(\lambda_s) = \min \left\{ (c + \lambda_s)^\top x + q^\top_s y : (x, y) \in X_s \right\}$$

- Each iteration requires solving $S$ “one scenario” QP’s and MILP’s
- See Dandurand talk for details: Monday, 27-June (5pm)
Closing the gap

Suppose we have (approximately) solved Lagrangian dual

- $\hat{\lambda} = (\hat{\lambda}^1, \ldots, \hat{\lambda}^S)$ and $\mathcal{L}(\hat{\lambda}) \leq z^{SMIP}$
- Primal subproblem solutions: $\hat{x}_1, \ldots, \hat{x}_S$
- Problem: $\hat{x}_s$ possibly not all equal!

Options:

- Find a heuristic solution $\Rightarrow z^{UB}$, compare to $\mathcal{L}(\hat{\lambda})$
- **Dual decomposition** [Carøe and Schultz, 1999]: Use Lagrangian dual in branch-and-bound algorithm
  - Implemented in DDSIP [Märkert and Gollmer, 2016]

Simple heuristic (works assuming relatively complete recourse):

- Choose any first-stage scenario solution $\hat{x}_s$
- Fix $x = \hat{x}_s$, and solve all second-stage subproblems
Branch-and-bound

Given approximate Lagrangian dual solution

- \( \hat{\lambda} = (\hat{\lambda}_1, \ldots, \hat{\lambda}_S) \) and \( \mathcal{L}(\hat{\lambda}) \leq z^{SMIP} \)
- Primal subproblem solutions: \( \hat{x}_1, \ldots, \hat{x}_S, \bar{x} = \sum_s \hat{x}_s \)

How to branch?

- Solution is infeasible \( \iff \exists i, s \text{ with } \hat{x}_{si} \neq \bar{x}_i \)
- If \( x_i \) is an integer decision variable:
  - Branch 1: \( x_i \leq \lfloor \bar{x}_i \rfloor \), Branch 2: \( x_i \geq \lceil \bar{x}_i \rceil \)
- Else:
  - Branch 1: \( x_i \leq \bar{x}_i \), Branch 2: \( x_i \geq \bar{x}_i \)
- Enforce branching constraints in all scenario subproblems
  - At least one subproblem solution must change!
  - Algorithm is finite if accept solutions as “feasible” when \( |\hat{x}_{si} - \bar{x}_i| \leq \delta \ \forall i, s \) for some tolerance \( \delta > 0 \)
Binary first-stage variables: Fewer nonanticipativity constraints?

Suppose $x_i$ is a binary variable: Clever modeling trick?

- Standard nonanticipitivitiy constraints: $S$ constraints
  \[ x_{si} = \sum_{s'} x_{s'i} \quad \forall s \]

- As $x_i \in \{0, 1\}$, can be enforced more compactly, e.g.,
  \[ \sum_{s=2}^{S} x_{si} = (S - 1)x_{1i} \]

  - $x_{1i} = 1 \iff \sum_{s=2}^{S} x_{si} = S - 1 \iff x_{si} = 1, s = 2, \ldots, S$

$n$ binary variables $\Rightarrow$ Only $n$ Lagrangian dual variables instead of $nS$!

Almost surely a **BAD IDEA**!

- Strength of Lagrangian dual bound can be (significantly) weaker $\Rightarrow$ More branching!
- Yet calculating Lagrangian dual still requires solving $S$ MIP subproblems
Scenario Bundling

Simple Idea

- Partition scenario set as: \( \{1, \ldots, S\} = \bigcup_{k=1}^{K} B_k, \ B_k \cap B_{k'} = \emptyset \)
- When doing decomposition, treat scenarios within each “bundle” as a single scenario

- Idea can be applied in either LP or Lagrangian based approaches
- See, e.g., [Wets, 1988], [Gade et al., 2016], [Escudero et al., 2013], [Crainic et al., 2014]
Scenario bundling in LP based methods

Scenario decomposition

\[
\begin{align*}
\min & \quad c^T x + \sum_{s = 1}^{S} p_s Q_s(x) \\
\text{s.t.} & \quad Ax \geq b \\
& \quad x \in \mathbb{R}^{n_1} \times \mathbb{Z}^{p_1}_+
\end{align*}
\]

where for \( s = 1, \ldots, S \)

\[
Q_s(x) = \min q_s^T y
\]

\[
\text{s.t.} \quad W_s y = h_s - T_s x
\]

\[
\quad y \in \mathbb{R}^{n_2} \times \mathbb{Z}^{p_2}_+
\]

Bundle decomposition

\[
\begin{align*}
\min & \quad c^T x + \sum_{k = 1}^{K} \tilde{Q}_k(x) \\
\text{s.t.} & \quad Ax \geq b \\
& \quad x \in \mathbb{R}^{n_1} \times \mathbb{Z}^{p_1}_+
\end{align*}
\]

where for \( k = 1, \ldots, K \)

\[
\tilde{Q}_k(x) = \min \sum_{s \in B_k} p_s q_s^T y_s
\]

\[
\text{s.t.} \quad W_s y_s = h_s - T_s x, \quad s \in B_k
\]

\[
\quad y_s \in \mathbb{R}^{n_2} \times \mathbb{Z}^{p_2}_+, \quad s \in B_k
\]

- Just apply all methods to “Bundle” formulation, with bundles treated like large scenarios
Scenario bundling in LP based methods

Cost of bundle formulation
- Bundle subproblems are larger

Potential benefits
- Faster convergence to solve relaxations (fewer $\theta_k$ vars $\Rightarrow$ fewer cuts)
- Fewer subproblems to solve per iteration
- Subproblem cuts derived from bundles can be stronger

Questions (answer empirically?)
- How much bundling gives best trade-off?
- Which scenarios bundle?
Scenario bundling in Lagrangian approach

Create one copy of first-stage variables per bundle

\[ z^{SMIP} = \min \sum_{k=1}^{K} \sum_{s \in B_k} p_s (c^T x_k + q_s^T y_s) \]

\[ Ax_k \geq b \quad k = 1, \ldots, K \]

\[ T_s x_k + W_s y_s = h_s \quad s \in B_k, k = 1, \ldots, K \]

\[ x_k = \sum_{k'=1}^{K} x_{k'} \quad k = 1, \ldots, K \]

\[ x_k \in \mathbb{R}_+^{n_1} \times \mathbb{Z}_+^{p_1}, \quad k = 1, \ldots, K \]

\[ y_s \in \mathbb{R}_+^{n_2} \times \mathbb{Z}_+^{p_2}, \quad s = 1, \ldots, S \]
Scenario bundling in Lagrangian approach

Relax the bundle nonanticipativity constraints with dual variables $\lambda_k$, $k = 1, \ldots, K$

$$x_k = \sum_{k' = 1}^{K} x_{k'}, \quad k = 1, \ldots, K$$

Bundle subproblems become:

$$D_k(\lambda_k) := \min (c + \lambda_k)^T x + \sum_{s \in B_k} p_s q_s^T y_s$$

s.t. $Ax \geq b$, $T_s x + W_s y = h_s$, $s \in B_k$

$y_s \in \mathbb{R}_{+}^{n_2} \times \mathbb{Z}_{+}^{p_2}$, $s \in B_k$

$x \in \mathbb{R}_{+}^{n_1} \times \mathbb{Z}_{+}^{p_1}$

Similar trade-offs:

- Fewer dual variables $\Rightarrow$ Lagrangian dual may be solved in fewer iterations
- Fewer MIP subproblems, but each is larger
- **Better Lagrangian bound** (possibly much better!)
Parting thoughts

The future is bright for stochastic mixed-integer programming!

- Many important applications
- Significant and steady progress in methods

But we have work to do

- SMIP **not** well solved, even in simplest case of continuous recourse
- Many current test instances: 10 – 20 first-stage variables, ≤ 50 scenarios

[Cameron Luedtke]
Let’s hit the beach!

Questions?
- jim.luedtke@wisc.edu

Thanks!
- Merve Bodur: Pictures
- Shabbir Ahmed: Course notes

And apologies...
- \( \mathbb{P}(\text{Reference list is incomplete}) = 1 \)


Strengthened Benders cuts for stochastic integer programs with continuous recourse.

*IJOC.*

**Bodur, M. and Luedtke, J. (2016).**
Mixed-integer rounding enhanced Benders decomposition for multiclass service system staffing and scheduling with arrival rate uncertainty.

*Management Science.*

**Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Linderoth, J., Luedtke, J., and Oliveira, F. (2016).**
Progressive hedging with a Frank-Wolfe method for computing stochastic mixed-integer programming Lagrangian dual bounds.

[optimization-online.org.](optimization-online.org)

**Carøe, C. C. (1998).**
"Decomposition in Stochastic Integer Programming."
PhD thesis, Department of Operations Research, University of Copenhagen, Denmark.


Odds and Ends

Scenario Bundling


Scenarios and policy aggregation in optimization under uncertainty.

A regularized decomposition method for minimizing a sum of polyhedral functions.

The $C^3$ theorem and a $D^2$ algorithm for large scale stochastic mixed-integer programming: set convexification.

Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming.
Mathematical Programming, 106:203–223.


