Stochastic Optimization
Decomposition methods for two-stage problems

V. Leclère

November 25, 2026





Presentation Outline

1. Scenario decomposition
   - Dual decomposition
   - Progressive hedging
   - Bounds
2. Stage decomposition
   - The L-shaped method
   - ✠ The adaptive partition method
3. Comparing the methods

Why bother?

- The extensive formulation of your two-stage problem is too large for a single call to a solver, and you need a method that exploits its shape.
- You have a working solver for the deterministic scenario problems, and want to use it without writing a new one.
- You want to know how far you are from the optimum.

Two families of methods, both exploiting the two-stage structure:

- scenario decomposition: one problem per scenario, coordinated by multipliers (dual decomposition, progressive hedging);
- stage decomposition: a first-stage problem with an approximated recourse value (L-shaped, adaptive partitions).

Why bother?

- The extensive formulation of your two-stage problem is too large for a single call to a solver, and you need a method that exploits its shape.
- You have a working solver for the deterministic scenario problems, and want to use it without writing a new one.
- You want to know how far you are from the optimum.

Two families of methods, both exploiting the two-stage structure:

- scenario decomposition: one problem per scenario, coordinated by multipliers (dual decomposition, progressive hedging);
- stage decomposition: a first-stage problem with an approximated recourse value (L-shaped, adaptive partitions).

Presentation Outline

1. Scenario decomposition
   - Dual decomposition
   - Progressive hedging
   - Bounds
2. Stage decomposition
   - The L-shaped method
   - ✠ The adaptive partition method
3. Comparing the methods

Two-stage problem: the extensive formulation

(FSN): the noise takes the values ξ^ω with probabilities $\pi^\omega > 0$, $\omega \in \Omega := [S]$.

Proposition (Extensive formulation)

Under (FSN), the two-stage problem is equivalent to the deterministic problem

$$\begin{aligned} \min_{y_1, (y_2^\omega)_{\omega\in\Omega}} \quad & \sum_{\omega\in\Omega} \pi^\omega \ell(y_1, \xi^\omega, y_2^\omega) \\ s.t. \quad & g(y_1, \xi^\omega, y_2^\omega) \leq 0, \qquad \forall \omega \in \Omega\,, \end{aligned}$$

in $n_1 + S\,n_2$ variables, with $y_2^\omega = \boldsymbol{y}_2$ on the event $\{\boldsymbol{\xi} = \xi^\omega\}$.

- Often too many variables for one solve.
- The S scenarios are one problem only because they share y_1.

One copy of the first stage per scenario

Give each scenario its own copy y_1^ω of the first-stage decision, and require the copies to agree: the problem in scenario form

$$\begin{aligned} \min_{(y_1^\omega)_{\omega\in\Omega},\,(y_2^\omega)_{\omega\in\Omega}} \quad & \sum_{\omega\in\Omega} \pi^\omega \ell(y_1^\omega, \xi^\omega, y_2^\omega) \\ \text{s.t.} \quad & (y_1^\omega, y_2^\omega) \in \mathcal{C}^\omega, && \forall \omega \in \Omega \\ & (y_1^\omega, y_2^\omega)_{\omega\in\Omega} \in \mathcal{N} \end{aligned}$$

with $\mathcal{C}^\omega := \{(y_1, y_2) \mid y_1 \in \mathcal{Y}_1,\ y_2 \in \mathcal{Y}_2(y_1, \xi^\omega)\}$ the admissible pairs of scenario ω, and $\mathcal{N}$ the nonanticipative tuples, those whose first-stage copies agree: a linear subspace that asks nothing of the recourse.

One copy of the first stage per scenario

Give each scenario its own copy y_1^ω of the first-stage decision, and require the copies to agree: the problem in scenario form

$$\begin{aligned} \min_{(y_1^\omega)_{\omega\in\Omega},\,(y_2^\omega)_{\omega\in\Omega}} \quad & \sum_{\omega\in\Omega} \pi^\omega \ell(y_1^\omega, \xi^\omega, y_2^\omega) \\ \text{s.t.} \quad & (y_1^\omega, y_2^\omega) \in \mathcal{C}^\omega, && \forall \omega \in \Omega \\ & (y_1^\omega, y_2^\omega)_{\omega\in\Omega} \in \mathcal{N} \end{aligned}$$

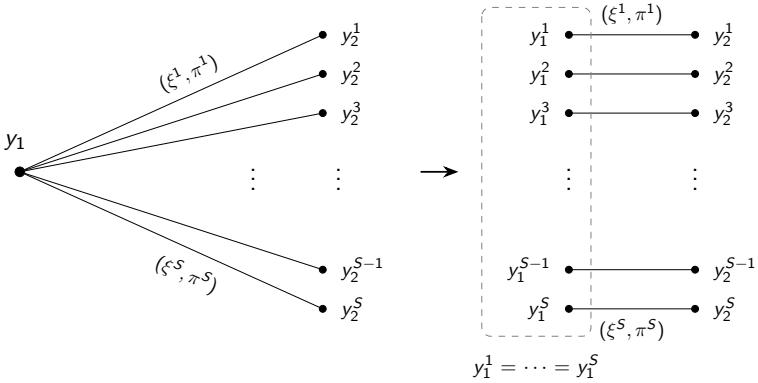
with $\mathcal{C}^\omega := \{(y_1, y_2) \mid y_1 \in \mathcal{Y}_1,\ y_2 \in \mathcal{Y}_2(y_1, \xi^\omega)\}$ the admissible pairs of scenario ω, and $\mathcal{N}$ the nonanticipative tuples, those whose first-stage copies agree: a linear subspace that asks nothing of the recourse.

That last constraint is the nonanticipativity constraint, the information constraint written on the copies. Two ways of writing it:

$$y_1^\omega = y_1^{\omega'} \ \ \forall \omega, \omega' \in \Omega \qquad \text{or} \qquad y_1^\omega = \bar{y}_1 := \sum_{\omega'\in\Omega} \pi^{\omega'} y_1^{\omega'} \ \ \forall \omega \in \Omega\,.$$

Pairwise, or each equal to the average $\bar{y}_1$: we use the second, one multiplier per scenario.

Extensive formulation and scenario form



Node form on the left; scenario form on the right.

Dualizing nonanticipativity

Nonanticipativity, in the form we use:

$$y_1^\omega = \bar{y}_1 := \sum_{\omega' \in \Omega} \pi^{\omega'} y_1^{\omega'} \quad \forall \omega \in \Omega \,.$$

Dualize it with a multiplier $\lambda^\omega \in \mathbb{R}^{n_1}$ per scenario, each term weighted by π^ω like the scenario's cost:

$$\mathcal{L}(y_1, y_2, \lambda) := \sum_{\omega \in \Omega} \pi^\omega \Big(\ell\big(y_1^\omega, \xi^\omega, y_2^\omega\big) + \lambda^{\omega\top} \big(y_1^\omega - \bar{y}_1\big) \Big) \,.$$

Write $\bar{\lambda} := \sum_\omega \pi^\omega \lambda^\omega$. The multiplier term is

$$\begin{aligned}
\sum_{\omega \in \Omega} \pi^\omega \lambda^{\omega\top} \big(y_1^\omega - \bar{y}_1\big) &= \sum_{\omega \in \Omega} \pi^\omega \lambda^{\omega\top} y_1^\omega \; - \; \bar{\lambda}^\top \bar{y}_1 && (\bar{y}_1 \text{ does not depend on } \omega) \\
&= \sum_{\omega \in \Omega} \pi^\omega \lambda^{\omega\top} y_1^\omega \; - \; \sum_{\omega \in \Omega} \pi^\omega \, \bar{\lambda}^\top y_1^\omega && (\text{definition of } \bar{y}_1) \\
&= \sum_{\omega \in \Omega} \pi^\omega \big(\lambda^\omega - \bar{\lambda}\big)^\top y_1^\omega \,.
\end{aligned}$$

Only $\lambda - \bar{\lambda}$ acts, so we restrict to centred multipliers $\Lambda := \big\{ (\lambda^\omega)_{\omega \in \Omega} \mid \sum_\omega \pi^\omega \lambda^\omega = 0 \big\}$, and then $\bar{y}_1$ vanishes: nothing couples the scenarios any more.

Dualizing nonanticipativity

Nonanticipativity, in the form we use:

$$y_1^\omega = \bar{y}_1 := \sum_{\omega' \in \Omega} \pi^{\omega'} y_1^{\omega'} \quad \forall \omega \in \Omega \,.$$

Dualize it with a multiplier $\lambda^\omega \in \mathbb{R}^{n_1}$ per scenario, each term weighted by π^ω like the scenario's cost:

$$\mathcal{L}(y_1, y_2, \lambda) := \sum_{\omega \in \Omega} \pi^\omega \Big(\ell\big(y_1^\omega, \xi^\omega, y_2^\omega\big) + {\lambda^\omega}^\top \big(y_1^\omega - \bar{y}_1\big) \Big) \,.$$

Write $\bar{\lambda} := \sum_\omega \pi^\omega \lambda^\omega$. The multiplier term is

$$\begin{aligned}
\sum_{\omega \in \Omega} \pi^\omega {\lambda^\omega}^\top \big(y_1^\omega - \bar{y}_1\big) &= \sum_{\omega \in \Omega} \pi^\omega {\lambda^\omega}^\top y_1^\omega \;-\; \bar{\lambda}^\top \bar{y}_1 && (\bar{y}_1 \text{ does not depend on } \omega) \\
&= \sum_{\omega \in \Omega} \pi^\omega {\lambda^\omega}^\top y_1^\omega \;-\; \sum_{\omega \in \Omega} \pi^\omega \, \bar{\lambda}^\top y_1^\omega && (\text{definition of } \bar{y}_1) \\
&= \sum_{\omega \in \Omega} \pi^\omega \big(\lambda^\omega - \bar{\lambda}\big)^\top y_1^\omega \,.
\end{aligned}$$

Only $\lambda - \bar{\lambda}$ acts, so we restrict to centred multipliers $\Lambda := \big\{(\lambda^\omega)_{\omega \in \Omega} \mid \sum_\omega \pi^\omega \lambda^\omega = 0\big\}$, and then $\bar{y}_1$ vanishes: nothing couples the scenarios any more.

Dualizing nonanticipativity

Nonanticipativity, in the form we use:

$$y_1^\omega = \bar{y}_1 := \sum_{\omega' \in \Omega} \pi^{\omega'} y_1^{\omega'} \quad \forall \omega \in \Omega \,.$$

Dualize it with a multiplier $\lambda^\omega \in \mathbb{R}^{n_1}$ per scenario, each term weighted by π^ω like the scenario's cost:

$$\mathcal{L}(y_1, y_2, \lambda) := \sum_{\omega \in \Omega} \pi^\omega \Big(\ell\big(y_1^\omega, \xi^\omega, y_2^\omega\big) + \lambda^{\omega\top} \big(y_1^\omega - \bar{y}_1\big) \Big) \,.$$

Write $\bar{\lambda} := \sum_\omega \pi^\omega \lambda^\omega$. The multiplier term is

$$\begin{aligned}
\sum_{\omega \in \Omega} \pi^\omega \lambda^{\omega\top} \big(y_1^\omega - \bar{y}_1\big) &= \sum_{\omega \in \Omega} \pi^\omega \lambda^{\omega\top} y_1^\omega \;-\; \bar{\lambda}^\top \bar{y}_1 && (\bar{y}_1 \text{ does not depend on } \omega) \\
&= \sum_{\omega \in \Omega} \pi^\omega \lambda^{\omega\top} y_1^\omega \;-\; \sum_{\omega \in \Omega} \pi^\omega \, \bar{\lambda}^\top y_1^\omega && (\text{definition of } \bar{y}_1) \\
&= \sum_{\omega \in \Omega} \pi^\omega \big(\lambda^\omega - \bar{\lambda}\big)^\top y_1^\omega \,.
\end{aligned}$$

Only $\lambda - \bar{\lambda}$ acts, so we restrict to centred multipliers $\Lambda := \big\{(\lambda^\omega)_{\omega \in \Omega} \mid \sum_\omega \pi^\omega \lambda^\omega = 0\big\}$, and then $\bar{y}_1$ vanishes: nothing couples the scenarios any more.

The dual function

Definition (Dual function)

For $\lambda \in \Lambda$,

$$
\begin{aligned}
v^{\mathrm{D}}(\lambda) &:= \inf_{(y_1^\omega, y_2^\omega) \in \mathcal{C}^\omega,\ \omega \in \Omega} \overbrace{\sum_{\omega \in \Omega} \pi^\omega \left(\ell(y_1^\omega, \xi^\omega, y_2^\omega) + \lambda^{\omega\top} y_1^\omega \right)}^{\mathcal{L}(y_1, y_2, \lambda)} \\
&= \sum_{\omega \in \Omega} \pi^\omega \underbrace{\inf \left\{ \ell(y_1, \xi^\omega, y_2) + \lambda^{\omega\top} y_1 \mid (y_1, y_2) \in \mathcal{C}^\omega \right\}}_{v^{\mathrm{D},\omega}(\lambda^\omega)} .
\end{aligned}
$$

Note that $v^{\mathrm{D},\omega}(\lambda^\omega)$ is a deterministic problem in scenario ω alone.

- One evaluation of $v^{\mathrm{D}} = S$ deterministic solves, in parallel.
- Each is the scenario problem you would write without uncertainty, plus one linear term.

Weak duality

Proposition (Weak duality)

Each $\lambda \in \Lambda$ gives a lower bound $v^{\mathrm{D}}(\lambda) \leq v^{\mathrm{SP}}$, and $0 \in \Lambda$, so

$$v^{\mathrm{PI}} = v^{\mathrm{D}}(0) \;\leq\; \sup_{\mu \in \Lambda} v^{\mathrm{D}}(\mu) \;\leq\; v^{\mathrm{SP}} \,.$$

Indeed, an admissible nonanticipative pair satisfies the dualized constraint, so its Lagrangian is its cost, and $v^{\mathrm{D}}(\lambda)$ takes the infimum over a larger set: the copies are free to disagree.

The dual problem is to find the best of these bounds, $\sup_{\lambda \in \Lambda} v^{\mathrm{D}}(\lambda)$. The next frame says when it closes the gap.

Weak duality

Proposition (Weak duality)

Each $\lambda \in \Lambda$ *gives a lower bound* $v^{\mathrm{D}}(\lambda) \leq v^{\mathrm{SP}}$, *and* $0 \in \Lambda$, *so*

$$v^{\mathrm{PI}} = v^{\mathrm{D}}(0) \;\leq\; \sup_{\mu \in \Lambda} v^{\mathrm{D}}(\mu) \;\leq\; v^{\mathrm{SP}} \,.$$

Indeed, an admissible nonanticipative pair satisfies the dualized constraint, so its Lagrangian is its cost, and $v^{\mathrm{D}}(\lambda)$ takes the infimum over a larger set: the copies are free to disagree.

The dual problem is to find the best of these bounds, $\sup_{\lambda \in \Lambda} v^{\mathrm{D}}(\lambda)$. The next frame says when it closes the gap.

Strong duality

(CVX) Convex scenario problems. Every $\mathcal{C}^\omega$ is convex, and $\ell(\cdot, \xi^\omega, \cdot)$ is convex on it.

For each scenario ω, the cost when y_1 is followed by its best recourse (convex under (CVX)), and its domain:

$$f^\omega(y_1) := \inf \left\{ \ell(y_1, \xi^\omega, y_2) \mid (y_1, y_2) \in \mathcal{C}^\omega \right\},$$
$$P^\omega := \left\{ y_1 \mid (y_1, y_2) \in \mathcal{C}^\omega \text{ for some } y_2 \right\},$$

the first-stage decisions scenario ω can live with.

Theorem (Strong duality)

Assume (CVX), v^{SP} finite, and $\bigcap_{\omega\in\Omega} \operatorname{ri} P^\omega \neq \emptyset$.

Then $\max_{\lambda\in\Lambda} v^{\mathrm{D}}(\lambda) = v^{\mathrm{SP}}$, the maximum being attained.

- Under relatively complete recourse every $P^\omega = \mathcal{Y}_1$: the qualification holds as soon as $\mathcal{Y}_1 \neq \emptyset$.
- If every $\mathcal{C}^\omega$ is polyhedral and ℓ affine: no qualification needed (LP duality).

Strong duality

(CVX) Convex scenario problems. Every $\mathcal{C}^\omega$ is convex, and $\ell(\cdot, \xi^\omega, \cdot)$ is convex on it.

For each scenario ω, the cost when y_1 is followed by its best recourse (convex under (CVX)), and its domain:

$$f^\omega(y_1) := \inf \left\{ \ell(y_1, \xi^\omega, y_2) \mid (y_1, y_2) \in \mathcal{C}^\omega \right\},$$
$$P^\omega := \left\{ y_1 \mid (y_1, y_2) \in \mathcal{C}^\omega \text{ for some } y_2 \right\},$$

the first-stage decisions scenario ω can live with.

Theorem (Strong duality)

Assume (CVX), v^{SP} finite, and $\bigcap_{\omega\in\Omega} \operatorname{ri} P^\omega \neq \emptyset$.

Then $\max_{\lambda\in\Lambda} v^{\mathrm{D}}(\lambda) = v^{\mathrm{SP}}$, the maximum being attained.

- Under relatively complete recourse every $P^\omega = \mathcal{Y}_1$: the qualification holds as soon as $\mathcal{Y}_1 \neq \emptyset$.
- If every $\mathcal{C}^\omega$ is polyhedral and ℓ affine: no qualification needed (LP duality).

Price of information

A maximizer $\lambda^\star$ is a price of information: an optimal multiplier of the constraint that keeps the first stage from seeing the scenario.

$$v^{\mathrm{PI}} = v^{\mathrm{D}}(0) \;\leq\; v^{\mathrm{D}}(\lambda^\star) = v^{\mathrm{SP}} \,.$$

At $\lambda = 0$ the multiplier term disappears and each scenario is solved alone: the dual value is the perfect-information value. Maximizing over Λ raises it by exactly $\mathrm{EVPI} = v^{\mathrm{SP}} - v^{\mathrm{PI}}$.

Proposition (♦ Prices and optimality)

Let $\tilde{y}_1$ be admissible in every scenario, with every $f^\omega(\tilde{y}_1)$ finite, and $\lambda \in \Lambda$.

$$\tilde{y}_1 \text{ optimal and } v^{\mathrm{D}}(\lambda) = v^{\mathrm{SP}} \quad\Longleftrightarrow\quad -\lambda^\omega \in \partial f^\omega(\tilde{y}_1) \;\; \forall \omega \in \Omega \,.$$

Note that each scenario is handed a marginal price at which it is individually content, and that the prices have expectation zero.

Price of information

A maximizer $\lambda^\star$ is a price of information: an optimal multiplier of the constraint that keeps the first stage from seeing the scenario.

$$v^{\mathrm{PI}} = v^{\mathrm{D}}(0) \;\leq\; v^{\mathrm{D}}(\lambda^\star) = v^{\mathrm{SP}} \,.$$

At $\lambda = 0$ the multiplier term disappears and each scenario is solved alone: the dual value is the perfect-information value. Maximizing over Λ raises it by exactly $\mathrm{EVPI} = v^{\mathrm{SP}} - v^{\mathrm{PI}}$.

Proposition (♦ Prices and optimality)

Let $\tilde{y}_1$ be admissible in every scenario, with every $f^\omega(\tilde{y}_1)$ finite, and $\lambda \in \Lambda$.

$$\tilde{y}_1 \text{ optimal and } v^{\mathrm{D}}(\lambda) = v^{\mathrm{SP}} \quad\Longleftrightarrow\quad -\lambda^\omega \in \partial f^\omega(\tilde{y}_1) \;\; \forall \omega \in \Omega \,.$$

Note that each scenario is handed a marginal price at which it is individually content, and that the prices have expectation zero.

Price of information

A maximizer $\lambda^\star$ is a price of information: an optimal multiplier of the constraint that keeps the first stage from seeing the scenario.

$$v^{\mathrm{PI}} = v^{\mathrm{D}}(0) \;\leq\; v^{\mathrm{D}}(\lambda^\star) = v^{\mathrm{SP}} \,.$$

At $\lambda = 0$ the multiplier term disappears and each scenario is solved alone: the dual value is the perfect-information value. Maximizing over Λ raises it by exactly $\mathrm{EVPI} = v^{\mathrm{SP}} - v^{\mathrm{PI}}$.

Proposition (♦ Prices and optimality)

Let $\tilde{y}_1$ be admissible in every scenario, with every $f^\omega(\tilde{y}_1)$ finite, and $\lambda \in \Lambda$.

$$\tilde{y}_1 \text{ optimal and } v^{\mathrm{D}}(\lambda) = v^{\mathrm{SP}} \quad\Longleftrightarrow\quad -\lambda^\omega \in \partial f^\omega(\tilde{y}_1) \;\; \forall \omega \in \Omega \,.$$

Note that each scenario is handed a marginal price at which it is individually content, and that the prices have expectation zero.

Climbing the dual

At λ, scenario ω pays λ^ω per unit of first-stage decision and answers with its copy y_1^ω.

- Copy above the average $\bar{y}_1$: the price is too low, raise it.
- Copy below: lower it.

Correcting every price in proportion to its scenario's disagreement,

$$\lambda^\omega \leftarrow \lambda^\omega + \tau_k \left(y_1^\omega - \bar{y}_1\right), \qquad \forall \omega \in \Omega \,,$$

with a step $\tau_k > 0$, is dual decomposition.

- The corrections have expectation zero: the multiplier stays centred.
- The prices stop moving only when the copies agree, and a $\lambda \in \Lambda$ at which they agree is an optimal price: their common value solves the original problem.
- It is supergradient ascent on the concave v^{D}.

Climbing the dual

At λ, scenario ω pays λ^ω per unit of first-stage decision and answers with its copy y_1^ω.

- Copy above the average $\bar{y}_1$: the price is too low, raise it.
- Copy below: lower it.

Correcting every price in proportion to its scenario's disagreement,

$$\lambda^\omega \leftarrow \lambda^\omega + \tau_k \left(y_1^\omega - \bar{y}_1\right), \qquad \forall \omega \in \Omega \,,$$

with a step $\tau_k > 0$, is dual decomposition.

- The corrections have expectation zero: the multiplier stays centred.
- The prices stop moving only when the copies agree, and a $\lambda \in \Lambda$ at which they agree is an optimal price: their common value solves the original problem.
- It is supergradient ascent on the concave v^{D}.

Climbing the dual

At λ, scenario ω pays λ^ω per unit of first-stage decision and answers with its copy y_1^ω.

- Copy above the average $\bar{y}_1$: the price is too low, raise it.
- Copy below: lower it.

Correcting every price in proportion to its scenario's disagreement,

$$\lambda^\omega \leftarrow \lambda^\omega + \tau_k \left(y_1^\omega - \bar{y}_1\right), \qquad \forall \omega \in \Omega \,,$$

with a step $\tau_k > 0$, is dual decomposition.

- The corrections have expectation zero: the multiplier stays centred.
- The prices stop moving only when the copies agree, and a $\lambda \in \Lambda$ at which they agree is an optimal price: their common value solves the original problem.
- It is supergradient ascent on the concave v^{D}.

Dual decomposition algorithm

Steps $\tau_k > 0$, scenario problems attaining their infimum (e.g. $\mathcal{C}^\omega$ compact, ℓ lower semicontinuous).

Start: $\lambda^0 = 0$, the perfect-information bound; $\mathrm{LB} = -\infty$, $\mathrm{UB} = +\infty$

for $k = 0, 1, 2, \ldots$ **do**

  foreach *scenario* $\omega \in \Omega$ **do** `// in parallel`

    solve $v^{\mathrm{D},\omega}(\lambda^{\omega,k}) = \min_{(y_1,y_2)\in\mathcal{C}^\omega} \ell(y_1, \xi^\omega, y_2) + \lambda^{\omega,k\top} y_1$, solution $(y_1^{\omega,k}, y_2^{\omega,k})$

  $\mathrm{LB} \leftarrow \max\left(\mathrm{LB}, \sum_\omega \pi^\omega v^{\mathrm{D},\omega}(\lambda^{\omega,k})\right)$ `// no extra solve`

  $\bar{y}_1^k = \sum_\omega \pi^\omega y_1^{\omega,k}$

  $\mathrm{UB} \leftarrow \min\left(\mathrm{UB}, \sum_\omega \pi^\omega f^\omega(\bar{y}_1^k)\right)$ `// S extra recourse solves`

  $\lambda^{\omega,k+1} = \lambda^{\omega,k} + \tau_k \left(y_1^{\omega,k} - \bar{y}_1^k\right)$

Both bounds are running best values, hence monotone.

Note that convergence asks more of the steps than $\tau_k > 0$: the results rest on strong convexity, on a diminishing step size, or on averaging the iterates.

Two iterations of dual decomposition

$$\min_{(y_1,y_2)\in\mathcal{C}^1} \ell\big(y_1,\xi^1,y_2\big) \qquad \xi^1$$

multiplier update

$$\bar{y}_1^0 = \sum_{\omega\in\Omega} \pi^\omega y_1^{\omega,0}$$

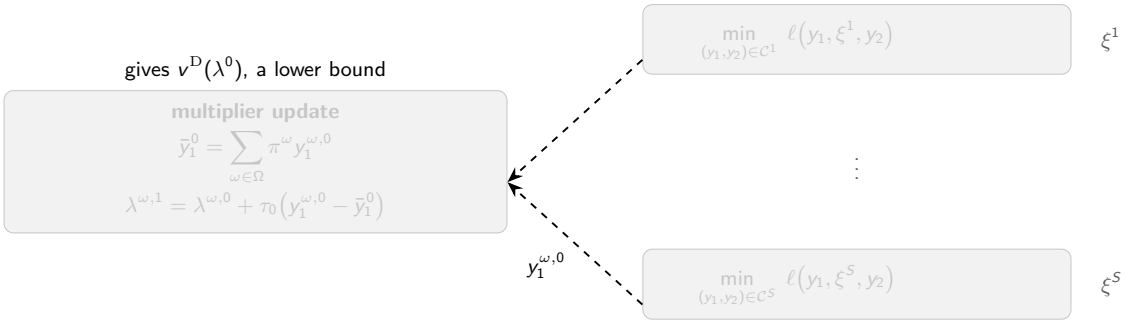
$$\lambda^{\omega,1} = \lambda^{\omega,0} + \tau_0\big(y_1^{\omega,0} - \bar{y}_1^0\big)$$

$$\vdots$$

$$\min_{(y_1,y_2)\in\mathcal{C}^S} \ell\big(y_1,\xi^S,y_2\big) \qquad \xi^S$$

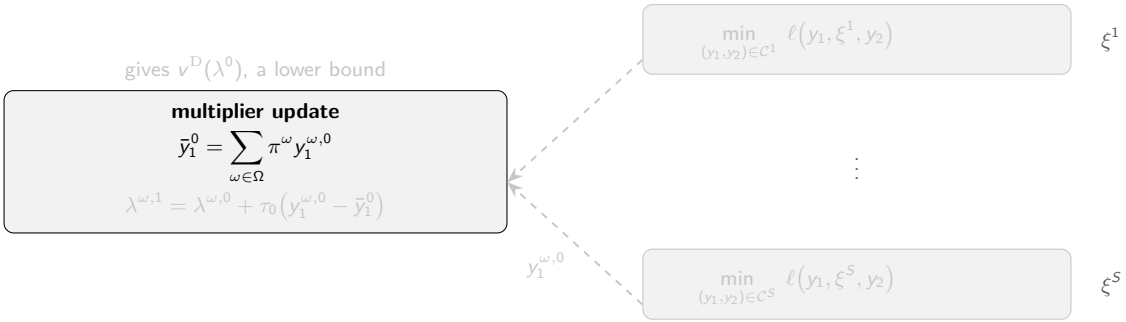
Start. $\lambda^0 = 0$: no multiplier is added to any objective, so each scenario is solved alone, as the algorithm starts it.

Two iterations of dual decomposition



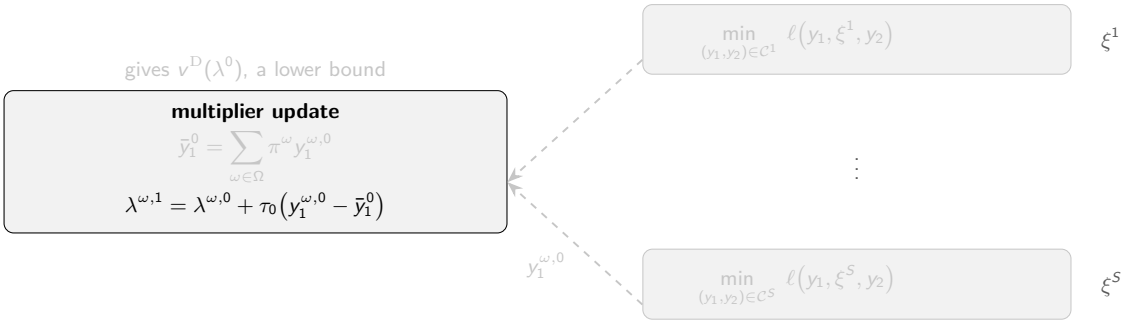
The copies come back. Scenario ω returns its copy $y_1^{\omega,0}$ and its value; the values give $v^{\mathrm{D}}(\lambda^0) = v^{\mathrm{PI}}$, the perfect-information bound, with no extra solve.

Two iterations of dual decomposition



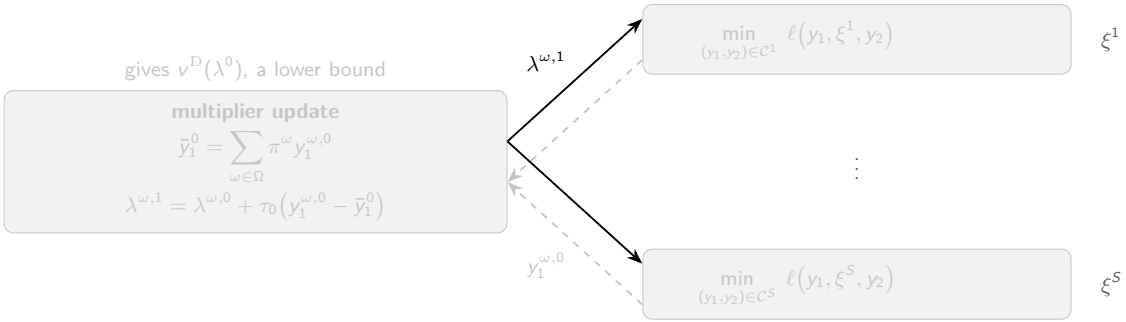
The average. The master forms $\bar{y}_1^0$, the weighted average of the copies; each copy's distance to it is its gap.

Two iterations of dual decomposition



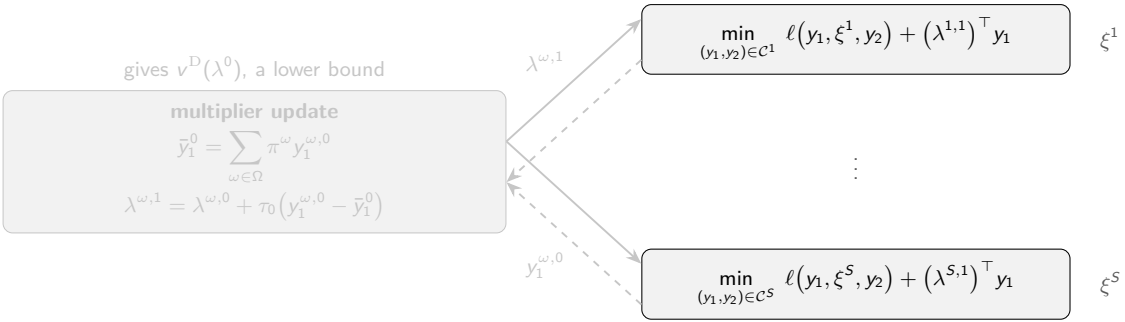
The update. Every multiplier moves by its own gap, with step τ_0. The gaps average to zero, so the multiplier stays centred.

Two iterations of dual decomposition



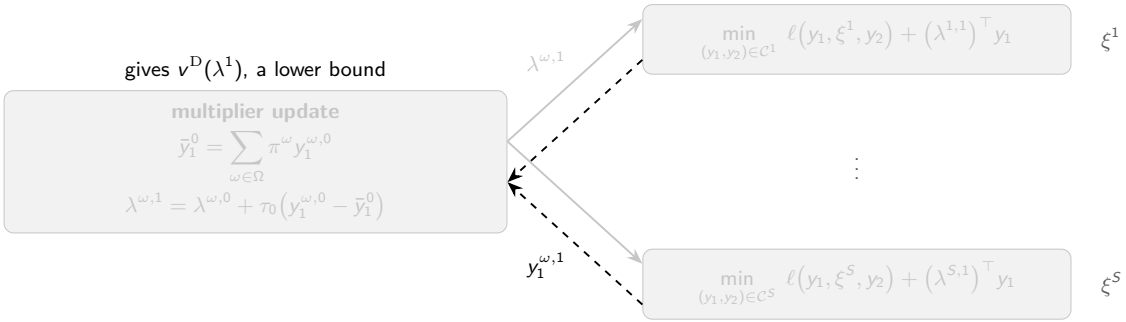
Prices out. Each scenario gets its own multiplier $\lambda^{\omega,1}$, and nothing else: no scenario sees another.

Two iterations of dual decomposition



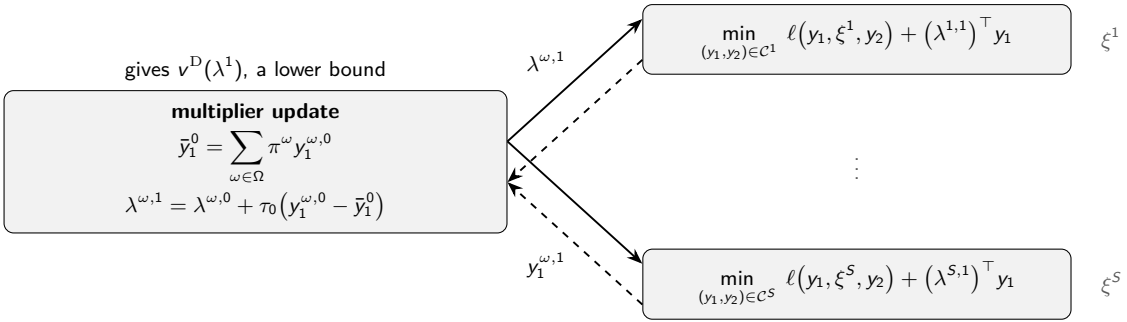
Iteration 1. Each problem now carries its own multiplier on the first-stage decision; solved again, still alone and in parallel.

Two iterations of dual decomposition



New copies. The solutions $y_1^{\omega,1}$ come back with their values: $v^{\mathrm{D}}(\lambda^1)$, and the bound kept is the best of the two. The dual is being climbed.

Two iterations of dual decomposition



The whole iteration. S scenario problems solved alone, each with its own multiplier; the master averages the copies and moves every multiplier by its gap.

Presentation Outline

1. Scenario decomposition
   - Dual decomposition
   - Progressive hedging
   - Bounds
2. Stage decomposition
   - The L-shaped method
   - ✠ The adaptive partition method
3. Comparing the methods

Progressive hedging: adding a quadratic term

On a linear scenario problem the minimizer jumps from vertex to vertex, so the copies need not meet at any price: dual decomposition may never make them agree.

So penalize the disagreement itself. Take the augmented Lagrangian: add $\frac{\tau}{2}\sum_\omega \pi^\omega \|y_1^\omega - \bar{y}_1\|^2$, then freeze $\bar{y}_1$ at the previous average. Each scenario problem becomes

$$\min_{(y_1,y_2)\in\mathcal{C}^\omega} \quad \ell(y_1,\xi^\omega,y_2) + \lambda^{\omega\top} y_1 + \frac{\tau}{2}\|y_1 - \bar{y}_1\|^2 .$$

- An augmented Lagrangian does not decompose; this one does because the average is frozen.
- Freezing is not an approximation: it is one step of the alternating direction method of multipliers (ADMM).

Progressive hedging: adding a quadratic term

On a linear scenario problem the minimizer jumps from vertex to vertex, so the copies need not meet at any price: dual decomposition may never make them agree.

So penalize the disagreement itself. Take the augmented Lagrangian: add $\frac{\tau}{2}\sum_\omega \pi^\omega \|y_1^\omega - \bar{y}_1\|^2$, then freeze $\bar{y}_1$ at the previous average. Each scenario problem becomes

$$\min_{(y_1,y_2)\in\mathcal{C}^\omega} \quad \ell(y_1,\xi^\omega,y_2) + \lambda^{\omega\top} y_1 + \frac{\tau}{2}\,\|y_1 - \bar{y}_1\|^2 \,.$$

- An augmented Lagrangian does not decompose; this one does because the average is frozen.
- Freezing is not an approximation: it is one step of the alternating direction method of multipliers (ADMM).

Progressive hedging algorithm

A proximal parameter $\tau > 0$, a tolerance $\varepsilon > 0$.

Start: each scenario alone, $(y_1^{\omega,0}, y_2^{\omega,0}) \in \arg\min_{\mathcal{C}^\omega} \ell(\cdot, \xi^\omega, \cdot)$; $\bar{y}_1^0 = \sum_\omega \pi^\omega y_1^{\omega,0}$,
$\lambda^{\omega,0} = \tau\,(y_1^{\omega,0} - \bar{y}_1^0)$, which is centred

for $k = 1, 2, \ldots$ **do**

  foreach *scenario* $\omega \in \Omega$ **do** // in parallel

    $(y_1^{\omega,k}, y_2^{\omega,k}) \in \underset{(y_1,y_2)\in\mathcal{C}^\omega}{\arg\min}\ \ell(y_1, \xi^\omega, y_2) + \lambda^{\omega,k-1\top} y_1 + \frac{\tau}{2}\|y_1 - \bar{y}_1^{k-1}\|^2$

  $\bar{y}_1^k = \sum_\omega \pi^\omega y_1^{\omega,k}$

  $\lambda^{\omega,k} = \lambda^{\omega,k-1} + \tau\,(y_1^{\omega,k} - \bar{y}_1^k)$ // fixed step τ

  $r^k = \left(\sum_\omega \pi^\omega \|y_1^{\omega,k} - \bar{y}_1^k\|^2\right)^{1/2}$ // the disagreement

  if $r^k \le \varepsilon$ **then** stop and return $\bar{y}_1^k$, λ^k

Implementations also test the motion of the average, $\tau\|\bar{y}_1^k - \bar{y}_1^{k-1}\| \le \varepsilon$.

Two iterations of progressive hedging

averaging and multiplier update

$$\bar{y}_1^0 = \sum_{\omega \in \Omega} \pi^\omega y_1^{\omega,0}$$

$$\lambda^{\omega,0} = \tau\left(y_1^{\omega,0} - \bar{y}_1^0\right)$$

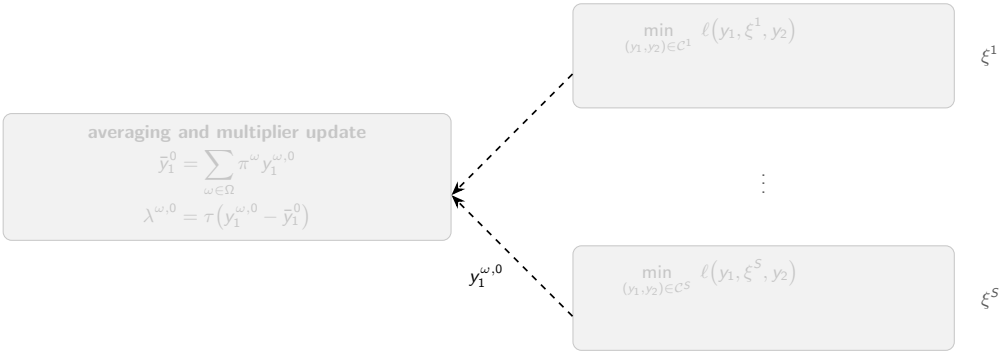
$$\min_{(y_1,y_2)\in\mathcal{C}^1} \ \ell\left(y_1, \xi^1, y_2\right) \qquad \xi^1$$

$$\vdots$$

$$\min_{(y_1,y_2)\in\mathcal{C}^S} \ \ell\left(y_1, \xi^S, y_2\right) \qquad \xi^S$$

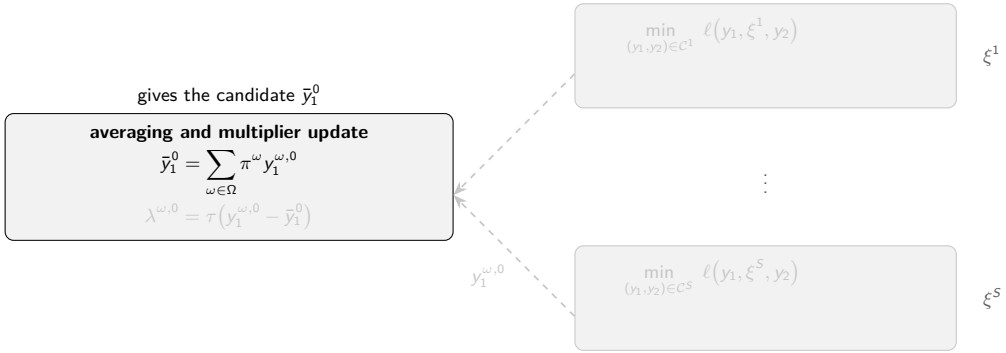
Start. Each scenario is solved alone, as the algorithm starts it: no multiplier, no proximal term, nothing from the other scenarios.

Two iterations of progressive hedging



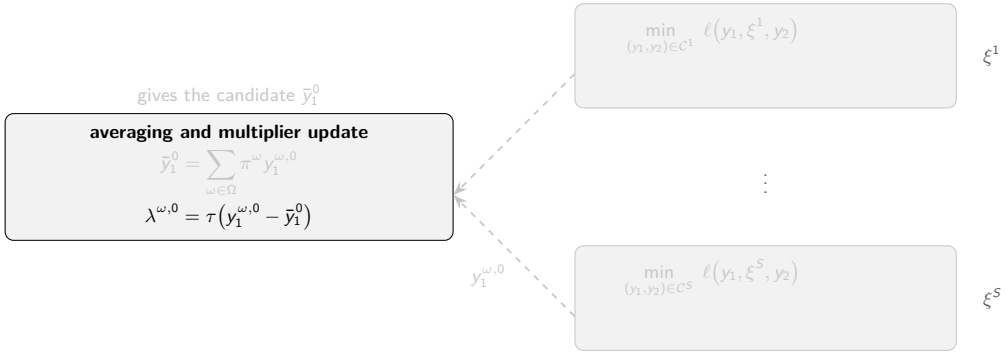
Copies. Scenario ω returns its first-stage copy $y_1^{\omega,0}$: the copies disagree.

Two iterations of progressive hedging



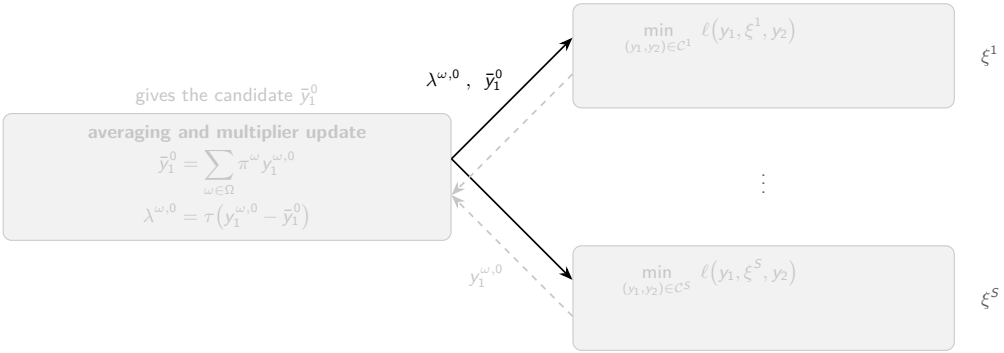
The average. The master forms $\bar{y}_1^0$, the weighted average of the copies, and that is the first candidate decision.

Two iterations of progressive hedging



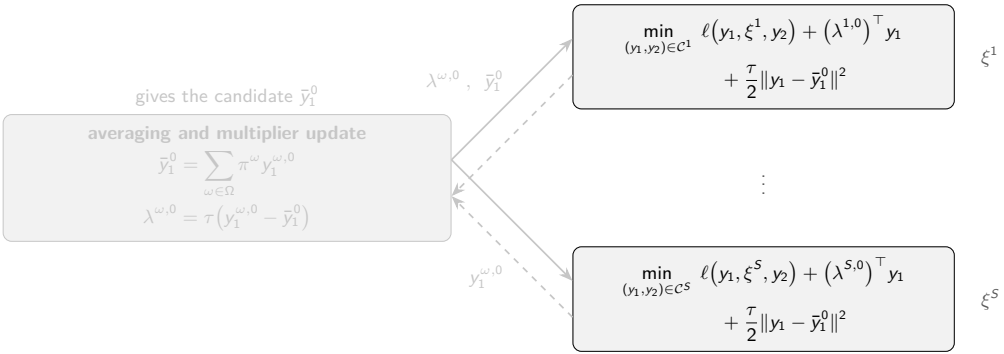
Multipliers. Each copy's multiplier is set to τ times its own gap to the average, so $\lambda^{\omega,0}$ is centred.

Two iterations of progressive hedging



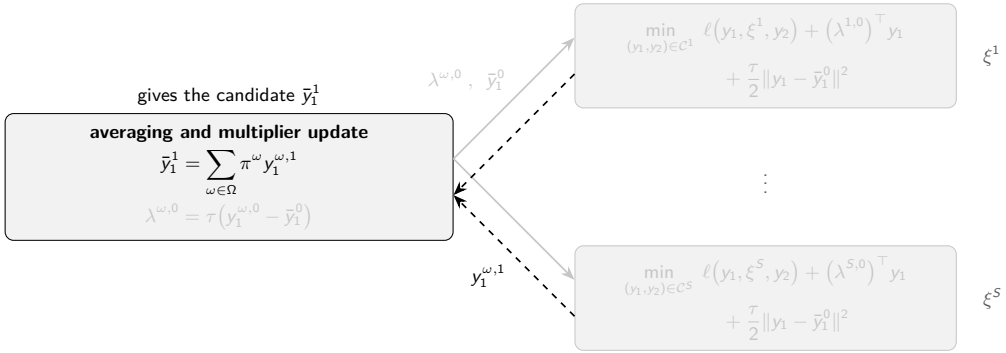
Back to the scenarios. Each scenario gets its multiplier $\lambda^{\omega,0}$ and the average $\bar{y}_1^0$ they are all pulled towards.

Two iterations of progressive hedging



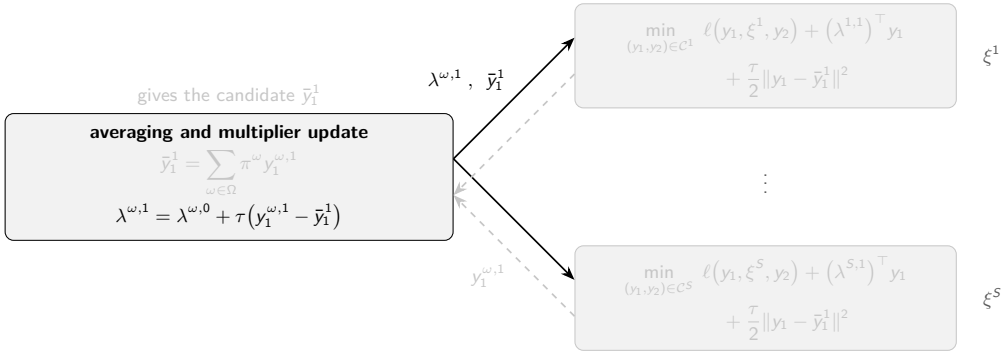
Iteration 1. Each problem now carries the linear term in λ and the proximal term towards $\bar{y}_1^0$; solved again, still alone.

Two iterations of progressive hedging



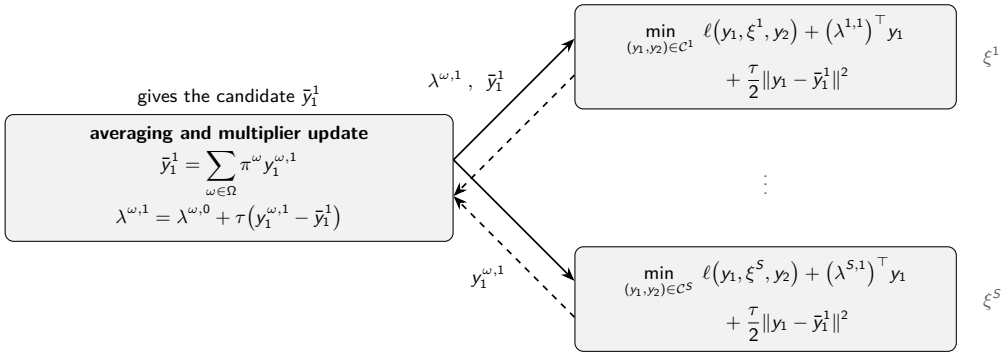
New copies. The solutions $y_1^{\omega,1}$ come back, and the master averages them into $\bar{y}_1^1$.

Two iterations of progressive hedging



New multipliers. Moved by the new gaps, with the fixed step τ, and sent back with the new average: ready for iteration 2.

Two iterations of progressive hedging



The whole iteration. S problems solved alone, each with its multiplier and its proximal term; the master averages the copies and moves every multiplier by its gap.

Convergence of progressive hedging



Theorem (Convergence of progressive hedging)

Assume (CVX), $\tau > 0$, and that every subproblem, the initial one included, attains its minimum.

(i) *If $r^k = s^k = 0$ at some k, then $\bar{y}_1^k$ is primal-optimal and λ^k is an optimal multiplier.*

(ii) *If v^{SP} is finite and attained, and $v^{\mathrm{D}}(\lambda^\star) = v^{\mathrm{SP}}$ for some $\lambda^\star \in \Lambda$, then $r^k \to 0$, $s^k \to 0$, the iterates are bounded, and*

$$\sum_\omega \pi^\omega f^\omega\big(y_1^{\omega,k}\big) \;\longrightarrow\; v^{\mathrm{SP}} .$$

(iii) *If moreover every f^ω is lsc, there are an optimal $\bar{y}_1^\infty$ and a price of information λ^∞ with $y_1^{\omega,k} \to \bar{y}_1^\infty$ for every ω and $\lambda^k \to \lambda^\infty$.*

Hypothesis (ii) holds under the strong-duality qualification, or for linear scenario problems.

Proof idea. The energy $\mathcal{E}^k := \frac{1}{\tau}\sum_\omega \pi^\omega \|\lambda^{\omega,k} - \lambda^{\star,\omega}\|^2 + \tau\|\bar{y}_1^k - y_1^\star\|^2$ decreases at every iteration by at least $\tau(r^k)^2 + (s^k)^2/\tau$.

Convergence of progressive hedging



Theorem (Convergence of progressive hedging)

Assume (CVX), $\tau > 0$, and that every subproblem, the initial one included, attains its minimum.

(i) *If $r^k = s^k = 0$ at some k, then $\bar{y}_1^k$ is primal-optimal and λ^k is an optimal multiplier.*

(ii) *If v^{SP} is finite and attained, and $v^{\mathrm{D}}(\lambda^\star) = v^{\mathrm{SP}}$ for some $\lambda^\star \in \Lambda$, then $r^k \to 0$, $s^k \to 0$, the iterates are bounded, and*

$$\sum_\omega \pi^\omega f^\omega\big(y_1^{\omega,k}\big) \;\longrightarrow\; v^{\mathrm{SP}} .$$

(iii) *If moreover every f^ω is lsc, there are an optimal $\bar{y}_1^\infty$ and a price of information λ^∞ with $y_1^{\omega,k} \to \bar{y}_1^\infty$ for every ω and $\lambda^k \to \lambda^\infty$.*

Hypothesis (ii) holds under the strong-duality qualification, or for linear scenario problems.

Proof idea. The energy $\mathcal{E}^k := \frac{1}{\tau}\sum_\omega \pi^\omega \|\lambda^{\omega,k} - \lambda^{\star,\omega}\|^2 + \tau\|\bar{y}_1^k - y_1^\star\|^2$ decreases at every iteration by at least $\tau(r^k)^2 + (s^k)^2/\tau$.

What the theorem does and does not say



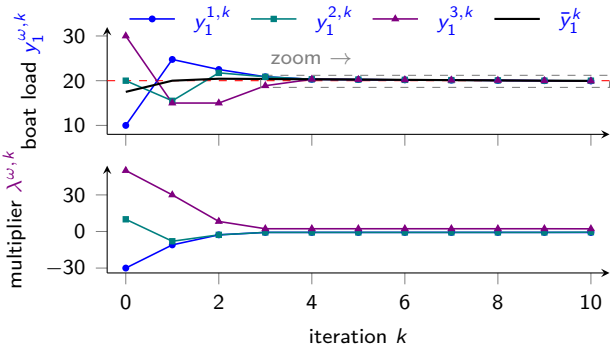
- The energy decreases whatever $\tau > 0$: no step-size rule.
- But τ strongly affects the rate, and choosing it is the main difficulty in practice:
  - large τ: every copy is pulled to the frozen average, so the copies agree quickly, on a decision that may be poor, and the average then moves slowly: many iterations to reach a good one;
  - small τ: the average moves fast, the scenarios disagree for many iterations.
- The cost of the copies in (ii) is neither the cost of a decision nor a bound.
- The convergence is that of the first stage only: the proximal term bears on y_1 alone, and $y_2^{\omega,k}$ may keep moving. The original algorithm of Rockafellar and Wets (1991) also penalizes the recourse, and its whole solution converges; implementations (e.g. PySP) keep the first-stage term only.
- $f^\omega(y_1) = \min\{\ell(y_1, \xi^\omega, y_2) \mid (y_1, y_2) \in \mathcal{C}^\omega\}$: that minimization keeps convexity, not lower semicontinuity, which is the one hypothesis it does not supply; it holds if $\mathcal{C}^\omega$ is compact and ℓ lower semicontinuous, and in the linear case.

What the theorem does and does not say

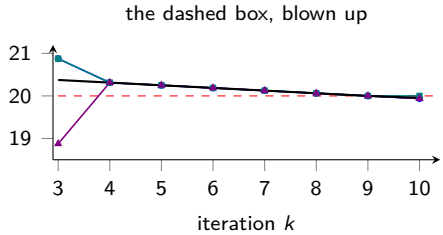


- The energy decreases whatever $\tau > 0$: no step-size rule.
- But τ strongly affects the rate, and choosing it is the main difficulty in practice:
  - large τ: every copy is pulled to the frozen average, so the copies agree quickly, on a decision that may be poor, and the average then moves slowly: many iterations to reach a good one;
  - small τ: the average moves fast, the scenarios disagree for many iterations.
- The cost of the copies in (ii) is neither the cost of a decision nor a bound.
- The convergence is that of the first stage only: the proximal term bears on y_1 alone, and $y_2^{\omega,k}$ may keep moving. The original algorithm of Rockafellar and Wets (1991) also penalizes the recourse, and its whole solution converges; implementations (e.g. PySP) keep the first-stage term only.
- $f^\omega(y_1) = \min\{\ell(y_1, \xi^\omega, y_2) \mid (y_1, y_2) \in \mathcal{C}^\omega\}$: that minimization keeps convexity, not lower semicontinuity, which is the one hypothesis it does not supply; it holds if $\mathcal{C}^\omega$ is compact and ℓ lower semicontinuous, and in the linear case.

Example: progressive hedging on the shipping problem



Answer $y_1^\star = 20$ (red), $v^{\mathrm{SP}} = 27.5$, $\lambda^\star = (-1, 0, 2)$. The copies reach the answer while the multipliers stall at $(-0.75, -0.75, 2.25)$, short of $\lambda^\star$.



The red line is still $y_1^\star = 20$. At $k = 3$ the copies disagree, two at 20.875 and one at 18.875. From $k = 4$ to 9 they coincide: with λ^k frozen, each scenario's slope plus its multiplier is the same, so every copy sits the same distance below the frozen average.

At $k = 10$ they part again, by 0.0625, too little to see here: 20 is the kink of scenario 2, so it stays while the others go on.

Without convexity



- Weak duality and the concavity of v^{D} never used (CVX): with mixed-integer scenario problems every $v^{\mathrm{D}}(\lambda^k)$ is still a valid lower bound.
- The gap left comes from convexifying the scenarios *separately*: for linear ℓ and bounded mixed-integer $\mathcal{C}^\omega$,

$$\max_{\Lambda} v^{\mathrm{D}} = \text{value of the scenario form with each } \mathcal{C}^\omega \text{ replaced by } \operatorname{conv} \mathcal{C}^\omega \ .$$

- Progressive hedging loses its guarantees but still runs: a heuristic for stochastic MIPs (Watson and Woodruff 2011). The iterates may cycle, the copies may never agree, and $\bar{y}_1^k$ may not be integer.

Without convexity



- Weak duality and the concavity of v^{D} never used (CVX): with mixed-integer scenario problems every $v^{\mathrm{D}}(\lambda^k)$ is still a valid lower bound.
- The gap left comes from convexifying the scenarios *separately*: for linear ℓ and bounded mixed-integer $\mathcal{C}^\omega$,

$$\max_{\Lambda} v^{\mathrm{D}} = \text{value of the scenario form with each } \mathcal{C}^\omega \text{ replaced by } \operatorname{conv} \mathcal{C}^\omega \ .$$

- Progressive hedging loses its guarantees but still runs: a heuristic for stochastic MIPs (Watson and Woodruff 2011). The iterates may cycle, the copies may never agree, and $\bar{y}_1^k$ may not be integer.

Without convexity



- Weak duality and the concavity of v^{D} never used (CVX): with mixed-integer scenario problems every $v^{\mathrm{D}}(\lambda^k)$ is still a valid lower bound.
- The gap left comes from convexifying the scenarios *separately*: for linear ℓ and bounded mixed-integer $\mathcal{C}^\omega$,

$$\max_{\Lambda} v^{\mathrm{D}} = \text{value of the scenario form with each } \mathcal{C}^\omega \text{ replaced by } \operatorname{conv} \mathcal{C}^\omega \ .$$

- Progressive hedging loses its guarantees but still runs: a heuristic for stochastic MIPs (Watson and Woodruff 2011). The iterates may cycle, the copies may never agree, and $\bar{y}_1^k$ may not be integer.

Presentation Outline

1. Scenario decomposition
   - Dual decomposition
   - Progressive hedging
   - Bounds
2. Stage decomposition
   - The L-shaped method
   - ✠ The adaptive partition method
3. Comparing the methods

Bounds in progressive hedging

The proximal term gives a converging decision, and takes away the lower bound that dual decomposition got with no extra solve. Each bound now takes S extra solves, so compute it when you want it.

- **Lower bound**: $v^{\mathrm{D}}(\lambda^k) \leq v^{\mathrm{SP}}$, valid since $\lambda^k \in \Lambda$. It costs S new solves: the scenario problems at λ^k *without* the quadratic term.
- **Upper bound**: UB^k, the cost of $\bar{y}_1^k$ with its optimal recourse in each scenario.

Bounds in progressive hedging

The proximal term gives a converging decision, and takes away the lower bound that dual decomposition got with no extra solve. Each bound now takes S extra solves, so compute it when you want it.

- **Lower bound**: $v^{\mathrm{D}}(\lambda^k) \leq v^{\mathrm{SP}}$, valid since $\lambda^k \in \Lambda$. It costs S new solves: the scenario problems at λ^k *without* the quadratic term.
- **Upper bound**: UB^k, the cost of $\bar{y}_1^k$ with its optimal recourse in each scenario.

Scenario decomposition reuses your deterministic solver



Takeaway

A scenario subproblem is the deterministic problem of that scenario with one linear term added (dual decomposition), or a linear and a convex quadratic one (progressive hedging).

- A solver for the deterministic model is all dual decomposition needs, and all progressive hedging needs if it accepts the quadratic term.
- The S subproblems of an iteration are independent.
- Both algorithms ask only for convexity, never for linearity; what coordinates the scenarios is one multiplier each.

Presentation Outline

1. Scenario decomposition
   - Dual decomposition
   - Progressive hedging
   - Bounds
2. Stage decomposition
   - The L-shaped method
   - The adaptive partition method
3. Comparing the methods

Linear two-stage stochastic program

Under (LIN): linear costs, affine constraints, second-stage matrix of full row rank.

$$\begin{aligned} \min_{y_1} \quad & c_1^\top y_1 + V_2(y_1) \\ \text{s.t.} \quad & C_1\, y_1 = d_1, \quad y_1 \geq 0\,, \end{aligned}$$

where, for each realization ξ,

$$\begin{aligned} \dot{V}_2(y_1, \xi) := \min_{y_2 \geq 0} \quad & c_2(\xi)^\top y_2 \\ \text{s.t.} \quad & C_2(\xi)\, y_2 = d_2(\xi) - B_2(\xi)\, y_1\,, \end{aligned}$$

and
$V_2(y_1) := \mathbb{E}\big[\dot{V}_2(y_1, \boldsymbol{\xi})\big] = \sum_{\omega \in \Omega} \pi^\omega\, \dot{V}_2^\omega(y_1)$.

The extensive formulation is

$$\begin{aligned} \min \quad & c_1^\top y_1 + \sum_{\omega \in \Omega} \pi^\omega\, (c_2^\omega)^\top y_2^\omega \\ \text{s.t.} \quad & C_1 y_1 = d_1, \quad y_1 \geq 0 \\ & B_2^\omega y_1 + C_2^\omega y_2^\omega = d_2^\omega,\ y_2^\omega \geq 0,\ \forall \omega\,. \end{aligned}$$

Note that Birge and Louveaux write (q, T, W, h) for (c_2, B_2, C_2, d_2), T being the technology and W the recourse matrix, and Q, $\mathcal{Q}$ for $\dot{V}_2$, V_2.

Linear two-stage stochastic program

Under (LIN): linear costs, affine constraints, second-stage matrix of full row rank.

$$\begin{aligned} \min_{y_1} \quad & c_1^\top y_1 + V_2(y_1) \\ \text{s.t.} \quad & C_1\, y_1 = d_1, \quad y_1 \geq 0\ , \end{aligned}$$

where, for each realization ξ,

$$\begin{aligned} \dot{V}_2(y_1, \xi) := \min_{y_2 \geq 0} \quad & c_2(\xi)^\top y_2 \\ \text{s.t.} \quad & C_2(\xi)\, y_2 = d_2(\xi) - B_2(\xi)\, y_1\ , \end{aligned}$$

and
$V_2(y_1) := \mathbb{E}\big[\dot{V}_2(y_1, \boldsymbol{\xi})\big] = \sum_{\omega \in \Omega} \pi^\omega\, \dot{V}_2^\omega(y_1)$.

The extensive formulation is

$$\begin{aligned} \min \quad & c_1^\top y_1 + \sum_{\omega \in \Omega} \pi^\omega\, (c_2^\omega)^\top y_2^\omega \\ \text{s.t.} \quad & C_1 y_1 = d_1, \quad y_1 \geq 0 \\ & B_2^\omega y_1 + C_2^\omega y_2^\omega = d_2^\omega,\ y_2^\omega \geq 0,\ \forall \omega\ . \end{aligned}$$

Note that Birge and Louveaux write (q, T, W, h) for (c_2, B_2, C_2, d_2), T being the technology and W the recourse matrix, and Q, $\mathcal{Q}$ for $\dot{V}_2$, V_2.

Stage decomposition: the idea

$$\min_{y_1 \in \mathcal{Y}_1} \quad c_1^\top y_1 + V_2(y_1)$$

- V_2 is polyhedral (convex, piecewise affine) under (FSN), as soon as the second-stage duals exist;
- evaluated at one point by S linear programs, with a subgradient.

Cutting planes: replace V_2 by a polyhedral model below it (a collection of cuts), solve, evaluate V_2 at the decision returned, add a cut exact there. The second-stage duals give the cut.

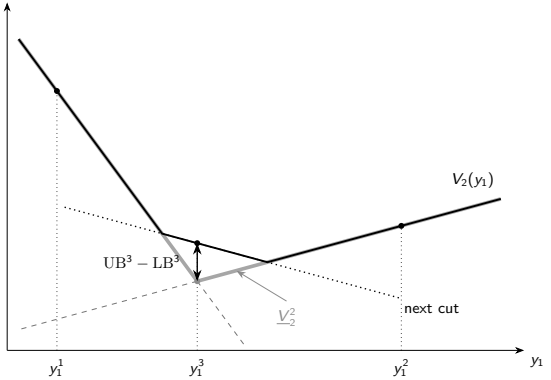
V_2 (black), the cuts at y_1^1 and y_1^2 and their model $\underline{V}_2^2$ (grey); the third master returns y_1^3, where the model lies below V_2 by $\mathrm{UB}^3 - \mathrm{LB}^3$; the cut there (dotted) closes the distance.

Stage decomposition: the idea

$$\min_{y_1 \in \mathcal{Y}_1} \quad c_1^\top y_1 + V_2(y_1)$$

- V_2 is polyhedral (convex, piecewise affine) under (FSN), as soon as the second-stage duals exist;
- evaluated at one point by S linear programs, with a subgradient.

Cutting planes: replace V_2 by a polyhedral model below it (a collection of cuts), solve, evaluate V_2 at the decision returned, add a cut exact there. The second-stage duals give the cut.



V_2 (black), the cuts at y_1^1 and y_1^2 and their model $\underline{V}_2^2$ (grey); the third master returns y_1^3, where the model lies below V_2 by $\mathrm{UB}^3 - \mathrm{LB}^3$; the cut there (dotted) closes the distance.

Second-stage duality

Write $h^\omega(y_1) := d_2^\omega - B_2^\omega y_1$, the right-hand side that y_1 leaves to scenario ω. Under relatively complete recourse, strong duality gives

$$\dot{V}_2^\omega(y_1) = \min_{y_2 \geq 0} \left\{ \left(c_2^\omega\right)^\top y_2 \mid C_2^\omega y_2 = h^\omega(y_1) \right\} = \max_{\lambda} \left\{ \lambda^\top h^\omega(y_1) \mid \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega \right\}.$$

The decision y_1 enters the objective of that maximum, never its constraints:

$$\dot{V}_2^\omega(y_1^k) = \max_{\lambda} \ \lambda^\top h^\omega(y_1^k) \quad \text{s.t.} \ \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega \,,$$

solved at the current iterate by $\lambda^{\omega,k}$: $\dot{V}_2^\omega(y_1^k) = \lambda^{\omega,k\top} h^\omega(y_1^k)$;

$$\dot{V}_2^\omega(y_1) = \max_{\lambda} \ \lambda^\top h^\omega(y_1) \quad \text{s.t.} \ \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega \,,$$

at any other y_1.

In both case the maximum is over the same dual admissible set $D^\omega := \{\lambda \mid \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega\}$. So $\lambda^{\omega,k}$ is admissible on the right too, and:

$$\dot{V}_2^\omega(y_1) = \max_{\lambda \in D^\omega} \lambda^\top h^\omega(y_1) \ \geq \ \left(\lambda^{\omega,k}\right)^\top h^\omega(y_1) \qquad \forall y_1, \quad \text{with equality at } y_1^k \,.$$

Second-stage duality

Write $h^\omega(y_1) := d_2^\omega - B_2^\omega y_1$, the right-hand side that y_1 leaves to scenario ω. Under relatively complete recourse, strong duality gives

$$\dot{V}_2^\omega(y_1) = \min_{y_2 \geq 0} \left\{ \left(c_2^\omega\right)^\top y_2 \mid C_2^\omega y_2 = h^\omega(y_1) \right\} = \max_{\lambda} \left\{ \lambda^\top h^\omega(y_1) \mid \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega \right\}.$$

The decision y_1 enters the objective of that maximum, never its constraints:

$$\dot{V}_2^\omega(y_1^k) = \max_{\lambda} \ \lambda^\top h^\omega(y_1^k) \quad \text{s.t.} \ \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega \ ,$$

solved at the current iterate by $\lambda^{\omega,k}$:
$\dot{V}_2^\omega(y_1^k) = \lambda^{\omega,k\top} h^\omega(y_1^k)$;

$$\dot{V}_2^\omega(y_1) = \max_{\lambda} \ \lambda^\top h^\omega(y_1) \quad \text{s.t.} \ \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega \ ,$$

at any other y_1.

In both case the maximum is over the same dual admissible set $D^\omega := \{\lambda \mid \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega\}$. So $\lambda^{\omega,k}$ is admissible on the right too, and:

$$\dot{V}_2^\omega(y_1) = \max_{\lambda \in D^\omega} \lambda^\top h^\omega(y_1) \ \geq \ \left(\lambda^{\omega,k}\right)^\top h^\omega(y_1) \qquad \forall y_1, \quad \text{with equality at } y_1^k \ .$$

Second-stage duality

Write $h^\omega(y_1) := d_2^\omega - B_2^\omega y_1$, the right-hand side that y_1 leaves to scenario ω. Under relatively complete recourse, strong duality gives

$$\dot{V}_2^\omega(y_1) = \min_{y_2 \geq 0} \left\{ \left(c_2^\omega\right)^\top y_2 \mid C_2^\omega y_2 = h^\omega(y_1) \right\} = \max_{\lambda} \left\{ \lambda^\top h^\omega(y_1) \mid \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega \right\}.$$

The decision y_1 enters the objective of that maximum, never its constraints:

$$\dot{V}_2^\omega(y_1^k) = \max_{\lambda} \ \lambda^\top h^\omega(y_1^k) \quad \text{s.t.} \ \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega \,,$$

solved at the current iterate by $\lambda^{\omega,k}$: $\dot{V}_2^\omega(y_1^k) = \lambda^{\omega,k\top} h^\omega(y_1^k)$;

$$\dot{V}_2^\omega(y_1) = \max_{\lambda} \ \lambda^\top h^\omega(y_1) \quad \text{s.t.} \ \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega \,,$$

at any other y_1.

In both case the maximum is over the same dual admissible set $D^\omega := \{\lambda \mid \left(C_2^\omega\right)^\top \lambda \leq c_2^\omega\}$. So $\lambda^{\omega,k}$ is admissible on the right too, and:

$$\dot{V}_2^\omega(y_1) = \max_{\lambda \in D^\omega} \lambda^\top h^\omega(y_1) \ \geq \ \left(\lambda^{\omega,k}\right)^\top h^\omega(y_1) \qquad \forall y_1, \quad \text{with equality at } y_1^k \,.$$

Optimality cut

Proposition

Suppose D^ω nonempty. Let $y_1^k \in \mathcal{Y}_1$ be a point where scenario ω is feasible, and $\lambda^{\omega,k}$ an optimal dual solution there. Set

$$\alpha^{\omega,k} := -(B_2^\omega)^\top \lambda^{\omega,k} \qquad \text{and} \qquad \beta^{\omega,k} := \dot{V}_2^\omega(y_1^k)\,.$$

Then $\alpha^{\omega,k}$ is a subgradient of $\dot{V}_2^\omega$ at y_1^k:

$$\dot{V}_2^\omega(y_1) \;\geq\; (\alpha^{\omega,k})^\top (y_1 - y_1^k) + \beta^{\omega,k} \quad \forall y_1, \qquad \text{with equality at } y_1^k\,.$$

The cut applies the dual $\lambda^{\omega,k}$ to how y_1 moves the right-hand side.

Optimality cut

Proposition

Suppose D^ω nonempty. Let $y_1^k \in \mathcal{Y}_1$ be a point where scenario ω is feasible, and $\lambda^{\omega,k}$ an optimal dual solution there. Set

$$\alpha^{\omega,k} := -(B_2^\omega)^\top \lambda^{\omega,k} \qquad \text{and} \qquad \beta^{\omega,k} := \dot{V}_2^\omega(y_1^k)\,.$$

Then $\alpha^{\omega,k}$ is a subgradient of $\dot{V}_2^\omega$ at y_1^k:

$$\dot{V}_2^\omega(y_1) \;\geq\; (\alpha^{\omega,k})^\top(y_1 - y_1^k) + \beta^{\omega,k} \quad \forall y_1, \qquad \text{with equality at } y_1^k\,.$$

The cut applies the dual $\lambda^{\omega,k}$ to how y_1 moves the right-hand side.

The master problem



The first stage, with the recourse values it cannot evaluate:

$$\min_{y_1 \in \mathcal{Y}_1} \quad c_1^\top y_1 + \sum_{\omega \in \Omega} \pi^\omega \, \dot{V}_2^\omega(y_1)$$

$$\text{s.t.} \quad \theta^\omega \geq$$

Step one: an epigraph variable θ^ω per scenario, held above the recourse cost. Since $\pi^\omega > 0$ and the objective is minimized, each θ^ω ends on $\dot{V}_2^\omega(y_1)$: the same decisions solve both problems.

Step two: $\dot{V}_2^\omega$ is polyhedral, so $\theta^\omega \geq \dot{V}_2^\omega(y_1)$ is finitely many affine constraints, and the cut of iteration i is one of them. Each cut is valid, so each of them relaxes $\theta^\omega \geq \dot{V}_2^\omega(y_1)$, and the master's value is a lower bound on v^{SP}.

Note that without relatively complete recourse the master also carries feasibility cuts, constraints on y_1 alone.

The master problem



The first stage, with the recourse values it cannot evaluate:

$$\begin{aligned} \min_{y_1 \in \mathcal{Y}_1,\, \theta} \quad & c_1^\top y_1 + \sum_{\omega \in \Omega} \pi^\omega \, \theta^\omega \\ \text{s.t.} \quad & \theta^\omega \;\geq\; \dot{V}_2^\omega(y_1), \quad \forall \omega \in \Omega \,. \end{aligned}$$

Step one: an epigraph variable θ^ω per scenario, held above the recourse cost. Since $\pi^\omega > 0$ and the objective is minimized, each θ^ω ends on $\dot{V}_2^\omega(y_1)$: the same decisions solve both problems.

Step two: $\dot{V}_2^\omega$ is polyhedral, so $\theta^\omega \geq \dot{V}_2^\omega(y_1)$ is finitely many affine constraints, and the cut of iteration i is one of them. Each cut is valid, so each of them relaxes $\theta^\omega \geq \dot{V}_2^\omega(y_1)$, and the master's value is a lower bound on v^{SP}.

Note that without relatively complete recourse the master also carries feasibility cuts, constraints on y_1 alone.

The master problem



The first stage, with the recourse values it cannot evaluate:

$$\begin{aligned} \min_{y_1 \in \mathcal{Y}_1,\, \theta} \quad & c_1^\top y_1 + \sum_{\omega \in \Omega} \pi^\omega \, \theta^\omega \\ \text{s.t.} \quad & \theta^\omega \;\geq\; \left(\alpha^{\omega,i}\right)^\top \left(y_1 - y_1^i\right) + \beta^{\omega,i}, \;\; \forall \omega \in \Omega, \; \forall i \in [k-1] \, . \end{aligned}$$

Step one: an epigraph variable θ^ω per scenario, held above the recourse cost. Since $\pi^\omega > 0$ and the objective is minimized, each θ^ω ends on $\dot{V}_2^\omega(y_1)$: the same decisions solve both problems.

Step two: $\dot{V}_2^\omega$ is polyhedral, so $\theta^\omega \geq \dot{V}_2^\omega(y_1)$ is finitely many affine constraints, and the cut of iteration i is one of them. Each cut is valid, so each of them relaxes $\theta^\omega \geq \dot{V}_2^\omega(y_1)$, and the master's value is a lower bound on v^{SP}.

Note that without relatively complete recourse the master also carries feasibility cuts, constraints on y_1 alone.

The multi-cut L-shaped method

Start: $y_1^0 \in \mathcal{Y}_1$ where every scenario is feasible, its S cuts, UB its cost, $y_1^{\text{best}} = y_1^0$

for $k = 1, 2, \dots$ **do**

solve the master, with solution $\left(y_1^k, (\theta^{\omega,k})_\omega\right)$:

$$\text{LB}^k = \min_{y_1 \in \mathcal{Y}_1,\, \theta} \left\{ c_1^\top y_1 + \sum_\omega \pi^\omega \theta^\omega \;\middle|\; \theta^\omega \geq \left(\alpha^{\omega,i}\right)^\top \left(y_1 - y_1^i\right) + \beta^{\omega,i},\ \ \forall \omega,\ \forall i < k \right\}$$

foreach *scenario* $\omega \in \Omega$ **do** `// in parallel`

solve the second-stage LP at y_1^k

if *infeasible* **then**

ray r: add the feasibility cut $r^\top \left(d_2^\omega - B_2^\omega y_1\right) \leq 0$ (later)

else

value $\dot{V}_2^\omega(y_1^k)$, dual $\lambda^{\omega,k}$: add the optimality cut $\theta^\omega \geq (\alpha^{\omega,k})^\top (y_1 - y_1^k) + \beta^{\omega,k}$, where $\alpha^{\omega,k} = -(B_2^\omega)^\top \lambda^{\omega,k}$ and $\beta^{\omega,k} = \dot{V}_2^\omega(y_1^k)$

$\text{UB}^k = c_1^\top y_1^k + \sum_\omega \pi^\omega \dot{V}_2^\omega(y_1^k)$; if $\text{UB}^k < \text{UB}$, update UB and y_1^{best}

if $\text{UB} - \text{LB}^k \leq \varepsilon$ **then** stop

One iteration: one master solve and S subproblems that differ only in a right-hand side; each iteration adds S rows to the master and no variable.

The multi-cut L-shaped method

Start: $y_1^0 \in \mathcal{Y}_1$ where every scenario is feasible, its S cuts, UB its cost, $y_1^{\text{best}} = y_1^0$

for $k = 1, 2, \dots$ **do**

solve the master, with solution $\left(y_1^k, (\theta^{\omega,k})_\omega\right)$:

$$\text{LB}^k = \min_{y_1 \in \mathcal{Y}_1,\, \theta} \left\{ c_1^\top y_1 + \sum_\omega \pi^\omega \theta^\omega \;\middle|\; \theta^\omega \geq \left(\alpha^{\omega,i}\right)^\top \left(y_1 - y_1^i\right) + \beta^{\omega,i},\ \ \forall \omega,\ \forall i < k \right\}$$

foreach *scenario* $\omega \in \Omega$ **do** `// in parallel`

solve the second-stage LP at y_1^k

if *infeasible* **then**

ray r: add the feasibility cut $r^\top \left(d_2^\omega - B_2^\omega y_1\right) \leq 0$ (later)

else

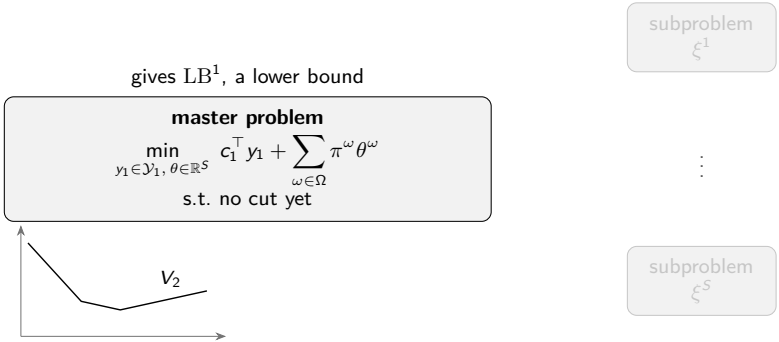
value $\dot{V}_2^\omega(y_1^k)$, dual $\lambda^{\omega,k}$: add the optimality cut $\theta^\omega \geq (\alpha^{\omega,k})^\top (y_1 - y_1^k) + \beta^{\omega,k}$, where $\alpha^{\omega,k} = -(B_2^\omega)^\top \lambda^{\omega,k}$ and $\beta^{\omega,k} = \dot{V}_2^\omega(y_1^k)$

$\text{UB}^k = c_1^\top y_1^k + \sum_\omega \pi^\omega \dot{V}_2^\omega(y_1^k)$; if $\text{UB}^k < \text{UB}$, update UB and y_1^{best}

if $\text{UB} - \text{LB}^k \leq \varepsilon$ **then** stop

One iteration: one master solve and S subproblems that differ only in a right-hand side; each iteration adds S rows to the master and no variable.

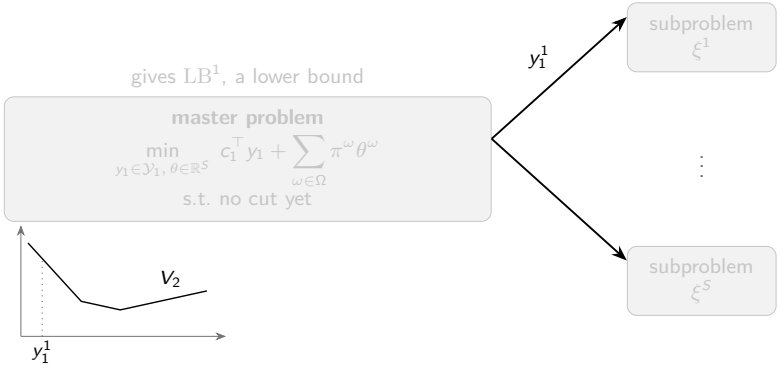
Two iterations of the L-shaped method



The master. With no cut yet it solves in the first stage alone: its value is the lower bound LB^1, its solution the candidate y_1^1.

Note that the master grows by S rows per iteration, and that its variables are the same at every one.

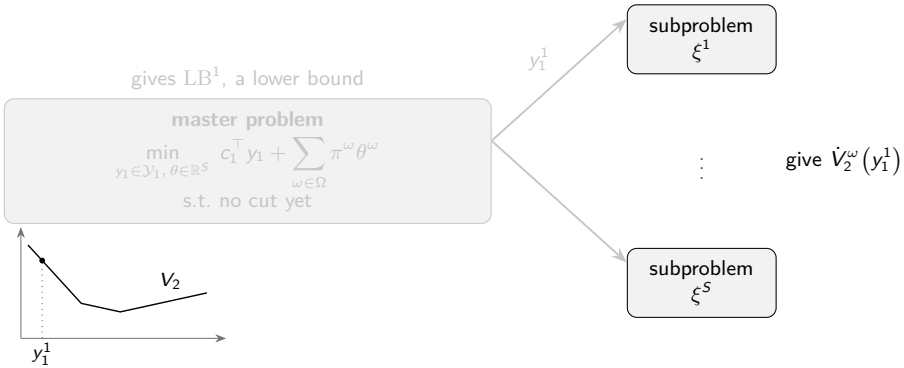
Two iterations of the L-shaped method



The candidate out. Every scenario is handed the same y_1^1, and holds it fixed.

Note that the master grows by S rows per iteration, and that its variables are the same at every one.

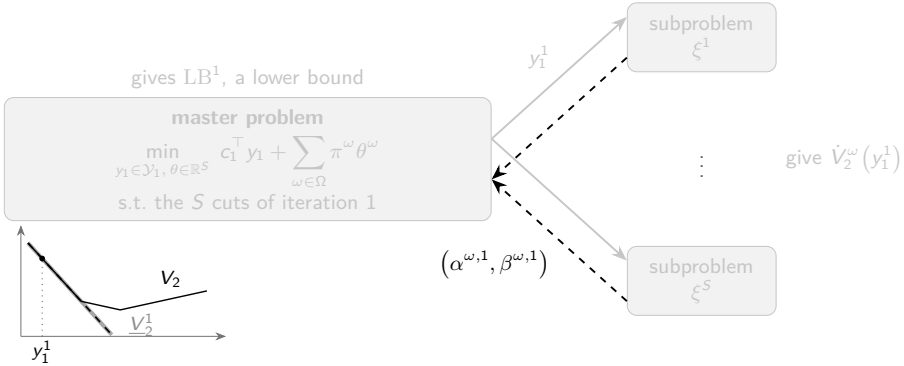
Two iterations of the L-shaped method



The subproblems. Each solves its second-stage program at y_1^1, in parallel: a recourse value, whose average gives an upper bound, and an optimal dual.

Note that the master grows by S rows per iteration, and that its variables are the same at every one.

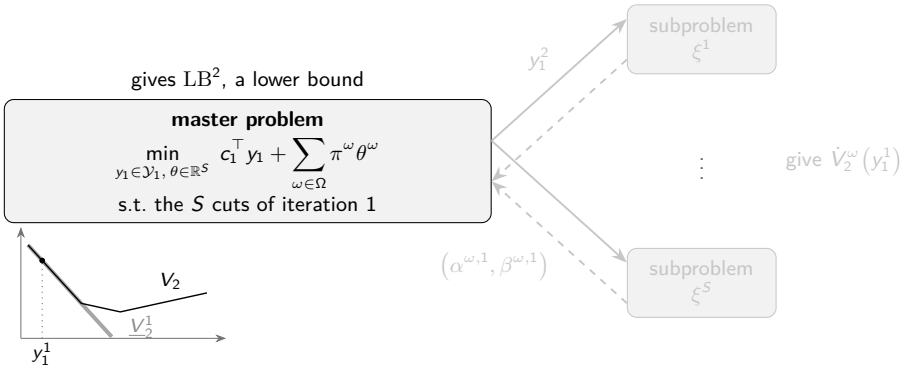
Two iterations of the L-shaped method



The cuts come back. One cut per scenario, exact at y_1^1: the master's model of V_2 gains S rows and rises.

Note that the master grows by S rows per iteration, and that its variables are the same at every one.

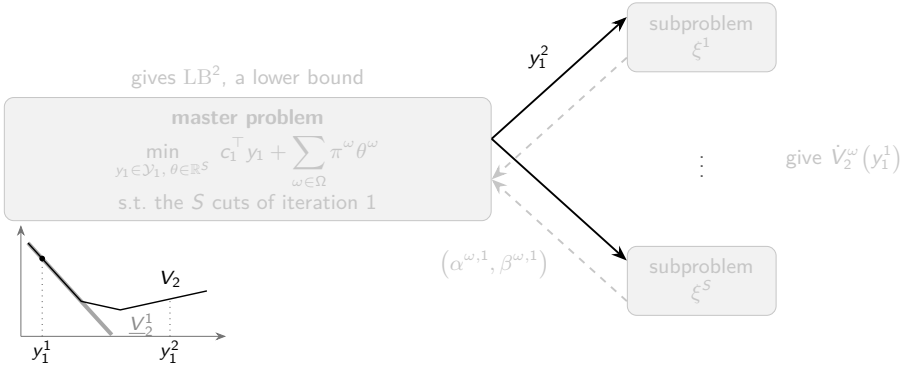
Two iterations of the L-shaped method



Iteration 2. The master, now holding those cuts, solves again: a higher LB^2, and a new candidate y_1^2, further right, where its model is lower.

Note that the master grows by S rows per iteration, and that its variables are the same at every one.

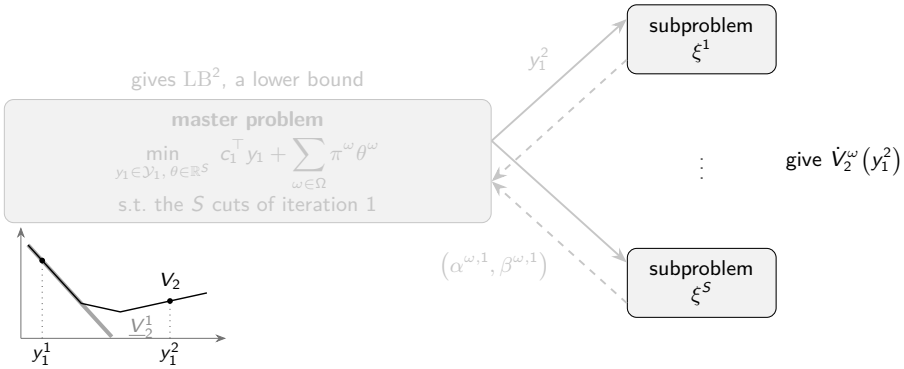
Two iterations of the L-shaped method



The candidate out. The same exchange, at the new point y_1^2.

Note that the master grows by S rows per iteration, and that its variables are the same at every one.

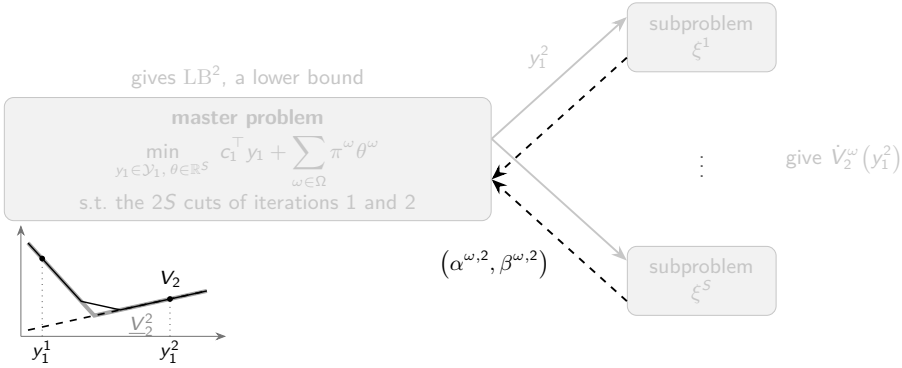
Two iterations of the L-shaped method



The subproblems. Solved again, at y_1^2 this time; nothing of the first iteration is reused.

Note that the master grows by S rows per iteration, and that its variables are the same at every one.

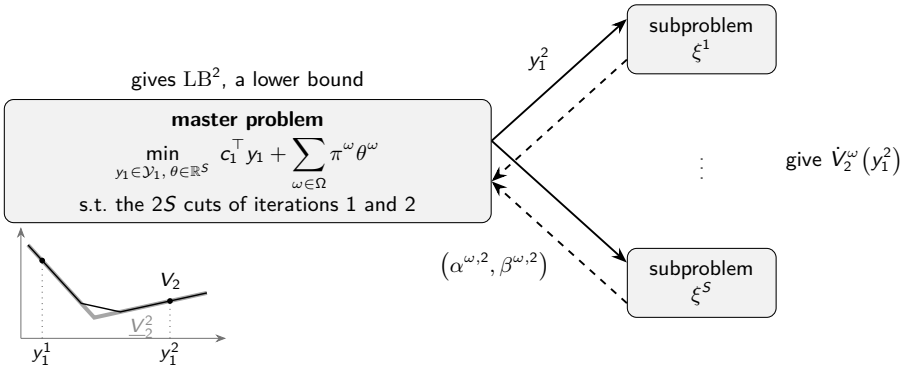
Two iterations of the L-shaped method



The cuts come back. S more rows, exact at y_1^2: the model rises where it was worst, and the two bounds close.

Note that the master grows by S rows per iteration, and that its variables are the same at every one.

Two iterations of the L-shaped method



The whole iteration. The master returns a candidate and a lower bound; every scenario is solved there and returns a cut, and the master's model rises.

Note that the master grows by S rows per iteration, and that its variables are the same at every one.

Bounds at every iteration



Proposition

Suppose every D^ω is nonempty and the master attains its minimum. At every iteration k:

- *the master value LB^k is a lower bound on v^{SP}, non-decreasing with k;*
- $\mathrm{UB}^k := c_1^\top y_1^k + \sum_{\omega\in\Omega} \pi^\omega \dot{V}_2^\omega(y_1^k)$ *is an upper bound;*
- *if $\mathrm{UB}^k = \mathrm{LB}^k$, then y_1^k is optimal.*

- Both bounds come from the iteration's own solves: the master's value, and the subproblem values behind the cuts.
- UB^k need not decrease: keep the best, $\mathrm{UB} = \min_{i\le k} \mathrm{UB}^i$, and return the incumbent.
- The pair is a certificate: $\mathrm{UB} - \mathrm{LB}^k$ bounds how far the decision returned is from optimal.

L-shaped is Benders' method on a two-stage linear program

One epigraph variable per scenario; each iteration adds one cut per scenario, built from a second-stage dual.

Bounds at every iteration



Proposition

Suppose every D^ω is nonempty and the master attains its minimum. At every iteration k:

- *the master value LB^k is a lower bound on v^{SP}, non-decreasing with k;*
- $\mathrm{UB}^k := c_1^\top y_1^k + \sum_{\omega\in\Omega} \pi^\omega \dot{V}_2^\omega(y_1^k)$ *is an upper bound;*
- *if $\mathrm{UB}^k = \mathrm{LB}^k$, then y_1^k is optimal.*

- Both bounds come from the iteration's own solves: the master's value, and the subproblem values behind the cuts.
- UB^k need not decrease: keep the best, $\mathrm{UB} = \min_{i\le k} \mathrm{UB}^i$, and return the incumbent.
- The pair is a certificate: $\mathrm{UB} - \mathrm{LB}^k$ bounds how far the decision returned is from optimal.

L-shaped is Benders' method on a two-stage linear program

One epigraph variable per scenario; each iteration adds one cut per scenario, built from a second-stage dual.

The single-cut variant

The cut of iteration k in scenario ω, valid at every y_1:

$$\dot{V}_2^{\omega}(y_1) \;\geq\; (\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k}$$

$$\sum_{\omega} \pi^{\omega}\, \dot{V}_2^{\omega}(y_1) \;\geq\; \sum_{\omega} \pi^{\omega}\Big((\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k}\Big)$$

$$V_2(y_1) \;\geq\; \Big(\underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \alpha^{\omega,k}}_{\alpha^k}\Big)^{\top}(y_1 - y_1^k) + \underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \beta^{\omega,k}}_{\beta^k = V_2(y_1^k)}$$

One θ for the expected value V_2, one constraint per iteration instead of S:

$$\min_{y_1 \in \mathcal{Y}_1,\, \theta} \Big\{ c_1^{\top} y_1 + \theta \mid \theta \geq (\alpha^i)^{\top}(y_1 - y_1^i) + \beta^i,\; i \in [k-1] \Big\}.$$

- From the same trial points and duals, the single-cut model lies below the multi-cut one.
- But the multi-cut master grows S times faster, and becomes the bottleneck when S is in the thousands.
- Single-cut is the default for large S; in between, one θ per group of scenarios.

The single-cut variant

The cut of iteration k in scenario ω, valid at every y_1:

$$\dot{V}_2^{\omega}(y_1) \;\geq\; (\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k}$$

$$\sum_{\omega} \pi^{\omega}\, \dot{V}_2^{\omega}(y_1) \;\geq\; \sum_{\omega} \pi^{\omega}\Big((\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k}\Big)$$

$$V_2(y_1) \;\geq\; \Big(\underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \alpha^{\omega,k}}_{\alpha^k}\Big)^{\top}(y_1 - y_1^k) + \underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \beta^{\omega,k}}_{\beta^k = V_2(y_1^k)}$$

One θ for the expected value V_2, one constraint per iteration instead of S:

$$\min_{y_1 \in \mathcal{Y}_1,\, \theta} \Big\{ c_1^{\top} y_1 + \theta \mid \theta \geq (\alpha^i)^{\top}(y_1 - y_1^i) + \beta^i,\ i \in [k-1] \Big\}.$$

- From the same trial points and duals, the single-cut model lies below the multi-cut one.
- But the multi-cut master grows S times faster, and becomes the bottleneck when S is in the thousands.
- Single-cut is the default for large S; in between, one θ per group of scenarios.

The single-cut variant

The cut of iteration k in scenario ω, valid at every y_1:

$$\begin{aligned} \dot{V}_2^{\omega}(y_1) &\geq (\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k} \\ \sum_{\omega} \pi^{\omega}\, \dot{V}_2^{\omega}(y_1) &\geq \sum_{\omega} \pi^{\omega}\Big((\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k}\Big) \\ V_2(y_1) &\geq \Big(\underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \alpha^{\omega,k}}_{\alpha^k}\Big)^{\top}(y_1 - y_1^k) + \underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \beta^{\omega,k}}_{\beta^k = V_2(y_1^k)} \end{aligned}$$

One θ for the expected value V_2, one constraint per iteration instead of S:

$$\min_{y_1 \in \mathcal{Y}_1,\, \theta} \left\{ c_1^{\top} y_1 + \theta \mid \theta \geq (\alpha^i)^{\top}(y_1 - y_1^i) + \beta^i,\ i \in [k-1] \right\}.$$

- From the same trial points and duals, the single-cut model lies below the multi-cut one.
- But the multi-cut master grows S times faster, and becomes the bottleneck when S is in the thousands.
- Single-cut is the default for large S; in between, one θ per group of scenarios.

The single-cut variant

The cut of iteration k in scenario ω, valid at every y_1:

$$\begin{aligned} \dot{V}_2^{\omega}(y_1) &\geq (\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k} \\ \sum_{\omega} \pi^{\omega}\, \dot{V}_2^{\omega}(y_1) &\geq \sum_{\omega} \pi^{\omega}\Big((\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k}\Big) \\ V_2(y_1) &\geq \Big(\underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \alpha^{\omega,k}}_{\alpha^k}\Big)^{\top}(y_1 - y_1^k) + \underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \beta^{\omega,k}}_{\beta^k = V_2(y_1^k)} \end{aligned}$$

One θ for the expected value V_2, one constraint per iteration instead of S:

$$\min_{y_1 \in \mathcal{Y}_1,\, \theta} \left\{ c_1^{\top} y_1 + \theta \mid \theta \geq (\alpha^i)^{\top}(y_1 - y_1^i) + \beta^i,\ i \in [k-1] \right\}.$$

- From the same trial points and duals, the single-cut model lies below the multi-cut one.
- But the multi-cut master grows S times faster, and becomes the bottleneck when S is in the thousands.
- Single-cut is the default for large S; in between, one θ per group of scenarios.

The single-cut variant

The cut of iteration k in scenario ω, valid at every y_1:

$$\begin{aligned} \dot{V}_2^{\omega}(y_1) &\geq (\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k} \\ \textstyle\sum_{\omega} \pi^{\omega}\, \dot{V}_2^{\omega}(y_1) &\geq \textstyle\sum_{\omega} \pi^{\omega}\Big((\alpha^{\omega,k})^{\top}(y_1 - y_1^k) + \beta^{\omega,k}\Big) \\ V_2(y_1) &\geq \Big(\underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \alpha^{\omega,k}}_{\alpha^k}\Big)^{\top}(y_1 - y_1^k) + \underbrace{\textstyle\sum_{\omega} \pi^{\omega}\, \beta^{\omega,k}}_{\beta^k = V_2(y_1^k)} \end{aligned}$$

One θ for the expected value V_2, one constraint per iteration instead of S:

$$\min_{y_1 \in \mathcal{Y}_1,\, \theta} \left\{ c_1^{\top} y_1 + \theta \mid \theta \geq (\alpha^i)^{\top}(y_1 - y_1^i) + \beta^i,\ i \in [k-1] \right\}.$$

- From the same trial points and duals, the single-cut model lies below the multi-cut one.
- But the multi-cut master grows S times faster, and becomes the bottleneck when S is in the thousands.
- Single-cut is the default for large S; in between, one θ per group of scenarios.

Feasibility cuts



Drop relatively complete recourse: at y_1^k scenario ω may be infeasible. Its value is $+\infty$, and, D^ω being nonempty, the dual

$$\max_{\lambda} \left\{ \lambda^\top (d_2^\omega - B_2^\omega y_1^k) \mid (C_2^\omega)^\top \lambda \leq c_2^\omega \right\}$$

is unbounded: there is a direction r with

$$(C_2^\omega)^\top r \leq 0 \qquad \text{and} \qquad r^\top (d_2^\omega - B_2^\omega y_1^k) > 0 \,.$$

D^ω does not depend on y_1, so r is such a direction at every y_1; where scenario ω is feasible the dual is bounded, hence the feasibility cut

$$r^\top (d_2^\omega - B_2^\omega y_1) \;\leq\; 0 \,,$$

satisfied by every y_1 the scenario can live with, since the reverse inequality would make the dual unbounded there too, and violated by y_1^k.

- A simplex solver returns r as the ray along which it found the dual unbounded.
- No θ in it; the upper bound is not updated at such an iteration.
- The cuts so produced are the induced constraints.

Feasibility cuts



Drop relatively complete recourse: at y_1^k scenario ω may be infeasible. Its value is $+\infty$, and, D^ω being nonempty, the dual

$$\max_{\lambda} \left\{ \lambda^\top (d_2^\omega - B_2^\omega y_1^k) \mid (C_2^\omega)^\top \lambda \leq c_2^\omega \right\}$$

is unbounded: there is a direction r with

$$(C_2^\omega)^\top r \leq 0 \qquad \text{and} \qquad r^\top (d_2^\omega - B_2^\omega y_1^k) > 0 \,.$$

D^ω does not depend on y_1, so r is such a direction at every y_1; where scenario ω is feasible the dual is bounded, hence the feasibility cut

$$r^\top (d_2^\omega - B_2^\omega y_1) \;\leq\; 0 \,,$$

satisfied by every y_1 the scenario can live with, since the reverse inequality would make the dual unbounded there too, and violated by y_1^k.

- A simplex solver returns r as the ray along which it found the dual unbounded.
- No θ in it; the upper bound is not updated at such an iteration.
- The cuts so produced are the induced constraints.

Feasibility cuts



Drop relatively complete recourse: at y_1^k scenario ω may be infeasible. Its value is $+\infty$, and, D^ω being nonempty, the dual

$$\max_{\lambda} \left\{ \lambda^\top (d_2^\omega - B_2^\omega y_1^k) \mid (C_2^\omega)^\top \lambda \leq c_2^\omega \right\}$$

is unbounded: there is a direction r with

$$(C_2^\omega)^\top r \leq 0 \qquad \text{and} \qquad r^\top (d_2^\omega - B_2^\omega y_1^k) > 0 \,.$$

D^ω does not depend on y_1, so r is such a direction at every y_1; where scenario ω is feasible the dual is bounded, hence the feasibility cut

$$r^\top (d_2^\omega - B_2^\omega y_1) \;\leq\; 0 \,,$$

satisfied by every y_1 the scenario can live with, since the reverse inequality would make the dual unbounded there too, and violated by y_1^k.

- A simplex solver returns r as the ray along which it found the dual unbounded.
- No θ in it; the upper bound is not updated at such an iteration.
- The cuts so produced are the induced constraints.

Finite convergence



The duals D^ω have vertices: nonempty, and (LIN)'s full row rank makes them contain no line.

Theorem

Assume (FSN), (LIN), every D^ω nonempty, a finite optimal value, and a first master bounded below; assume too that each subproblem returns the dual of an optimal basis, and each unbounded ray it returns an extreme ray of the recession cone of D^ω. Then both variants, run with $\varepsilon = 0$, terminate after finitely many iterations with an optimal first-stage decision.

- The number of iterations is bounded by the number of bases, exponential in the second-stage dimension: a termination theorem, not a rate.
- **Integer first stage**: fine, the master becomes a MILP, cuts and finiteness survive (rational data or bounded $\mathcal{Y}_1$).
- **Integer second stage**: V_2 is no longer convex, LP cuts are no longer tight; it needs other cuts (integer L-shaped, Laporte and Louveaux 1993, which needs a binary first stage).

Finite convergence



The duals D^ω have vertices: nonempty, and (LIN)'s full row rank makes them contain no line.

Theorem

Assume (FSN), (LIN), every D^ω nonempty, a finite optimal value, and a first master bounded below; assume too that each subproblem returns the dual of an optimal basis, and each unbounded ray it returns an extreme ray of the recession cone of D^ω. Then both variants, run with $\varepsilon = 0$, terminate after finitely many iterations with an optimal first-stage decision.

- The number of iterations is bounded by the number of bases, exponential in the second-stage dimension: a termination theorem, not a rate.
- **Integer first stage**: fine, the master becomes a MILP, cuts and finiteness survive (rational data or bounded $\mathcal{Y}_1$).
- **Integer second stage**: V_2 is no longer convex, LP cuts are no longer tight; it needs other cuts (integer L-shaped, Laporte and Louveaux 1993, which needs a binary first stage).

Presentation Outline

1. Scenario decomposition
   - Dual decomposition
   - Progressive hedging
   - Bounds
2. Stage decomposition
   - The L-shaped method
   - ✠ The adaptive partition method
3. Comparing the methods

Aggregating scenarios



Another lower model: keep the recourse problem whole, shorten the list of scenarios.

Assume fixed recourse and cost: C_2, c_2 deterministic, only B_2^ω, d_2^ω random. Then one dual set $D := \{\lambda \mid C_2^\top \lambda \leq c_2\}$, assumed nonempty; relatively complete recourse assumed.

- A partition $\mathcal{K}$ of Ω into cells K; $\mathcal{K}'$ refines $\mathcal{K}$ if every cell of $\mathcal{K}'$ lies in one of $\mathcal{K}$.
- A cell has probability $\pi^K := \sum_{\omega \in K} \pi^\omega$ and mean data $B_2^K := \sum_{\omega \in K} \frac{\pi^\omega}{\pi^K} B_2^\omega$, $d_2^K := \sum_{\omega \in K} \frac{\pi^\omega}{\pi^K} d_2^\omega$ (conditional expectations).
- Partitioned recourse value: $\underline{V}_2^{\mathcal{K}}(y_1) := \sum_{K \in \mathcal{K}} \pi^K \dot{V}_2^K(y_1)$, with $\dot{V}_2^K$ the recourse value at the mean data.

The partition master

$$\min_{y_1 \in \mathcal{Y}_1,\, (y_2^K)_{K \in \mathcal{K}} \geq 0} \left\{ c_1^\top y_1 + \sum_{K \in \mathcal{K}} \pi^K c_2^\top y_2^K \mid C_2 y_2^K = d_2^K - B_2^K y_1,\ \forall K \in \mathcal{K} \right\}$$

is the extensive formulation of the problem whose scenarios are the cells: $n_1 + |\mathcal{K}| n_2$ variables. Singletons give V_2; $\{\Omega\}$ gives the problem at the mean data.

Aggregating scenarios



Another lower model: keep the recourse problem whole, shorten the list of scenarios.

Assume fixed recourse and cost: C_2, c_2 deterministic, only B_2^ω, d_2^ω random. Then one dual set $D := \{\lambda \mid C_2^\top \lambda \leq c_2\}$, assumed nonempty; relatively complete recourse assumed.

- A partition $\mathcal{K}$ of Ω into cells K; $\mathcal{K}'$ refines $\mathcal{K}$ if every cell of $\mathcal{K}'$ lies in one of $\mathcal{K}$.
- A cell has probability $\pi^K := \sum_{\omega \in K} \pi^\omega$ and mean data $B_2^K := \sum_{\omega \in K} \frac{\pi^\omega}{\pi^K} B_2^\omega$, $d_2^K := \sum_{\omega \in K} \frac{\pi^\omega}{\pi^K} d_2^\omega$ (conditional expectations).
- Partitioned recourse value: $\underline{V}_2^{\mathcal{K}}(y_1) := \sum_{K \in \mathcal{K}} \pi^K \dot{V}_2^K(y_1)$, with $\dot{V}_2^K$ the recourse value at the mean data.

The partition master

$$\min_{y_1 \in \mathcal{Y}_1,\, (y_2^K)_{K \in \mathcal{K}} \geq 0} \left\{ c_1^\top y_1 + \sum_{K \in \mathcal{K}} \pi^K c_2^\top y_2^K \mid C_2 y_2^K = d_2^K - B_2^K y_1,\ \forall K \in \mathcal{K} \right\}$$

is the extensive formulation of the problem whose scenarios are the cells: $n_1 + |\mathcal{K}| n_2$ variables. Singletons give V_2; $\{\Omega\}$ gives the problem at the mean data.

Aggregating scenarios



Another lower model: keep the recourse problem whole, shorten the list of scenarios.

Assume fixed recourse and cost: C_2, c_2 deterministic, only B_2^ω, d_2^ω random. Then one dual set $D := \{\lambda \mid C_2^\top \lambda \leq c_2\}$, assumed nonempty; relatively complete recourse assumed.

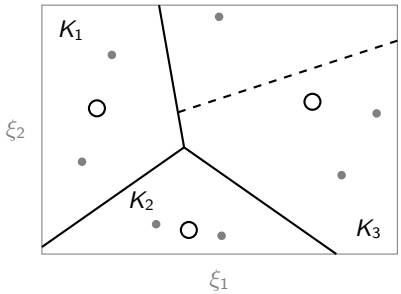
- A partition $\mathcal{K}$ of Ω into cells K; $\mathcal{K}'$ refines $\mathcal{K}$ if every cell of $\mathcal{K}'$ lies in one of $\mathcal{K}$.
- A cell has probability $\pi^K := \sum_{\omega \in K} \pi^\omega$ and mean data $B_2^K := \sum_{\omega \in K} \frac{\pi^\omega}{\pi^K} B_2^\omega$, $d_2^K := \sum_{\omega \in K} \frac{\pi^\omega}{\pi^K} d_2^\omega$ (conditional expectations).
- Partitioned recourse value: $\underline{V}_2^{\mathcal{K}}(y_1) := \sum_{K \in \mathcal{K}} \pi^K \dot{V}_2^K(y_1)$, with $\dot{V}_2^K$ the recourse value at the mean data.

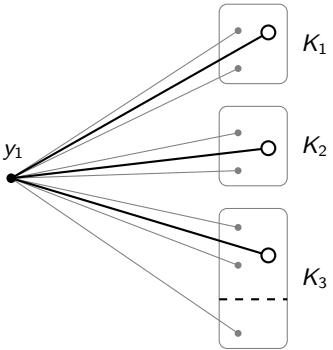
The partition master

$$\min_{y_1 \in \mathcal{Y}_1,\, (y_2^K)_{K \in \mathcal{K}} \geq 0} \left\{ c_1^\top y_1 + \sum_{K \in \mathcal{K}} \pi^K c_2^\top y_2^K \mid C_2 y_2^K = d_2^K - B_2^K y_1,\ \forall K \in \mathcal{K} \right\}$$

is the extensive formulation of the problem whose scenarios are the cells: $n_1 + |\mathcal{K}| n_2$ variables. Singletons give V_2; $\{\Omega\}$ gives the problem at the mean data.

Cells and their means





Seven equiprobable scenarios (dots) in three cells, each replaced by its mean (circles): left, as points of the plane of ξ; right, as branches of the scenario fan. The dashed line is the next split, between scenarios of K_3 that returned different duals.

Dual vectors constant on cells

By LP duality, each scenario takes its dual vector in D:

$$V_2(y_1) = \sup \Big\{ \sum_{\omega \in \Omega} \pi^\omega \, \lambda^{\omega\top} h^\omega(y_1) \mid \lambda^\omega \in D, \ \forall \omega \Big\} .$$

A cell is one scenario at the mean data, taking its dual vector in the same D:

$$\underline{V_2^{\mathcal{K}}}(y_1) = \sup \Big\{ \sum_{\omega \in \Omega} \pi^\omega \, \lambda^{\omega\top} h^\omega(y_1) \mid \lambda^\omega = \lambda^K \in D, \ \forall \omega \in K, \ \forall K \in \mathcal{K} \Big\} .$$

The partitioned value is the recourse value with the dual vectors constant on each cell.

Proposition (Aggregating is a relaxation, and refining tightens it)

With C_2, c_2 deterministic and $D \neq \emptyset$:

(i) *for every cell K, $\dot{V}_2^K \leq \sum_{\omega \in K} \frac{\pi^\omega}{\pi^K} \dot{V}_2^\omega$, so $\underline{V_2^{\mathcal{K}}} \leq V_2$;*

(ii) *if $\mathcal{K}'$ refines $\mathcal{K}$, then $\underline{V_2^{\mathcal{K}}} \leq \underline{V_2^{\mathcal{K}'}}$.*

Dual vectors constant on cells



By LP duality, each scenario takes its dual vector in D:

$$V_2(y_1) = \sup \Big\{ \sum_{\omega \in \Omega} \pi^\omega \, \lambda^{\omega\top} h^\omega(y_1) \mid \lambda^\omega \in D, \ \forall \omega \Big\} .$$

A cell is one scenario at the mean data, taking its dual vector in the same D:

$$\underline{V}_2^{\mathcal{K}}(y_1) = \sup \Big\{ \sum_{\omega \in \Omega} \pi^\omega \, \lambda^{\omega\top} h^\omega(y_1) \mid \lambda^\omega = \lambda^K \in D, \ \forall \omega \in K, \ \forall K \in \mathcal{K} \Big\} .$$

The partitioned value is the recourse value with the dual vectors constant on each cell.

Proposition (Aggregating is a relaxation, and refining tightens it)

With C_2, c_2 deterministic and $D \neq \emptyset$:

(i) *for every cell K, $\dot{V}_2^K \leq \sum_{\omega \in K} \frac{\pi^\omega}{\pi^K} \dot{V}_2^\omega$, so $\underline{V}_2^{\mathcal{K}} \leq V_2$;*

(ii) *if $\mathcal{K}'$ refines $\mathcal{K}$, then $\underline{V}_2^{\mathcal{K}} \leq \underline{V}_2^{\mathcal{K}'}$.*

When is a cell exact?

A partition is adapted to a first-stage decision y_1 if $\underline{V}_2^{\mathcal{K}}(y_1) = V_2(y_1)$.

Proposition (A cell is exact where its scenarios share a dual)

With C_2, c_2 deterministic, $D \neq \emptyset$ and relatively complete recourse,

$$\pi^K \dot{V}_2^K(y_1) = \sum_{\omega \in K} \pi^\omega \dot{V}_2^\omega(y_1) \quad \Longleftrightarrow \quad \text{the scenarios of } K \text{ share an optimal dual at } y_1 \ .$$

So: split each cell by the duals its scenarios return, and the partition is adapted to the current decision. Song and Luedtke (2015) call a partition whose master has the optimal value *completely sufficient*.

When is a cell exact?

A partition is adapted to a first-stage decision y_1 if $\underline{V}_2^{\mathcal{K}}(y_1) = V_2(y_1)$.

Proposition (A cell is exact where its scenarios share a dual)

With C_2, c_2 deterministic, $D \neq \emptyset$ and relatively complete recourse,

$$\pi^K \dot{V}_2^K(y_1) = \sum_{\omega \in K} \pi^\omega \dot{V}_2^\omega(y_1) \quad \Longleftrightarrow \quad \text{the scenarios of } K \text{ share an optimal dual at } y_1 \ .$$

So: split each cell by the duals its scenarios return, and the partition is adapted to the current decision. Song and Luedtke (2015) call a partition whose master has the optimal value *completely sufficient*.

The adaptive partition method



1. Start with one cell, $\mathcal{K}^0 = \{\Omega\}$: the master is the problem at the mean data.
2. Solve the master on $\mathcal{K}^{k-1}$: lower bound LB^k, decision y_1^k.
3. Solve the second-stage problem of every scenario (not every cell) at y_1^k: values $\dot{V}_2^\omega(y_1^k)$, duals $\lambda^{\omega,k}$.
4. Upper bound $\mathrm{UB}^k = c_1^\top y_1^k + \sum_\omega \pi^\omega \dot{V}_2^\omega(y_1^k)$; keep the best.
5. Split each cell into the classes of its scenarios that returned the same dual:

$$\mathcal{K}^k = \left\{ \{\omega \in K \mid \lambda^{\omega,k} = \lambda\} \mid K \in \mathcal{K}^{k-1},\ \lambda \in \{\lambda^{\omega',k} \mid \omega' \in K\} \right\}.$$

6. Stop if $\mathrm{UB} - \mathrm{LB}^k \leq \varepsilon$; else go to 2.

- The scenario solves are the L-shaped method's, solve for solve; only the use of the duals differs: cuts there, a partition here.
- The new partition is adapted to y_1^k: the next master is exact there, as a cut would be.

The adaptive partition method



1. Start with one cell, $\mathcal{K}^0 = \{\Omega\}$: the master is the problem at the mean data.
2. Solve the master on $\mathcal{K}^{k-1}$: lower bound LB^k, decision y_1^k.
3. Solve the second-stage problem of every scenario (not every cell) at y_1^k: values $\dot{V}_2^\omega(y_1^k)$, duals $\lambda^{\omega,k}$.
4. Upper bound $\mathrm{UB}^k = c_1^\top y_1^k + \sum_\omega \pi^\omega \dot{V}_2^\omega(y_1^k)$; keep the best.
5. Split each cell into the classes of its scenarios that returned the same dual:

$$\mathcal{K}^k = \left\{ \{\omega \in K \mid \lambda^{\omega,k} = \lambda\} \mid K \in \mathcal{K}^{k-1},\ \lambda \in \{\lambda^{\omega',k} \mid \omega' \in K\} \right\}.$$

6. Stop if $\mathrm{UB} - \mathrm{LB}^k \leq \varepsilon$; else go to 2.

- The scenario solves are the L-shaped method's, solve for solve; only the use of the duals differs: cuts there, a partition here.
- The new partition is adapted to y_1^k: the next master is exact there, as a cut would be.

Four decomposition methods I

	Dual decomposition	**Progressive hedging**	**L-shaped**	**Adaptive partitions**
problems	convex; integer still gives a bound	convex; integer gives a bound and a heuristic	linear; the first stage may be integer	linear, recourse matrix and cost fixed; first stage may be integer
what couples	nonanticipativity, dualized	nonanticipativity, dualized and penalized	the first-stage variable, kept; the recourse approximated by cuts	the first-stage variable, kept; the scenarios aggregated in cells
starts from	$\lambda = 0$, where $v^{\mathrm{D}}(0) = v^{\mathrm{PI}}$	the scenarios' own optima, as at $\lambda = 0$	initial cuts or bounds on each θ^{ω}	one cell, the mean data (v^{EV} for affine data)
iterate	S disagreeing copies	S copies, agreeing in the limit	one decision in $\mathcal{Y}_1$	one decision in $\mathcal{Y}_1$

Four decomposition methods II

	Dual decomposition	**Progressive hedging**	**L-shaped**	**Adaptive partitions**
bounds	lower, from the scenario solves; upper at any admissible decision, S extra solves	lower $v^{\mathrm{D}}(\lambda^k)$; upper at $\bar{y}_1^k$, possibly $+\infty$; S extra solves each; a loose lower bound with none	both from the solves of the iteration	both from the solves of the iteration
convergence	of the bound, with diminishing steps; the copies need not converge	asymptotic, any constant τ	finite, through the bases	finite, at most S masters under relatively complete recourse
per iteration	S scenario problems	S scenario problems with a quadratic term	S subproblems, a master gaining rows	S subproblems, a master gaining scenarios
implementation	a loop around a deterministic solver	a loop around a deterministic solver	a master to build and maintain	a master to rebuild at each split

Choosing a method

- **Subproblems.** All four solve S independent subproblems per iteration, in parallel. Scenario methods: both stages of one scenario, any solver (integer, nonlinear); stage methods: the second-stage LP, only its right-hand side changes (warm start).
- **Master.** What separates them. Scenario methods: an average and an update, in closed form. Stage methods: an LP that grows, and in the worst case ends equivalent to the extensive formulation.
- **Integer variables.** Integer first stage: stage methods unchanged, with a MILP master; scenario methods keep a duality gap (a bound, a heuristic). Integer recourse reverses the roles: L-shaped needs other cuts, the partition master is no longer a relaxation.
- **Certificates.** Stage methods need a linear second stage, and relatively complete recourse or feasibility cuts; they give a certificate at every iteration and terminate finitely: the ones to reach for when the answer must be certified.
- **Tuning.** Scenario methods need a parameter (steps, τ) and wrap an existing model; stage methods need the problem cut into master and subproblem, with the duals read from the solver.

Choosing a method

- **Subproblems.** All four solve S independent subproblems per iteration, in parallel. Scenario methods: both stages of one scenario, any solver (integer, nonlinear); stage methods: the second-stage LP, only its right-hand side changes (warm start).
- **Master.** What separates them. Scenario methods: an average and an update, in closed form. Stage methods: an LP that grows, and in the worst case ends equivalent to the extensive formulation.
- **Integer variables.** Integer first stage: stage methods unchanged, with a MILP master; scenario methods keep a duality gap (a bound, a heuristic). Integer recourse reverses the roles: L-shaped needs other cuts, the partition master is no longer a relaxation.
- **Certificates.** Stage methods need a linear second stage, and relatively complete recourse or feasibility cuts; they give a certificate at every iteration and terminate finitely: the ones to reach for when the answer must be certified.
- **Tuning.** Scenario methods need a parameter (steps, τ) and wrap an existing model; stage methods need the problem cut into master and subproblem, with the duals read from the solver.

Choosing a method

- **Subproblems.** All four solve S independent subproblems per iteration, in parallel. Scenario methods: both stages of one scenario, any solver (integer, nonlinear); stage methods: the second-stage LP, only its right-hand side changes (warm start).
- **Master.** What separates them. Scenario methods: an average and an update, in closed form. Stage methods: an LP that grows, and in the worst case ends equivalent to the extensive formulation.
- **Integer variables.** Integer first stage: stage methods unchanged, with a MILP master; scenario methods keep a duality gap (a bound, a heuristic). Integer recourse reverses the roles: L-shaped needs other cuts, the partition master is no longer a relaxation.
- **Certificates.** Stage methods need a linear second stage, and relatively complete recourse or feasibility cuts; they give a certificate at every iteration and terminate finitely: the ones to reach for when the answer must be certified.
- **Tuning.** Scenario methods need a parameter (steps, τ) and wrap an existing model; stage methods need the problem cut into master and subproblem, with the duals read from the solver.

Choosing a method

- **Subproblems.** All four solve S independent subproblems per iteration, in parallel. Scenario methods: both stages of one scenario, any solver (integer, nonlinear); stage methods: the second-stage LP, only its right-hand side changes (warm start).
- **Master.** What separates them. Scenario methods: an average and an update, in closed form. Stage methods: an LP that grows, and in the worst case ends equivalent to the extensive formulation.
- **Integer variables.** Integer first stage: stage methods unchanged, with a MILP master; scenario methods keep a duality gap (a bound, a heuristic). Integer recourse reverses the roles: L-shaped needs other cuts, the partition master is no longer a relaxation.
- **Certificates.** Stage methods need a linear second stage, and relatively complete recourse or feasibility cuts; they give a certificate at every iteration and terminate finitely: the ones to reach for when the answer must be certified.
- **Tuning.** Scenario methods need a parameter (steps, τ) and wrap an existing model; stage methods need the problem cut into master and subproblem, with the duals read from the solver.

Choosing a method

- **Subproblems.** All four solve S independent subproblems per iteration, in parallel. Scenario methods: both stages of one scenario, any solver (integer, nonlinear); stage methods: the second-stage LP, only its right-hand side changes (warm start).
- **Master.** What separates them. Scenario methods: an average and an update, in closed form. Stage methods: an LP that grows, and in the worst case ends equivalent to the extensive formulation.
- **Integer variables.** Integer first stage: stage methods unchanged, with a MILP master; scenario methods keep a duality gap (a bound, a heuristic). Integer recourse reverses the roles: L-shaped needs other cuts, the partition master is no longer a relaxation.
- **Certificates.** Stage methods need a linear second stage, and relatively complete recourse or feasibility cuts; they give a certificate at every iteration and terminate finitely: the ones to reach for when the answer must be certified.
- **Tuning.** Scenario methods need a parameter (steps, τ) and wrap an existing model; stage methods need the problem cut into master and subproblem, with the duals read from the solver.

Exercise: choosing a method

For each two-stage problem, which method would you run first, and why?

1. A linear program with 10^5 sampled scenarios and three first-stage variables.
2. Binary first-stage decisions, a linear recourse, and 20 scenarios.
3. A linear program with 500 first-stage variables, 20 scenarios and random second-stage costs, with a fast solver already written for one deterministic scenario.

Answers.

1. Single-cut L-shaped: one row per iteration and a master in three variables, where multi-cut would gain 10^5 rows at each one. A certificate at every iteration.
2. The extensive formulation first, one mixed-integer program with 20 blocks. If it is too large, L-shaped with a mixed-integer master: the cuts and the finite termination survive an integer first stage, while the scenario methods keep a duality gap there, so they return a bound and a heuristic rather than a certificate.
3. Multi-cut L-shaped adds only 20 rows per iteration, but its master is a linear program in 500 variables and is slow in its tail. Progressive hedging has no master and calls the solver you already have, in exchange for a decision without a certificate, its bounds costing extra solves.

Exercise: choosing a method

For each two-stage problem, which method would you run first, and why?

1. A linear program with 10^5 sampled scenarios and three first-stage variables.
2. Binary first-stage decisions, a linear recourse, and 20 scenarios.
3. A linear program with 500 first-stage variables, 20 scenarios and random second-stage costs, with a fast solver already written for one deterministic scenario.

Answers.

1. Single-cut L-shaped: one row per iteration and a master in three variables, where multi-cut would gain 10^5 rows at each one. A certificate at every iteration.
2. The extensive formulation first, one mixed-integer program with 20 blocks. If it is too large, L-shaped with a mixed-integer master: the cuts and the finite termination survive an integer first stage, while the scenario methods keep a duality gap there, so they return a bound and a heuristic rather than a certificate.
3. Multi-cut L-shaped adds only 20 rows per iteration, but its master is a linear program in 500 variables and is slow in its tail. Progressive hedging has no master and calls the solver you already have, in exchange for a decision without a certificate, its bounds costing extra solves.

Exercise: choosing a method

For each two-stage problem, which method would you run first, and why?

1. A linear program with 10^5 sampled scenarios and three first-stage variables.
2. Binary first-stage decisions, a linear recourse, and 20 scenarios.
3. A linear program with 500 first-stage variables, 20 scenarios and random second-stage costs, with a fast solver already written for one deterministic scenario.

Answers.

1. Single-cut L-shaped: one row per iteration and a master in three variables, where multi-cut would gain 10^5 rows at each one. A certificate at every iteration.
2. The extensive formulation first, one mixed-integer program with 20 blocks. If it is too large, L-shaped with a mixed-integer master: the cuts and the finite termination survive an integer first stage, while the scenario methods keep a duality gap there, so they return a bound and a heuristic rather than a certificate.
3. Multi-cut L-shaped adds only 20 rows per iteration, but its master is a linear program in 500 variables and is slow in its tail. Progressive hedging has no master and calls the solver you already have, in exchange for a decision without a certificate, its bounds costing extra solves.

Exercise: choosing a method

For each two-stage problem, which method would you run first, and why?

1. A linear program with 10^5 sampled scenarios and three first-stage variables.
2. Binary first-stage decisions, a linear recourse, and 20 scenarios.
3. A linear program with 500 first-stage variables, 20 scenarios and random second-stage costs, with a fast solver already written for one deterministic scenario.

Answers.

1. Single-cut L-shaped: one row per iteration and a master in three variables, where multi-cut would gain 10^5 rows at each one. A certificate at every iteration.
2. The extensive formulation first, one mixed-integer program with 20 blocks. If it is too large, L-shaped with a mixed-integer master: the cuts and the finite termination survive an integer first stage, while the scenario methods keep a duality gap there, so they return a bound and a heuristic rather than a certificate.
3. Multi-cut L-shaped adds only 20 rows per iteration, but its master is a linear program in 500 variables and is slow in its tail. Progressive hedging has no master and calls the solver you already have, in exchange for a decision without a certificate, its bounds costing extra solves.

What you must remember

- Before choosing a method, look at the scenario problems (linear, convex, integer), the number of scenarios and of first-stage variables, and whether you need a certificate.
- Many convex scenario problems your solver already handles: dualize nonanticipativity; dual decomposition for a bound, progressive hedging for a decision.
- A linear second stage and a certificate: L-shaped, with every constraint you know on y_1 written in the first stage.
- Fixed recourse matrix and cost, few distinct second-stage duals: aggregate into cells, split where the duals disagree.
- Return the best decision found, not the last iterate, wherever the method keeps an incumbent, and report its gap to a lower bound.

What you should remember

- The multiplier of nonanticipativity is the price of information: under convexity and qualification, and when v^{SP} is finite, the dual climbs from v^{PI} to v^{SP}.
- The quadratic term of progressive hedging makes the copies agree, under convexity and when an optimal decision and an optimal price of information exist, whatever the penalty.
- An optimality cut is an affine minorant of the recourse value built from second-stage duals; a feasibility cut is an induced constraint.
- Fixed recourse matrix and cost: replacing a group of scenarios by its mean underestimates the recourse cost; under relatively complete recourse it is exact at a decision if and only if the group shares an optimal second-stage dual there.