Multistage stochastic linear programming problems

Stochastic dual dynamic programming

Welington de Oliveira

BAS Lecture 19, May 12, 2016, IMPA

Multistage stochastic linear programs - T-SLP

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathbb{E}\left[\min_{\substack{B_2 x_1 + A_2 x_2 = b_2 \\ x_2 \geq 0}} c_2^\top x_2 + \mathbb{E}\left[\cdots + \mathbb{E}[\min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T]\right]\right]$$

- Some elements of the data $\xi = (c_t, B_t, A_t, b_t)$ depend on uncertainties.

Dynamic programming formulation

Time (stage) dependent stochastic process $\xi = (\xi_1, \xi_2, \ldots, \xi_T)$

- Stage $t = T$

$$Q_T(x_{T-1}, \xi_{[T]}) := \min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T$$

- At stages $t = 2, \ldots, T-1$

$$Q_t(x_{t-1}, \xi_{[t]}) := \min_{\substack{B_t x_{t-1} + A_t x_t = b_t \\ x_t \geq 0}} c_t^\top x_t + \mathcal{Q}_{t+1}(x_t, \xi_{[t]})$$

- Stage $t = 1$

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathcal{Q}_2(x_1, \xi_{[1]})$$

Recourse function

$$\mathcal{Q}_{t+1}(x_t, \xi_{[t]}) := \mathbb{E}_{|\xi_{[t]}} \left[Q_{t+1}(x_t, \xi_{[t+1]}) \right]$$

Dynamic programming formulation

Stage-wise independent process $\xi = (\xi_1, \xi_2, \ldots, \xi_T)$

- Stage $t = T$

$$Q_T(x_{T-1}, \xi_t) := \min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T$$

- At stages $t = 2, \ldots, T-1$

$$Q_t(x_{t-1}, \xi_t) := \min_{\substack{B_t x_{t-1} + A_t x_t = b_t \\ x_t \geq 0}} c_t^\top x_t + \mathcal{Q}_{t+1}(x_t)$$

- Stage $t = 1$

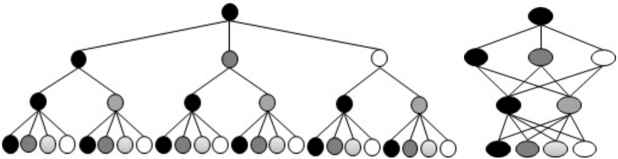
$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathcal{Q}_2(x_1)$$

Recourse function

$$\mathcal{Q}_{t+1}(x_t) := \mathbb{E}_{|\xi_t} \left[Q_{t+1}(x_t, \xi_{t+1}) \right]$$

Stagewise independence

A stochastic process is stagewise independent if $P(\xi_t|\xi_{[t-1]}) = P(\xi_t)$



Figures by Felipe B. Rodríguez

Every nodes at the same stage share the same set of children nodes.

Time (stage) dependent stochastic process

Sometimes it is possible to change the problem's formulation in order to obtain an equivalent problem but with stage-wise stochastic process.

Example

Suppose that all the problem's data except b_t is stage-wise independent (in particular c_t, B_t and A_t are deterministic). Moreover, assume that random vectors b_t, $t = 2, \ldots, T$, form a first order autoregressive process, i.e., $b_t = \phi_t b_{t-1} + \epsilon_t$, with error vectors $\epsilon_2, \ldots, \epsilon_t$ being independent of each other.

Time (stage) dependent stochastic process

Sometimes it is possible to change the problem's formulation in order to obtain an equivalent problem but with stage-wise stochastic process.

Example

Suppose that all the problem's data except b_t is stage-wise independent (in particular c_t, B_t and A_t are deterministic). Moreover, assume that random vectors b_t, $t = 2, \dots, T$, form a first order autoregressive process, i.e., $b_t = \phi_t b_{t-1} + \epsilon_t$, with error vectors $\epsilon_2, \dots, \epsilon_t$ being independent of each other. Then

$$Q_t(x_{t-1}, \xi_{[t]}) = \min_{\substack{B_t x_{t-1} + A_t x_t = b_t \\ x_t \geq 0}} c_t^\top x_t + \mathcal{Q}_{t+1}(x_t, \xi_{[t]})$$

can be rewritten by

$$Q_t(x_{t-1}, b_{t-1}, \tilde{\xi}_{[t]}) = \begin{cases} \min\limits_{x_t \geq 0,\, b_t} & c_t^\top x_t + \mathcal{Q}_{t+1}(x_t, b_t, \tilde{\xi}_{[t]}) \\ \text{s.t.} & B_t x_{t-1} + A_t x_t = \phi_t b_{t-1} + \epsilon_t \\ & b_t = \phi_t b_{t-1} + \epsilon_t \end{cases}$$

$\tilde{\xi}_t := (c_t, B_t, A_t, \epsilon_t)$ is stage-wise independent.
$\tilde{x}_t := (x_t, b_t)$ becomes a variable.

Time (stage) dependent stochastic process

Sometimes it is possible to change the problem's formulation in order to obtain an equivalent problem but with stage-wise stochastic process.

Example

Suppose that all the problem's data except b_t is stage-wise independent (in particular c_t, B_t and A_t are deterministic). Moreover, assume that random vectors b_t, $t = 2, \dots, T$, form a first order autoregressive process, i.e., $b_t = \phi_t b_{t-1} + \epsilon_t$, with error vectors $\epsilon_2, \dots, \epsilon_t$ being independent of each other. Then

$$Q_t(x_{t-1}, \xi_{[t]}) = \min_{\substack{B_t x_{t-1} + A_t x_t = b_t \\ x_t \geq 0}} c_t^\top x_t + \mathcal{Q}_{t+1}(x_t, \xi_{[t]})$$

can be rewritten by

$$Q_t(\tilde{x}_{t-1}, \tilde{\xi}_t) = \begin{cases} \min\limits_{x_t \geq 0,\, b_t} & c_t^\top x_t + \mathcal{Q}_{t+1}(\tilde{x}_t) \\ \text{s.t.} & B_t x_{t-1} + A_t x_t = \phi_t b_{t-1} + \epsilon_t \\ & b_t = \phi_t b_{t-1} + \epsilon_t \end{cases}$$

$\tilde{\xi}_t := (c_t, B_t, A_t, \epsilon_t)$ is stage-wise independent.
$\tilde{x}_t := (x_t, b_t)$ becomes a variable.

Cutting-plane approximation

- Stage $t = T$

$$Q_T(x_{T-1}^k, \xi_T) := \min_{\substack{B_T x_{T-1}^k + A_T x_T = b_T \\ x_T \geq 0}} c_T{}^\top x_T$$

- At stages $t = 2, \ldots, T-1$

$$\underline{Q_t}(x_{t-1}^k, \xi_t) := \begin{cases} \min\limits_{x_t \geq 0, r_{t+1}} & c_t{}^\top x_t + r_{t+1} \\ \text{s.t.} & B_t x_{t-1}^k + A_t x_t = b_t \\ & r_{t+1} \geq \alpha_{t+1}^j + \beta_{t+1}^j x_t \quad j = 1, \ldots, k \end{cases}$$

- Stage $t = 1$

$$\underline{z}^k := \begin{cases} \min\limits_{x_1 \geq 0, r_2} & c_1^\top x_1 + r_2 \\ \text{s.t.} & A_1 x_1 = b_1 \\ & r_2 \geq \alpha_2^j + \beta_2^j x_1 \quad j = 1, \ldots, k \end{cases}$$

Cutting-plane approximation

- At stages $t = 2, \ldots, T-1$

$$\underline{Q_t(x_{t-1}^k, \xi_t)} := \begin{cases} \min\limits_{x_t \geq 0, r_{t+1}} & {c_t}^\top x_t + r_{t+1} & & \\ \text{s.t.} & B_t x_{t-1}^k + A_t x_t = b_t & & (\pi_t) \\ & r_{t+1} \geq \alpha_{t+1}^j + \beta_{t+1}^j x_t & j = 1, \ldots, k & (\rho_j) \end{cases}$$

- Cuts $(t = T)$

$$\alpha_T^k := \mathbb{E}_{|\xi_{T-1}}[b_T^\top \pi_T^k] \quad \text{and} \quad \beta_T^k := -\mathbb{E}_{|\xi_{T-1}}[B_T^\top \pi_T^k]$$

- Cuts $(t = T-1, \ldots, 2)$

$$\alpha_t^k := \mathbb{E}_{|\xi_{t-1}}[{b_t}^\top \pi_t^k + \sum_{j=1}^{k} \alpha_{t+1}^j \rho_k^j] \quad \text{and} \quad \beta_t^k := -\mathbb{E}_{|\xi_{t-1}}[{B_t}^\top \pi_t^k]$$

$$\check{\mathcal{Q}}_{t+1}^k(x_t) := \max_{j=1,\ldots,k} \{\alpha_{t+1}^k + {\beta_{t+1}^k}^\top x_t\} \quad \leq \quad \mathcal{Q}_{t+1}(x)$$

Nested decomposition

The Nested decomposition could be applied

$$\underline{Q}_t(x_{t-1}^k, \xi_t) = \min_{\substack{B_t x_{t-1}^k + A_t x_t = b_t \\ x_t \geq 0}} c_t^\top x_t + \check{\mathcal{Q}}_{t+1}^k(x_t)$$

- **Step 0: initialization**. Define $k = 1$ and add the constraint $r_t = 0$ in all LPs $\underline{Q}_t$, $t = 2, \ldots, T-1$. Compute $\underline{z}^1$ and let its solution be x_1^1.
- **Step 1: forward**. For t=2,...,T, solve the LP $\underline{Q}_t$ to obtain $x_t^k := x_t^k(\xi_t)$. Define $\bar{z}^k := \mathbb{E}[\sum_{t=1}^T c_t^\top x_t^k]$.
- **Step 2: backward**. Compute α_T^k and β_T^k. Set $t = T$. Loop:
  - While $t > 2$
  - $t \leftarrow t - 1$
  - solve the LP $\underline{Q}_t(x_{t-1}^k, \xi_t)$
  - Compute α_t^k and β_t^k

  Compute $\underline{z}^k$ and let its solution be x_1^{k+1}.
- **Step 3: Stopping test**. If $\bar{z}^k - \underline{z}^k \leq \epsilon$, stop. Otherwise set $k \leftarrow k + 1$ and go back to Step 1.

Nested decomposition

The Nested decomposition could be applied

$$\underline{Q}_t(x_{t-1}^k, \xi_t) = \min_{\substack{B_t x_{t-1}^k + A_t x_t = b_t \\ x_t \geq 0}} c_t{}^\top x_t + \check{\mathcal{Q}}_{t+1}^k(x_t)$$

- **Step 0: initialization**. Define $k = 1$ and add the constraint $r_t = 0$ in all LPs $\underline{Q}_t$, $t = 2, \dots, T-1$. Compute $\underline{z}^1$ and let its solution be x_1^1.
- **Step 1: forward**. For t=2,...,T, solve the LP $\underline{Q}_t$ to obtain $x_t^k := x_t^k(\xi_t)$. Define $\bar{z}^k := \mathbb{E}[\sum_{t=1}^T c_t^\top x_t^k]$.
- **Step 2: backward**. Compute α_T^k and β_T^k. Set $t = T$. Loop:
  - While $t > 2$
  - $t \leftarrow t - 1$
  - solve the LP $\underline{Q}_t(x_{t-1}^k, \xi_t)$
  - Compute α_t^k and β_t^k

  Compute $\underline{z}^k$ and let its solution be x_1^{k+1}.
- **Step 3: Stopping test**. If $\bar{z}^k - \underline{z}^k \leq \epsilon$, stop. Otherwise set $k \leftarrow k + 1$ and go back to Step 1.

But we aren't taking into consideration the stage-wise independence assumption!

Nested decomposition

The Nested decomposition could be applied

$$\underline{Q}_t(x_{t-1}^k, \xi_t) = \min_{\substack{B_t x_{t-1}^k + A_t x_t = b_t \\ x_t \geq 0}} c_t^\top x_t + \check{\mathcal{Q}}_{t+1}^k(x_t)$$

- **Step 0: initialization**. Define $k = 1$ and add the constraint $r_t = 0$ in all LPs $\underline{Q}_t$, $t = 2, \ldots, T-1$. Compute $\underline{z}^1$ and let its solution be x_1^1.
- **Step 1: forward**. For t=2,...,T, solve the LP $\underline{Q}_t$ to obtain $x_t^k := x_t^k(\xi_t)$. Define $\bar{z}^k := \mathbb{E}[\sum_{t=1}^T c_t^\top x_t^k]$.
- **Step 2: backward**. Compute α_T^k and β_T^k. Set $t = T$. Loop:
  - While $t > 2$
  - $t \leftarrow t - 1$
  - solve the LP $\underline{Q}_t(x_{t-1}^k, \xi_t)$
  - Compute α_t^k and β_t^k

  Compute $\underline{z}^k$ and let its solution be x_1^{k+1}.
- **Step 3: Stopping test**. If $\bar{z}^k - \underline{z}^k \leq \epsilon$, stop. Otherwise set $k \leftarrow k + 1$ and go back to Step 1.

But we aren't taking into consideration the stage-wise independence assumption! Can we do better?

Yes, we can!

Notice that:

- there exists only one recourse function $\mathcal{Q}_t$ per stage (thanks to the stage-wise independence)
- any chosen scenario in the tree "touches" $\mathcal{Q}_t$
- even if we choose only one scenario in the forward pass, the backward pass visits all the nodes

As a result, if we make a forward pass only on subset of scenarios (e.g. a single scenario), we may improve the cutting-plane models $\check{Q}_t$ with much less effort.

Yes, we can!

Notice that:

- there exists only one recourse function $\mathcal{Q}_t$ per stage (thanks to the stage-wise independence)
- any chosen scenario in the tree "touches" $\mathcal{Q}_t$
- even if we choose only one scenario in the forward pass, the backward pass visits all the nodes

As a result, if we make a forward pass only on subset of scenarios (e.g. a single scenario), we may improve the cutting-plane models $\check{Q}_t$ with much less effort.

This is main idea behind the Stochastic Dual Dynamic Programming (SDDP) method

SDDP algorithm

In addition to the stage-wise independence assumptions, let's assume that

- the recourse functions $\mathcal{Q}_t$ are finite valued (in particular we assume the relatively complete recourse)
- a finite scenario tree is considered.

Let $\mathcal{S}_K := \{\xi^1, \ldots, \xi^K\}$ be the set of scenarios representing the considered tree.

Forward pass

At iteration k, the forward step of the SDDP algorithm consists in choosing $M < K$ scenarios $\tilde{\mathcal{S}}_M^k := \{\tilde{\xi}^1, \ldots, \tilde{\xi}^M\}$ and computing the respective optimal values

$$\underline{Q_t}(x_{t-1}^k, \tilde{\xi}_t) = \min_{\substack{B_t x_{t-1}^k + A_t x_t = b_t \\ x_t \geq 0}} c_t{}^\top x_t + \check{\mathcal{Q}}_{t+1}^k(x_t)$$

for all[1] $t = 2, \ldots, T$ and all $\tilde{\xi} \in \tilde{\mathcal{S}}_M^k$.

Upper bound **estimate**

The value $\tilde{z}^k := \mathbb{E}_{\tilde{\mathcal{S}}_M^k}[\sum_{t=1}^T c_t^\top x_t^k]$ is not necessary an upper bound on the optimal value.

[1] We define $\check{\mathcal{Q}}_{T+1} \equiv 0$.

SDDP algorithm

Backward pass

As in the Nested decomposition, but considering only points x_t^k related to the sample set $\tilde{\mathcal{S}}_M^k$

Lower bound

$$\underline{z}^k = \min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1{}^\top x_1 + \check{\mathcal{Q}}_2^k(x_1)$$

is a lower bound for the optimal solution.

Stopping

The test $\tilde{z}^k - \underline{z}^k \leq$ tol does not make sense...
A more meaningful criterion would be to stop the algorithm if

$$(\tilde{z}^k + 1.96\hat{\sigma}/\sqrt{M}) - \underline{z}^k \leq \text{tol},$$

where $\hat{\sigma}^2$ is the sample variance of the costs related to the scenarios $\tilde{\xi} \in \tilde{\mathcal{S}}_M^k$, and 1.96 corresponds to the 95% confidence interval.
This will suggest, with confidence of about 95%, that the problem (with K scenarios) is solved with accuracy tol > 0.

Stochastic dual dynamic programming algorithm

A simple version

$$\underline{Q}_t(x_{t-1}^k, \xi_t) = \min_{\substack{B_t x_{t-1}^k + A_t x_t = b_t \\ x_t \geq 0}} c_t{}^\top x_t + \check{\mathcal{Q}}_{t+1}^k(x_t)$$

- **Step 0: initialization**. Define $k = 1$ and add the constraint $r_t = 0$ in all $(T-1)$ LPs $\underline{Q}_t$, $t = 2, \ldots, T-1$. Compute $\underline{z}^1$ and let its solution be x_1^1.

- **Step 1: forward**. Choose $1 \leq M \leq K$. Randomly draw (with replacement) a sample $\tilde{\mathcal{S}}_M^k$ out of $\mathcal{S}_K$. For all $\tilde{\xi} \in \tilde{\mathcal{S}}_M^k$ and all $t = 2, \ldots, T$, solve the LPs $\underline{Q}_t(\cdot, \tilde{\xi}_t)$ to obtain $x_t^k := x_t^k(\tilde{\xi}_t)$. Define $\tilde{z}^k := \mathbb{E}_{\tilde{\mathcal{S}}_M^k}[\sum_{t=1}^T c_t^\top x_t^k]$.

- **Step 2: backward**. Compute α_T^k and β_T^k. Set $t = T$. Loop:
  - While $t > 2$
  - $t \leftarrow t - 1$
  - solve the LP $\underline{Q}_t(x_{t-1}^k, \xi_t)$
  - Compute α_t^k and β_t^k

  Compute $\underline{z}^k$ and let its solution be x_1^{k+1}.

- **Step 3: Stopping test**. Stop if a given criteria is satisfied (e.g. maximum number of iterations). Otherwise set $k \leftarrow k + 1$ and go back to Step 1.

Some remarks on the SDDP

- The number M of scenarios visited during the forward pass my vary along iteration. Efficient SDDP algorithms start with $M = 1$ in the first forward passes and then increase M progressively (until a certain threshold)
- For convergence analysis purposes we assume that every scenario $\xi \in \mathcal{S}_K$ is, w.p.1, chosen infinitely many times by the forward pass if the algorithm loops forever. This happens, for instance, if $\tilde{\mathcal{S}}^k_M$ is randomly chosen.
- The algorithm may stop when the lower bound $\underline{z}^k$ stabilizes (and a minimum number of iteration is reached)
- An interesting property of the SDDP method is that the computational complexity of one run of the involved backward and forward step procedures is proportional to the sum of sampled data points at every stage and not to the total number of scenarios given by their product.
- The problem studied was risk neutral. However a lot of works has been done recently about how to solve risk averse problems Variants of the SDDP method have been applied to solve nonconvex stochastic programs

Convergence analysis

Propositon

Suppose that in the forward steps of the SDDP algorithm the subsampling procedure is used, and in the backward steps basic optimal dual solutions are employed to compute cuts. Furthermore, assume the problem has relatively complete recourse and that feasible sets are bounded for all $t = 1, \ldots, T$. Then w.p.1 after a sufficiently large number of backward and forward steps of the algorithm, the forward step procedure defines an optimal policy for the problem (defined by a finite scenario tree).

Reference:
Shapiro, A., "*Analysis of Stochastic Dual Dynamic Programming Method*", European Journal of Operational Research, vol. 209, pp. 63-72, 2011.

Available at `http://www2.isye.gatech.edu/people/faculty/Alex_Shapiro/EJOR-2011.pdf`