Business Analytics – 2022/23

Decomposition methods in optimization

Prof. Paolo Brandimarte
Dip. di Scienze Matematiche – Politecnico di Torino
e-mail: paolo.brandimarte@polito.it
URL: staff.polito.it/paolo.brandimarte

This version: August 6, 2024

- A.J. Conejo , E. Castillo , R. Mínguez, R. García-Bertrand. *Decomposition Techniques in Mathematical Programming*. Springer, 2006.
- P. Kall, S.W. Wallace. *Stochastic Programming*. Wiley, 1994. The book pdf can be downloaded from `http://stoprog.org/index.html?resources.html`
- J.L. Higle, S. Sen. *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming.* Kluwer, 1996.

Decomposition methods play a prominent role in optimization, as they allow:

- To take advantage of favorable structure:
  - network sub problems;
  - from NP–hard to polynomial.
- To parallelize the solution of large-scale problems.
- To tackle large-scale, scenario-based stochastic optimization models.
- To tackle hard combinatorial problems by dealing with a sequence of simpler subproblems, possibly mixing different solution strategies.
- To avoid modeling issues with very difficult constraints.

Consider the optimization problem

$$\begin{array}{ll} \text{opt} & \displaystyle\sum_{j \in [n]} f_j(\mathbf{x}_j) \\ \text{s.t.} & \mathbf{x}_j \in S_j. \end{array}$$

Clearly, we may decompose the problem into decoupled subproblems $\text{opt}_{\mathbf{x}_j \in S_j} f_j(\mathbf{x}_j)$, as subvectors $\mathbf{x}_j$ are subject to independent constraints.

In a linear programming model,

$$\begin{array}{ll} \min & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

this would correspond to a block-diagonal matrix structure (missing submatrices are all zeros):

$$\mathbf{A} = \begin{bmatrix} \mathbf{D}_1 & & & & \\ & \mathbf{D}_2 & & & \\ & & \mathbf{D}_3 & & \\ & & & \ddots & \\ & & & & \mathbf{D}_n \end{bmatrix}$$

Usually, we are not that lucky, and there will be a complicating factor.

For instance, we may deal with a large-scale LP featuring a block-angular structure in the technological matrix:

$$\mathbf{A} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{C}_3 & \cdots & \mathbf{C}_n \\ \mathbf{D}_1 & & & & \\ & \mathbf{D}_2 & & & \\ & & \mathbf{D}_3 & & \\ & & & \ddots & \\ & & & & \mathbf{D}_n \end{bmatrix}, \qquad \text{or} \qquad \mathbf{A} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{D}_1 & & & & \\ \mathbf{C}_2 & & \mathbf{D}_2 & & & \\ \mathbf{C}_3 & & & \mathbf{D}_3 & & \\ \vdots & & & & \ddots & \\ \mathbf{C}_n & & & & & \mathbf{D}_n \end{bmatrix}.$$

In the first case, the complicating factor is a set of of interaction constraints coupling subproblems and preventing the decomposition by the block diagonal structure. We may decompose the problem if we relax the interaction constraints in some way (e.g., Lagrangian dual decomposition).

In the second case, we have interaction variables preventing the decomposition. We may decompose the problem if we fix the interaction variables (e.g., L-shaped decomposition, which is Benders decomposition for stochastic programming).

There is a fair number of inter-related decomposition methods. Some are exact, at least in principle, some lead to approximate algorithms:

- Lagrangian relaxation and Lagrangian decomposition.
- Dual heuristics.
- Dantzig–Wolfe decomposition.
- Column generation.
- Hierarchical decomposition.
- Matheuristics (not to be confused with metaheuristics).
- Benders decomposition for MILPs with special structure.
- L-shaped decomposition in two-stage stochastic programming with recourse.
- Progressive hedging for multistage stochastic programming with recourse.
- Dynamic programming (time-based decomposition).

Part 1

Dual decomposition and progressive hedging for stochastic programming

To see how duality can help in a simple setting, let us consider a problem like

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} f_i(\mathbf{x}_i) && (1)\\
\text{s.t.} \quad & \sum_{i=1}^{n} g_i(\mathbf{x}_i) \le b, && (2)\\
& \mathbf{x}_i \in S_i, \quad i = 1, \ldots, n && (3)
\end{aligned}$$

Let us interpret the decision variables $\mathbf{x}_i$, $i = 1, \ldots, n$, as activities yielding a profit $f_i(\mathbf{x}_i)$ and consuming a resource amount $g_i(\mathbf{x}_i)$. The objective function (1) is total profit, and (2) is a *budget* constraint on the resource.

Note that the objective function is measured in monetary terms, whereas b is measured in resource units.

If we could get rid of the budget constraint, the problem could be decomposed.

Let us dualize the budget constraint by introducing the multiplier $\mu \geq 0$ and writing the Lagrangian function:

$$\mathcal{L}(\mathbf{x}, \mu) = \sum_{i=1}^{n} f_i(\mathbf{x}_i) + \mu \left(b - \sum_{i=1}^{n} g_i(\mathbf{x}_i) \right) = \sum_{i=1}^{n} [f_i(\mathbf{x}_i) - \mu g_i(\mathbf{x}_i)] + \mu b.$$

Note that here we must adjust the problem to account for the optimization sense.

The Lagrangian function should be maximized with respect to the primal variable, resulting in a set of problems:

$$\max_{\mathbf{x}_i \in S_i} [f_i(\mathbf{x}_i) - \mu g_i(\mathbf{x}_i)] \equiv \pi_i(\mathbf{x}_i)$$

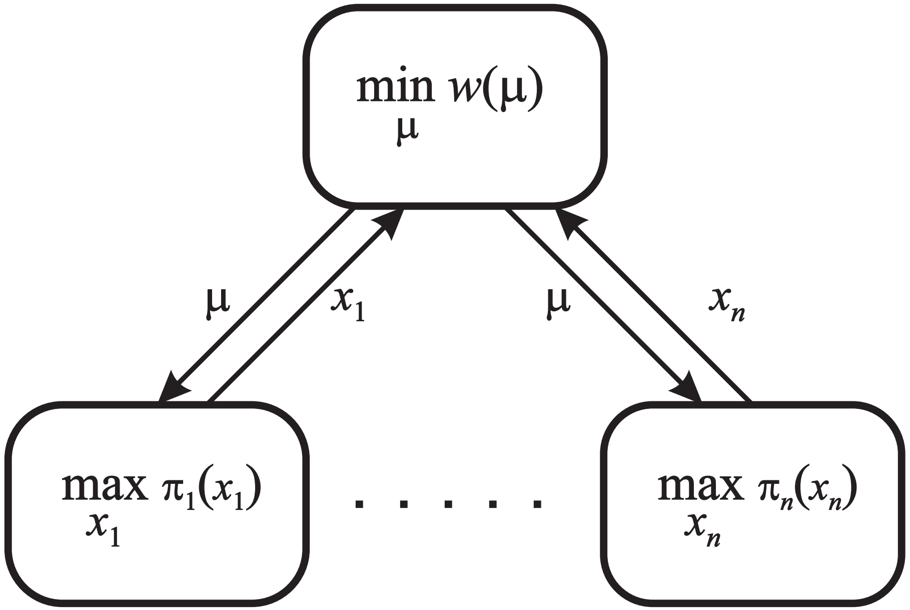
- Each subproblem requires maximizing profit contribution minus resource cost.
- The multiplier μ is a shadow price, measured in unit of money per unit amount of resource.
- In this case, we should minimize the dual function with respect to μ.

- Given relaxed solutions $\mathbf{x}_i^*$, we get a subgradient of the dual function:

$$\sum_{i=1}^{n} g_i(\mathbf{x}_i^*) - b$$

- This is positive when the budget is exceeded, in which case we should increase the resource price. The price must be reduced when the budget is not exceeded.

- The resulting demand-offer scheme can be depicted as follows.



Dual decomposition

- Dual decomposition may converge poorly in practice, but it might be a good approach for some specially structured large-scale problems.

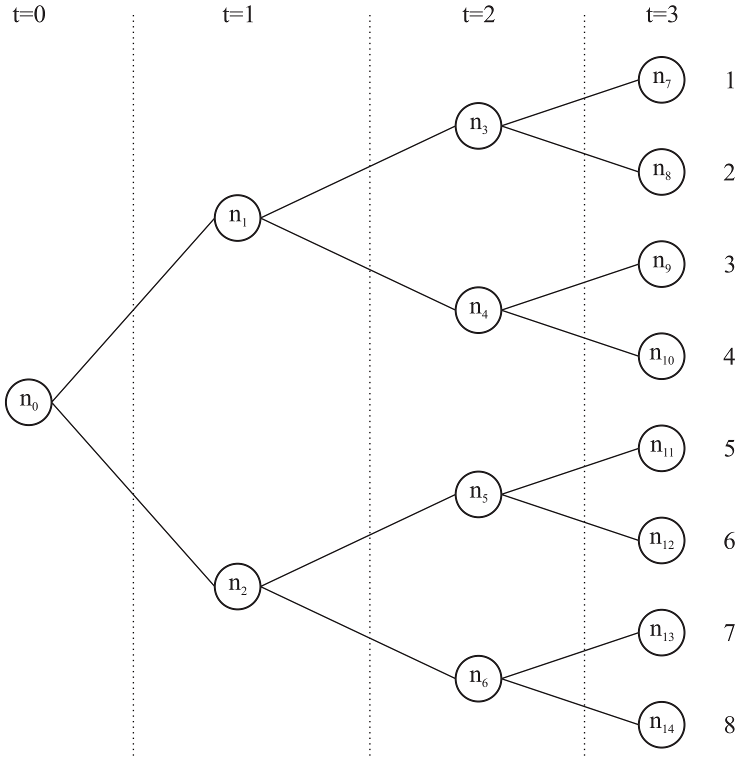
- Sometimes, we are satisfied by a suitably good solution. If we may recover a good *primal feasible* solution from dual decomposition, we obtain dual heuristic algorithm.

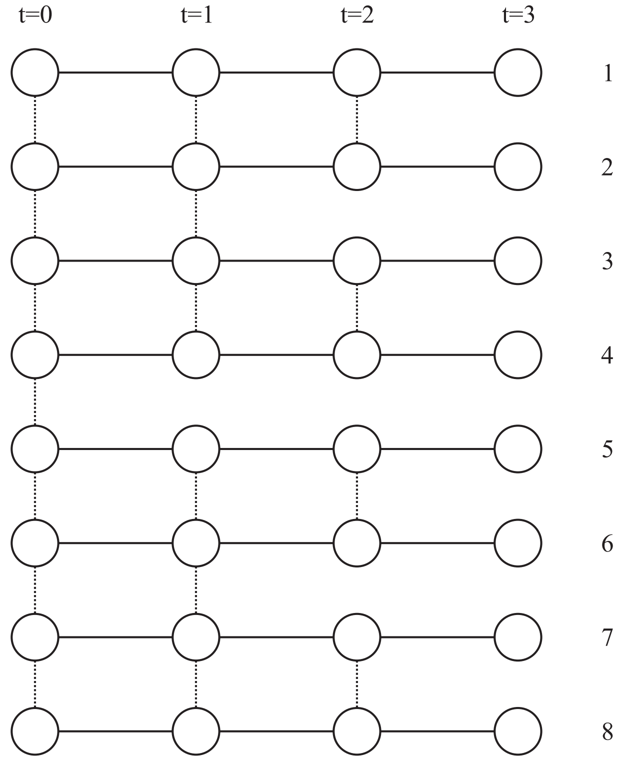
- Lagrangian methods can be integrated with penalty function methods, resulting in augmented Lagrangian schemes based, e.g., on the minimization of

$$f(\mathbf{x}) + \sum_{i \in I} \lambda_i h_i(\mathbf{x}) + \sigma \sum_{i \in I} h_i^2(\mathbf{x})$$

  for an equality-constrained problem.

Uncertainty in multistage stochastic programming is represented by scenario trees:





- In a compact model formulation, we define decision variables $\mathbf{x}^n$ associated with node n.

- In a split-variable model formulation, we decision variables $\mathbf{x}_t^s$, associated with scenario s at time t.

In the second case, we need to enforce non-anticipativity constraints explicitly: decision variables corresponding to different scenarios at the same time t must take the same value, if the two scenarios are indistinguishable at time t

In the case depicted in the figures, let us consider node n_1. Scenarios $s = 1, 2, 3, 4$ pass through this node and are indistinguishable at time $t = 1$. Hence, we must have:

$$\mathbf{x}_1^1 = \mathbf{x}_1^2 = \mathbf{x}_1^3 = \mathbf{x}_1^4.$$

By the same token, at time $t = 2$ we have constraints like

$$\mathbf{x}_2^5 = \mathbf{x}_2^6.$$

More generally, it is customary to denote by $\{s\}_t$ the set of scenarios which are not distinguishable from s up to time t. For instance:

$$\begin{aligned} \{1\}_0 &= \{1, 2, 3, 4, 5, 6, 7, 8\} \\ \{2\}_1 &= \{1, 2, 3, 4\} \\ \{5\}_2 &= \{5, 6\}. \end{aligned}$$

Then, the non-anticipativity constraints may be written as

$$\mathbf{x}_t^s = \mathbf{x}_t^{s'} \qquad \forall t, s, s' \in \{s\}_t.$$

This is not the only way of expressing the non-anticipativity requirement, and the best approach depends on the chosen solution algorithm.

Progressive hedging is a strategy that can be applied to multistage, possibly nonlinear stochastic programming problems.

Let us consider a dynamic system with state vector $\mathbf{z}_t$ and state equation $\mathbf{z}_{t+1} = G_t(\mathbf{z}_t, \mathbf{x}_t, \boldsymbol{\xi}_{t+1})$, $t = 0, 1, \ldots, T$, where $\mathbf{z}_0$ is given, $\mathbf{x}_t$ is the control variable, and $\boldsymbol{\xi}_{t+1}$ is a random variable realized after the decision at time t.

Let $s = (\boldsymbol{\xi}_1^s, \boldsymbol{\xi}_2^s, \ldots, \boldsymbol{\xi}_{T+1}^s)$ be a scenario, with probability π^s, and let us consider the individual scenario problem:

$$\begin{aligned}
\min \quad & \sum_{t=0}^{T} \gamma^t f_t(\mathbf{z}_t, \mathbf{x}_t, \boldsymbol{\xi}_{t+1}^s) + \gamma^{T+1} Q(\mathbf{z}_{T+1}) \\
\text{s.t.} \quad & \mathbf{z}_{t+1} = G_t(\mathbf{z}_t, \mathbf{x}_t, \boldsymbol{\xi}_{t+1}^s), \qquad t = 0, 1, \ldots, T \\
& \mathbf{L}_t(\mathbf{z}_t) \le \mathbf{x}_t \le \mathbf{U}_t(\mathbf{z}_t), \qquad t = 0, 1, \ldots, T
\end{aligned}$$

with terminal state cost $Q(\cdot)$ and discount factor $\gamma \in (0, 1]$.

Let $(\mathbf{x}_0^s, \mathbf{x}_1^s, \ldots, \mathbf{x}_T^s)$ be the optimal solution for the scenario problem s.

Can we just aggregate the individual solutions and choose $\underline{\mathbf{x}}_t = \sum_{s \in \mathcal{S}} \pi^s \mathbf{x}_t^s$? unfortunately, there is no reason to believe that this would be the optimal solution (or even a feasible one, for that matter).

A good policy should take advantage of information, but it cannot be anticipative. We can formulate the multistage problem within a split-variable modeling framework, introducing nonanticipativity constraints in a suitable form.

Let $\{s\}_t$ be the set of scenarios that, up to time instant t, cannot be distinguished from scenario s, and let $\mathbb{P}(\{s\}_t)$ be the sum of their probabilities.

The stochastic optimization problem can be stated as:

$$
\begin{aligned}
\min \quad & \sum_{s \in \mathcal{S}} \pi^s \left(\sum_{t=0}^{T} \gamma^t f_t(\mathbf{z}_t^s, \mathbf{x}_t^s, \boldsymbol{\xi}_{t+1}^s) + \gamma^{T+1} Q(\mathbf{z}_{T+1}^s) \right) \\
\text{s.t.} \quad & \mathbf{z}_{t+1}^s = G_t(\mathbf{z}_t^s, \mathbf{x}_t^s, \boldsymbol{\xi}_{t+1}^s), \qquad \forall s \in \mathcal{S},\ t = 0, 1, \ldots, T \\
& \mathbf{L}_t(\mathbf{z}_t^s) \leq \mathbf{x}_t^s \leq \mathbf{U}_t(\mathbf{z}_t^s), \qquad \forall s \in \mathcal{S},\ t = 0, 1, \ldots, T \\
& \mathbf{x}_t^s = \sum_{s' \in \{s\}_t} \frac{\pi^{s'} \mathbf{x}_t^{s'}}{\mathbb{P}(\{s\}_t)}, \qquad \forall s \in \mathcal{S},\ t = 0, 1, \ldots, T
\end{aligned} \tag{4}
$$

Constraints (4) enforce nonanticipativity: each decision in $\{s\}_t$ must be the same at time t. Here we use a single conditional expectation per scenario and time period, rather than pairwise equalities between decisions $\{s\}_t$. Let us introduce Lagrange multipliers w_t^s for nonanticipativity constraints.

Then, dualization of (4) yields:

$$
\min \quad \sum_{s \in \mathcal{S}} \pi^s \left\{ \sum_{t=0}^{T} \gamma^t \Big[f_t(\mathbf{z}_t^s, \mathbf{x}_t^s, \boldsymbol{\xi}_{t+1}^s) + w_t^s \big(\mathbf{x}_t^s - \underline{\mathbf{x}}(\{s\}_t) \big) \Big] + \gamma^{T+1} Q(\mathbf{z}_{T+1}^s) \right\}
$$

where the conditional expectation

$$
\underline{\mathbf{x}}(\{s\}_t) \equiv \sum_{s' \in \{s\}_t} \frac{\pi^{s'} \mathbf{x}_t^{s'}}{\mathsf{P}(\{s\}_t)}
$$

can be interpreted as a projection on the set of nonanticipative policies.

This problem can be rearranged and decomposed by scenarios. In order to improve convergence, we may add a quadratic penalty term to yield an augmented Lagrangian method.

Unfortunately, the penalty term would destroy the decomposition structure.

However, we may use the scenario-aggregated solution $\underline{\mathbf{x}}(\{s\}_t)$ from the *previous* iteration, which yields (neglecting constant terms), the individual scenario problem:

$$\min \quad \sum_{t=0}^{T} \gamma^t \left\{ f_t(\mathbf{z}_t^s, \mathbf{x}_t^s, \boldsymbol{\xi}_{t+1}^s) + w_t^s \mathbf{x}_t^s + \frac{1}{2}\rho \big[\mathbf{x}_t^s - \underline{\mathbf{x}}(\{s\}_t)\big]^2 \right\} + \gamma^{T+1} Q(\mathbf{z}_{T+1}^s)$$

where the quadratic term may be interpreted as a regularization term.

The choice of ρ and of the multiplier adjustment scheme are quite critical for convergence.

Nevertheless, progressive hedging is useful in order to take advantage of problem structure, and it has been used to devise heuristics for hard mixed-integer, multistage stochastic programs.

Part 2

Benders decomposition and L-shaped decomposition for stochastic programming

Consider the convex problem $\min_{\mathbf{x}\in S} f(\mathbf{x})$, and suppose that, for a given point $\mathbf{x}^k$, we are not only able to compute the function value $f(\mathbf{x}^k)$, but also a subgradient $\boldsymbol{\gamma}_k$, which does exist if the function is convex on the set S.

In other words, we are able to find an affine function such that

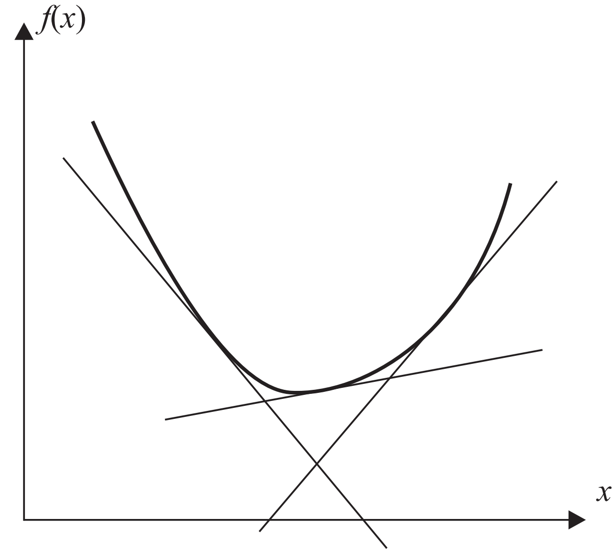
$$f(\mathbf{x}^k) = \alpha_k + \boldsymbol{\gamma}_k^{\mathsf{T}}\mathbf{x}^k \tag{5}$$

$$f(\mathbf{x}) \geq \alpha_k + \boldsymbol{\gamma}_k^{\mathsf{T}}\mathbf{x} \qquad \forall \mathbf{x} \in S. \tag{6}$$

The availability of such a support hyperplane suggests the possibility of approximating f from below, by the upper envelope of support hyperplanes.

The Kelley's cutting plane algorithm exploits this idea by building and improving a lower bounding function until some convergence criterion is met.

If S is polyhedral, we have to solve a sequence of LPs.



Step 0. Let $\mathbf{x}^1 \in S$ be an initial feasible solution; initialize the iteration counter $k \leftarrow 0$, the upper bound $u_0 = f(\mathbf{x}^1)$, the lower bound $l_0 = -\infty$, and the lower bounding function $\beta_0(\mathbf{x}) = -\infty$.

Step 1. Increment the iteration counter $k \leftarrow k+1$. Find a subgradient of f at $\mathbf{x}^k$, such that equation (5) and condition (6) hold.

Step 2. Update the upper bound

$$u_k = \min\{u_{k-1}, f(\mathbf{x}^k)\}$$

and the lower bounding function

$$\beta_k(\mathbf{x}) = \max\{\beta_{k-1}(\mathbf{x}), \alpha_k + \boldsymbol{\gamma}_k^\top \mathbf{x}\}.$$

Step 3. Solve the problem $l_k = \min_{\mathbf{x}\in S} \beta_k(\mathbf{x})$, and let $\mathbf{x}^{k+1}$ be the optimal solution.

Step 4. If $u_k - l_k < \epsilon$, stop: $\mathbf{x}^{k+1}$ is a satisfactory approximation of the optimal solution; otherwise, go to step 1.

Let us consider the two-stage SLP with recourse

$$\begin{aligned} \text{min} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} + \mathbb{E}_{\boldsymbol{\xi}}\left[Q(\mathbf{x}, \boldsymbol{\xi})\right] \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

where

$$Q(\mathbf{x}, \boldsymbol{\xi}) \equiv \min_{\mathbf{y}} \left\{ \mathbf{q}(\boldsymbol{\xi})^{\mathsf{T}}\mathbf{y} \mid \mathbf{W}(\boldsymbol{\xi})\mathbf{y} = \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}(\boldsymbol{\xi})\mathbf{x}, \mathbf{y} \geq \mathbf{0} \right\}.$$

We speak of *fixed recourse* when $\mathbf{W}$ deterministic, and of *random recourse* in general.

Let us consider the recourse function $\mathcal{Q}(\mathbf{x}) \equiv \mathbb{E}\left[Q(\mathbf{x}, \xi)\right]$ in the case of deterministic recourse cost:

$$Q(\mathbf{x}, \xi) \equiv \min_{\mathbf{y}} \left\{ \mathbf{q}^{\mathsf{T}}\mathbf{y} \mid \mathbf{W}\mathbf{y} = \mathbf{h}(\xi) - \mathbf{T}(\xi)\mathbf{x}, \mathbf{y} \geq \mathbf{0} \right\}$$

The effective domain of $\mathcal{Q}(\mathbf{x})$ consists of vectors $\mathbf{x}$ that are feasible for the first-stage constraints and such that $\mathcal{Q}(\mathbf{x}) < +\infty$, i.e., the second stage problem is (almost surely) feasible.

Note: we assume that, barring second-stage infeasibility, the above expectation is always defined, i.e., random variables are not heavy-tailed.

By LP duality, we have $Q(\mathbf{x}, \xi) = \max_{\boldsymbol{\pi}} \left\{ [\mathbf{h}(\xi) - \mathbf{T}(\xi)\mathbf{x}]^{\mathsf{T}}\boldsymbol{\pi} \mid \mathbf{W}^{\mathsf{T}}\boldsymbol{\pi} \le \mathbf{q} \right\}$.

Let us denote the feasible set of the dual by $\Pi = \{\boldsymbol{\pi} \mid \mathbf{W}^{\mathsf{T}}\boldsymbol{\pi} \le \mathbf{q}\}$. Note that, since $\mathbf{q}$ is deterministic, that this set is nonrandom.

Let us consider $\mathbf{x}^1, \mathbf{x}^2 \in \mathcal{D}$, $\lambda \in [0, 1]$, and $\mathbf{x}_\lambda = \lambda\mathbf{x}^1 + (1 - \lambda)\mathbf{x}^2$.

$$
\begin{aligned}
\mathcal{Q}(\mathbf{x}_\lambda) &= \int_\Xi Q(\mathbf{x}_\lambda, \xi)\, dP \\
&= \int_\Xi \max_{\boldsymbol{\pi}\in\Pi} \left\{ [\mathbf{h}(\xi) - \mathbf{T}(\xi)\mathbf{x}_\lambda]^{\mathsf{T}}\boldsymbol{\pi} \right\} dP \\
&= \int_\Xi \max_{\boldsymbol{\pi}\in\Pi} \left\{ \lambda \left[\mathbf{h}(\xi) - \mathbf{T}(\xi)\mathbf{x}^1\right]^{\mathsf{T}}\boldsymbol{\pi} + (1-\lambda) \left[\mathbf{h}(\xi) - \mathbf{T}(\xi)\mathbf{x}^2\right]^{\mathsf{T}}\boldsymbol{\pi} \right\} dP \\
&\le \lambda \int_\Xi \max_{\boldsymbol{\pi}\in\Pi} \left\{ \left[\mathbf{h}(\xi) - \mathbf{T}(\xi)\mathbf{x}^1\right]^{\mathsf{T}}\boldsymbol{\pi} \right\} dP \\
&\qquad + (1-\lambda) \int_\Xi \max_{\boldsymbol{\pi}\in\Pi} \left\{ \left[\mathbf{h}(\xi) - \mathbf{T}(\xi)\mathbf{x}^2\right]^{\mathsf{T}}\boldsymbol{\pi} \right\} dP \\
&= \lambda\mathcal{Q}(\mathbf{x}^1) + (1-\lambda)\mathcal{Q}(\mathbf{x}^2)
\end{aligned}
$$

The previous proof is rather simple, because the dual feasible region is nonrandom. Actually, convexity of the recourse function can be shown in a more general setting.

To cut a long story short (and cutting a few technical corners along the way...):

- The recourse function is typically differentiable for continuous probability distributions.

- The recourse function is polyhedral for discrete probability distributions.

In both cases, we may rely on Kelley's cutting planes to solve the problem.

The bonus, given problem structure, is that we may find cutting planes by a decomposition strategy.

Consider the deterministic equivalent:

$$\begin{aligned}
\text{min} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} + \sum_{s \in S} p^s \mathbf{q}_s^{\mathsf{T}} \mathbf{y}^s \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\
& \mathbf{W}\mathbf{y}_s + \mathbf{T}_s\mathbf{x} = \mathbf{h}_s \qquad \forall s \in S \\
& \mathbf{x}, \mathbf{y}_s \geq \mathbf{0}
\end{aligned}$$

The technological matrix of the overall problem has a *dual block angular* structure:

$$\begin{bmatrix}
\mathbf{A} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{T}_1 & \mathbf{W} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{T}_2 & \mathbf{0} & \mathbf{W} & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{T}_S & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{W}
\end{bmatrix}.$$

For a given first-stage, we get independent second-stage problems (dual structure of the one used in Lagrangian decomposition).

Let us rewrite the two-stage LP as:

$$
\begin{aligned}
\text{min} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} + \theta \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\
& \theta \geq \mathcal{Q}(\mathbf{x}) \qquad (7) \\
& \mathbf{x} \geq \mathbf{0}.
\end{aligned}
$$

We form a relaxed master problem by relaxing the constraint (7), which is approximated by cutting planes:

- Optimality cuts of form $\theta \geq \boldsymbol{\alpha}^{\mathsf{T}}\mathbf{x} + \beta$.

- Feasibility cuts of form $0 \geq \boldsymbol{\alpha}^{\mathsf{T}}\mathbf{x} + \beta$.

The coefficients of each cut are obtained by solving the scenario subproblems for given first-stage decisions.

Let $\hat{\mathbf{x}}$ be the optimal solution of the initial master problem and consider the dual of the second-stage problem for scenario s:

$$\begin{aligned} Q_s(\hat{\mathbf{x}}) \equiv \max \quad & (\mathbf{h}_s - \mathbf{T}_s\hat{\mathbf{x}})^{\mathsf{T}} \boldsymbol{\pi}_s \\ \text{s.t.} \quad & \mathbf{W}^{\mathsf{T}} \boldsymbol{\pi}_s \leq \mathbf{q}_s. \end{aligned} \tag{8}$$

Given an optimal dual solution $\hat{\boldsymbol{\pi}}_s$, it is easy to see that the following relationships hold:

$$Q_s(\hat{\mathbf{x}}) = (\mathbf{h}_s - \mathbf{T}_s\hat{\mathbf{x}})^{\mathsf{T}} \hat{\boldsymbol{\pi}}_s \tag{9}$$

$$Q_s(\mathbf{x}) \geq (\mathbf{h}_s - \mathbf{T}_s\mathbf{x})^{\mathsf{T}} \hat{\boldsymbol{\pi}}_s \qquad \forall \mathbf{x}. \tag{10}$$

The inequality (10) derives from the fact that $\hat{\boldsymbol{\pi}}_s$ is the optimal dual solution corresponding to $\hat{\mathbf{x}}$, but not to a generic $\mathbf{x}$.

Summing (10) over the scenarios, we get

$$\mathcal{Q}(\mathbf{x}) = \sum_{s \in S} p_s Q_s(\mathbf{x}) \geq \sum_{s \in S} p_s \left(\mathbf{h}_s - \mathbf{T}_s\mathbf{x}\right)^{\mathsf{T}} \hat{\boldsymbol{\pi}}_s.$$

Hence, we may add the following optimality cut to the relaxed master

$$\theta \geq \sum_{s \in S} p_s \left(\mathbf{h}_s - \mathbf{T}_s\mathbf{x}\right)^{\mathsf{T}} \hat{\boldsymbol{\pi}}_s.$$

If the recourse is not complete, some of scenario subproblems may be infeasible for a certain first-stage decision $\hat{\mathbf{x}}$. In this case, we may again exploit the dual of the scenario subproblem to find a feasibility cut, eliminating $\hat{\mathbf{x}}$ from further consideration.

Note that the feasibility region of this dual does not depend on the first-stage decisions, since $\hat{\mathbf{x}}$ does not occur in constraints (8). Thus, if a dual problem is infeasible, it means that the second-stage problem for the corresponding scenario will be infeasible for any first-stage decision.

Ruling out this case, which is likely to be due to a modeling error, when the primal problem is infeasible, the dual will be unbounded.

Hence, there is an extreme ray of the dual feasible set along which the optimal dual solution goes to infinity.

If we consider the dual of the second-stage subproblem, we see that it is unbounded if we find dual variables $\boldsymbol{\pi}^*$ such that

$$\mathbf{W}^{\mathsf{T}}\boldsymbol{\pi}^* \leq \mathbf{0}, \qquad (\mathbf{h}_s - \mathbf{T}_s\hat{\mathbf{x}})^{\mathsf{T}}\boldsymbol{\pi}^* > 0$$

We may derive a condition on $\mathbf{x}$, preventing this to occur. Primal feasibility requires

$$\mathbf{W}\mathbf{y} = \mathbf{h}_s - \mathbf{T}_s\mathbf{x} \quad \Rightarrow \quad \underbrace{\boldsymbol{\pi}^{*T}\mathbf{W}}_{\leq \mathbf{0}}\underbrace{\mathbf{y}}_{\geq \mathbf{0}} = \boldsymbol{\pi}^{*T}(\mathbf{h}_s - \mathbf{T}_s\mathbf{x}) \leq \mathbf{0}$$

Hence $\boldsymbol{\pi}^{*T}(\mathbf{h}_s - \mathbf{T}_s\mathbf{x}) \leq \mathbf{0}$ is a valid cut that is added to the relaxed master.

L-shaped decomposition iterates between:

1. the solution of the relaxed master problem, which yields $\widehat{\theta}$ and $\hat{\mathbf{x}}$;

2. the solution of the corresponding scenario subproblems.

At each iteration, cuts are added to the master problem.

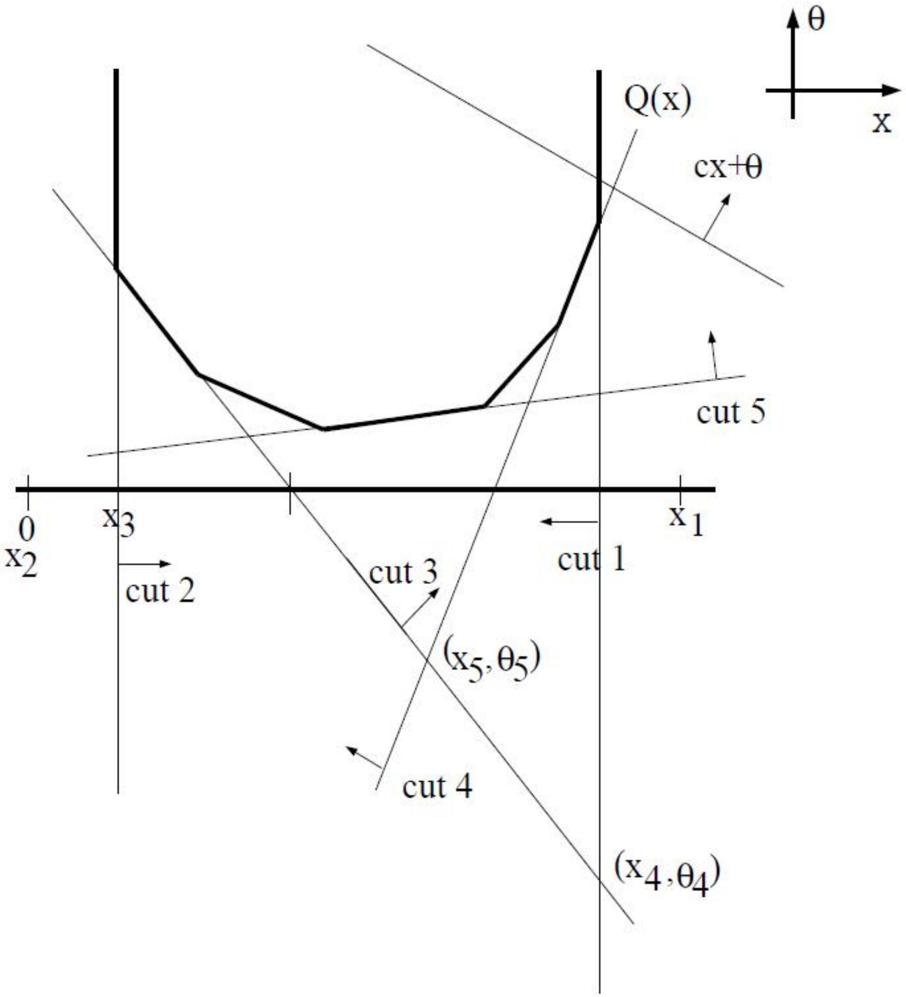
The algorithm stops when the optimal solution of the master problem satisfies

$$\widehat{\theta} \geq \mathcal{Q}(\hat{\mathbf{x}}).$$

DA RIVEDERE BENE, DIRE LB E UB

This condition may be relaxed if a near-optimal solution is good enough for our purposes.

An illustration, from (Kall, Wallace, 1994)



- There are variants of L-shaped decomposition to manage cuts and to improve convergence (e.g., regularized L-shaped decomposition).
- The idea can be generalized to multi-stage problems:
  - nested Benders decomposition
  - abridged nested Benders decomposition
  - stochastic dual dynamic programming
- In stochastic decomposition methods (Higle and Sen), there is an interplay between sampling and optimization: the idea is to build asymptotically valid cuts.
- Alternative strategies rely on bounding the recourse function from above and below, partitioning the domain in order to get tight bounds.
- When simplex or interior point methods manage to solve a stochastic LP, there is usually no advantage in using decomposition; however, decomposition is needed for really large-scale problems.

Part 3

Dantzig–Wolfe decomposition and column generation

We must cut rolls (all with width L) of some material into shorter rolls, in order to satisfy a given demand. Say that we introduce a set of n items, featuring demand d_i and width $w_i \le L$, $i \in [n]$. Our aim is to pack items in such a way that we use the minimum amount of rolls.

This is a stylized example of a large class of problems known under the common umbrella of cutting-stock problems.

Say that m rolls are available, indexed by $j \in [m]$, and let us introduce decision variables:

- $\delta_j \in \{0, 1\}$, set to 1 if we use roll j;
- $x_{ij} \in \mathbb{Z}_+$, the number of items i cut from roll j.

Then, we could consider the following MILP model:

$$
\begin{aligned}
\min \quad & \sum_{j \in [m]} \delta_j \\
\text{s.t.} \quad & \sum_{i \in [n]} w_i x_{ij} \le L\delta_j, && j \in [m] \\
& \sum_{j \in [m]} x_{ij} = d_i, && i \in [n] \\
& \delta_j \in \{0, 1\},\ x_{ij} \in \mathbb{Z}_+, && i \in [n], j \in [m].
\end{aligned}
$$

While there is nothing theoretically wrong with the model, it suffers from a few difficulties related with big-M constraints and symmetry issues (multiple optima may have a detrimental effect on commercial solvers).

We may resort to a different modeling framework based on a set of prespecified cutting patterns. Each cutting pattern, indexed by k, is associated with the number a_{ik} of items i that are obtained from the roll. Given a set $\mathcal{K}$ of cutting patterns, we may introduce the integer number γ_k or rolls cut according to that pattern, and solve the problem:

$$
\begin{aligned}
\min \quad & \sum_{k \in \mathcal{K}} \gamma_k \\
\text{s.t.} \quad & \sum_{k \in \mathcal{K}} a_{ik} \gamma_k = d_i, \qquad i \in [n] \\
& \gamma_k \in \mathbb{Z}_+, \qquad k \in \mathcal{K}.
\end{aligned}
$$

This solves the symmetry issue, but the cardinality of set $\mathcal{K}$ is huge. In fact, only a limited number of cutting patterns are really useful, but there is no obvious way to restrict set $\mathcal{K}$ a priori.

Thus, we may start with a limited number of patterns, and dynamically generate cutting patterns, following a column generation scheme.

Let us consider an LP-relaxation of a restricted version of the cutting stock model:

$$\begin{aligned} \text{min} \quad & \sum_{k\in\mathcal{K}'} \gamma_k \\ \text{s.t.} \quad & \sum_{k\in\mathcal{K}'} a_{ik}\gamma_k = d_i, \qquad i \in [n] \\ & \gamma_k \geq 0, \qquad k \in \mathcal{K}', \end{aligned} \tag{11}$$

where $\mathcal{K}' \subset \mathcal{K}$. How can we improve the solution by introducing a new column?

Let π_i, $i \in [n]$, denote the dual variables of constraints (11). We know from the theory behind the primal simplex algorithm that we should bring into the basis a variable with negative reduced cost.

In our case, the reduced cost for an alternative pattern k is

$$\widehat{c}_k = 1 - \sum_{i\in[n]} \pi_i a_{ik},$$

and we should find the column q with minimum reduced cost, $\widehat{c}_q = \min_k \widehat{c}_k$. If $\widehat{c}_q \geq 0$, then the current basis is optimal. Otherwise, we introduce the non-basic variable γ_q into the basis and we iterate by pivoting to another basis.

The minimization can be accomplished by solving the integer knapsack problem

$$\begin{aligned} \max \quad & \sum_{i \in [n]} \pi_i y_i \\ \text{s.t.} \quad & \sum_{i \in [n]} w_i y_i \leq L \\ & y_i \in \mathbb{Z}_+. \end{aligned}$$

This gives a new pattern, which is introduced into the set of columns if the value of the optimal solution is larger than 1.

This provides a lower bound for a branch-and-price algorithm. Alternatively, we may generate a set of columns, then restore integrality of the decision variables γ_k and solve a single integer program (which is a heuristic).

Note that, in a less stylized case, the complexity of constraints may be hidden in the column generation subproblem. The column generation scheme is at the heart of Dantzig–Wolfe decomposition.

Consider an LP model

$$\begin{aligned} \min \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} && (12) \\ & \mathbf{D}\mathbf{x} \geq \mathbf{f} && (13) \\ & \mathbf{x} \geq \mathbf{0}_n. \end{aligned}$$

We split the constraints, as we are going to interpret (12) as the nasty, complicating constraints, whereas (13), along with the non-negativity restriction, is the set of easy constraints. This means that minimizing any linear function over the polyhedron

$$\mathcal{X} \doteq \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{D}\mathbf{x} \geq \mathbf{f}, \mathbf{x} \geq \mathbf{0}_n \right\}$$

is easily accomplished. This may happen because of a decomposition into independent subproblems (i.e., $\mathbf{D}$ is block-diagonal) or because of favorable structure, e.g., a network flow structure. Whatever the reason, we know from Minkowski–Weyl theorem that we may represent $\mathcal{X}$ as the sum of a polytope and a cone:

$$\mathcal{X} = \mathsf{conv}\{\mathcal{V}\} + \mathsf{cone}\{\mathcal{D}\},$$

where $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_Q\}$ is the set of extreme points (vertices) of $\mathcal{X}$ and $\mathcal{D}\{\mathbf{d}_1, \ldots, \mathbf{d}_R\}$ is the set of its extreme directions.

Hence, the original model

$$\begin{aligned} \min \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \in \mathcal{X}, \end{aligned}$$

may be rewritten as follows. We express the constraint $\mathbf{x} \in \mathcal{X}$ as

$$\mathbf{x} = \sum_{q \in \mathcal{V}} \lambda_q \mathbf{v}_q + \sum_{r \in \mathcal{D}} \mu_r \mathbf{d}_r \tag{14}$$

$$\sum_{q \in \mathcal{V}} \lambda_q = 1 \tag{15}$$

$$\lambda_q \geq 0,\ q \in \mathcal{V}; \quad \mu_r \geq 0,\ r \in \mathcal{D}, \tag{16}$$

where (15) is the convexity constraint. We may plug (14) into the original model and obtain

$$\begin{aligned} \min \quad & \mathbf{c}^{\mathsf{T}} \left(\sum_{q \in \mathcal{V}} \lambda_q \mathbf{v}_q + \sum_{r \in \mathcal{D}} \mu_r \mathbf{d}_r \right) \\ \text{s.t.} \quad & \mathbf{A} \left(\sum_{q \in \mathcal{V}} \lambda_q \mathbf{v}_q + \sum_{r \in \mathcal{D}} \mu_r \mathbf{d}_r \right) \geq \mathbf{b} \\ & (15), (16). \end{aligned}$$

Note that this is an optimization model with respect to the decision variables λ_q and

μ_r. If we define the following scalars and vectors,

$$c_q \doteq \mathbf{c}^{\mathsf{T}}\mathbf{v}_q, \quad \mathbf{a}_q \doteq \mathbf{A}\mathbf{v}_q, \qquad q \in \mathcal{V}$$
$$c_r \doteq \mathbf{c}^{\mathsf{T}}\mathbf{d}_r, \quad \mathbf{a}_r \doteq \mathbf{A}\mathbf{d}_r, \qquad r \in \mathcal{D},$$

we may recast the original problem in the following form, referred to as the **master problem** MP:

$$\begin{aligned}
\text{(MP)} \qquad \min \quad & \sum_{q\in\mathcal{V}} c_q\lambda_q + \sum_{r\in\mathcal{D}} c_r\mu_r \\
\text{s.t.} \quad & \sum_{q\in\mathcal{V}} \mathbf{a}_q\lambda_q + \sum_{r\in\mathcal{D}} \mathbf{a}_r\mu_r \geq \mathbf{b} \\
& \sum_{q\in\mathcal{V}} \lambda_q = 1 \\
& \lambda_q \geq 0,\ q \in \mathcal{V}; \quad \mu_r \geq 0,\ r \in \mathcal{D}.
\end{aligned}$$

The clear trouble with the master problem is that it may involve a huge number of decision variables. Moreover, it may not be possible, given the description of $\mathcal{X}$ in terms of inequality constraints, to find its full set of extreme points and extreme directions.

The solution strategy relies on column generation. We initialize a **Restricted Master Problem** RMP with a subset $\mathcal{V}' \subset \mathcal{V}$ of extreme points and a subset $\mathcal{D}' \subset \mathcal{D}$ of extreme directions:

$$
\begin{aligned}
\text{(RMP)} \qquad \min \quad & \sum_{q \in \mathcal{V}'} c_q \lambda_q + \sum_{r \in \mathcal{D}'} c_r \mu_r \\
\text{s.t.} \quad & \sum_{q \in \mathcal{V}'} \mathbf{a}_q \lambda_q + \sum_{r \in \mathcal{D}'} \mathbf{a}_r \mu_r \geq \mathbf{b} && (17) \\
& \sum_{q \in \mathcal{V}'} \lambda_q = 1 && (18) \\
& \lambda_q \geq 0, \ q \in \mathcal{V}, \quad \mu_r \geq 0, \ r \in \mathcal{D}.
\end{aligned}
$$

Let us denote the dual variables associated with constraints (17) and (18) by $\boldsymbol{\pi}$ and π_0, respectively.

In the standard simplex method, we would bring into the basis a column if its reduced cost is negative. For extreme points, this means checking whether there is any $q \in \mathcal{Q}$ such that

$$
\bar{c}_q = c_q - \begin{bmatrix} \boldsymbol{\pi}^\mathsf{T} & \pi_0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_q \\ 1 \end{bmatrix} = c_q - \boldsymbol{\pi}^\mathsf{T} \mathbf{a}_q - \pi_0 = c_q - \boldsymbol{\pi}^\mathsf{T} \mathbf{A} \mathbf{v}_q - \pi_0 < 0.
$$

A similar check should be carried out for extreme directions:

$$
\bar{c}_r = c_r - \begin{bmatrix} \boldsymbol{\pi}^\mathsf{T} & \pi_0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_r \\ 0 \end{bmatrix} = c_r - \boldsymbol{\pi}^\mathsf{T} \mathbf{a}_r = c_r - \boldsymbol{\pi}^\mathsf{T} \mathbf{A} \mathbf{d}_r < 0.
$$

Hence, we should solve the following auxiliary problem,

$$\min \left\{ \min_{q \in \mathcal{V}} \overline{c}_q, \min_{r \in \mathcal{D}} \overline{c}_r, \right\}.$$

Clearly, this cannot be tackled by complete enumeration, but we may recast it into the form of the following **pricing problem** PP:

$$\begin{array}{lll} \text{(PP)} \qquad & \min & \left(\mathbf{c}^{\mathsf{T}} - \pi^{\mathsf{T}}\mathbf{A}\right)\mathbf{x} \\ & \text{s.t.} & \mathbf{D}\mathbf{x} \geq \mathbf{d} \\ & & \mathbf{x} \geq \mathbf{0}_n. \end{array}$$

Note that we are disregarding the constant term π_0, and use the same problem to find a vector $\mathbf{x} \in \mathbb{R}^n$ that will play the role of an extreme point or extreme direction, depending on the following cases, where z^*_{PP} denotes the optimal value of PP:

- If $z^*_{\mathsf{PP}} = -\infty$, i.e., PP is unbounded below, we have found an extreme direction of $\mathcal{X}$, say $\mathbf{x}^*$, with negative reduced cost (commercial LP solvers provide us with this information). Then, we include a new variable μ_r into RMP, associated with the with column

$$\begin{bmatrix} \mathbf{A}\mathbf{x}^* \\ 0 \end{bmatrix}.$$

- If $-\infty < z^*_{\mathsf{PP}} < \pi_0$, i.e., the PP is bounded and its value is less than the dual variable of the convexity constraint, we have discovered a new extreme point, say $\mathbf{x}^*$, with negative reduced cost. Then, we include a new variable λ_q into RMP, associated with column

$$\begin{bmatrix} \mathbf{A}\mathbf{x}^* \\ 1 \end{bmatrix}.$$

- If $z^*_{\mathsf{PP}} \geq \pi_0$, then there is no useful extreme point or direction to add to RMP, and we may stop.

In intuitive terms, the optimal dual variables from solving RMP suggest a direction $\left(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\pi\right)$ along which we should "probe" the polyhedron $\mathcal{X}$. Clearly, if this is a polytope, we can only find an extreme point.

In the case of an unbounded polyhedron, we may find an extreme direction.

We do not need to find all of them, as only those who have a negative reduced cost are interesting, as they can improve the value we obtain from solving RMP.