Part 6
Stochastic Programming

Introduction

What is Stochastic Programming?

- It is a way to deal with uncertainty in the parameters.
- Goal: Transformation to a so-called deterministic equivalent
- two kind of decision variables: anticipative and/or adaptive
- i.e. here-and-now versus wait-and-see
- multi-stage with recourse: anticipative and adaptive variables
- e.g. rebalancing of portfolio

Two-stage problems with recourse

$$\begin{array}{llclr} \max_x & a^T x & + & E[\max_{y(w)} c(w)^T y(w)] & \\ & Ax & & & = b \\ & B(w)x & + & C(w)y(w) & = d(w) \\ & x \geqslant 0, & & y(w) \geqslant 0. & \end{array}$$

$$\begin{array}{lccc} \max_x & a^T x & + & f(x) \\ & Ax = b & & \\ & x \geqslant 0. & & \end{array}$$

with $f(x) = E[f(x, w)]$ and

$$\begin{array}{llll} f(x, w) = & \max_{y(w)} & c(w)^T y(w) & \\ & & C(w)y(w) & = d(w) - B(w)x \\ & & y(w) \geqslant 0. & \end{array}$$

Two-stage problems with recourse and finite state space

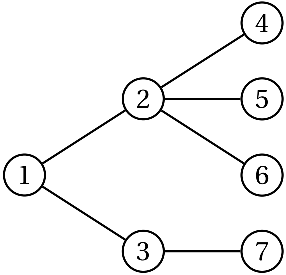
- Let $\Omega = \{\omega_1, \ldots, \omega_S\}$ with probabilities $p_1, \ldots, p_S$

$$
\begin{array}{llllll}
\max_{x,y_k} & a^T x & + & \sum_{k=1}^{S} p_k c_k^T y_k & & \\
& Ax & & & = b & \\
& B_k x & + & C_k y_k & = d_k & \text{for } k = 1, \ldots, S \\
& x \geqslant 0 & & & & \\
& & & y_k \geqslant 0 & & \text{for } k = 1, \ldots, S
\end{array}
$$

- $y_1, \ldots, y_k$ are independent.

Multi-stage

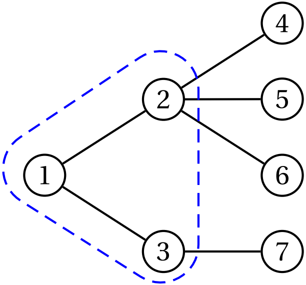
Scenario tree



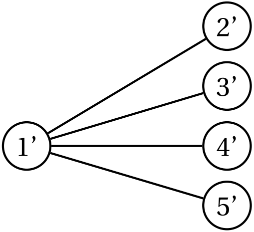
- $\{1\}$ root node
- $\{4,5,6,7\}$ terminal nodes
- Four scenarios
- Three stages
- $a(i)$ is the father of i
- scenario tree could be huge

$$\begin{array}{lccccl}
\max_{x_1,\ldots,x_N} & c_1^T x_1 & + & \sum_{i=2}^N q_i c_i^T x_i & & \\
 & Ax_1 & & & = b & \\
 & B_i x_{a(i)} & + & C_i x_i & = d_i & \text{for } i = 2,\ldots,N \\
 & & & x_i \geqslant 0 & &
\end{array}$$

From Multi-stage to Two-stage



$$\begin{pmatrix} A & & & & & & \\ B_2 & C_2 & & & & & \\ B_3 & & C_3 & & & & \\ & B_4 & & C_4 & & & \\ & B_5 & & & C_5 & & \\ & B_6 & & & & C_6 & \\ & & B_7 & & & & C_7 \end{pmatrix}$$



$$\begin{pmatrix} A' & & & & \\ B'_2 & C'_2 & & & \\ B'_3 & & C'_3 & & \\ B'_4 & & & C'_4 & \\ B'_5 & & & & C'_5 \end{pmatrix}$$

Benders decomposition

We exploit the structure of the two-stage problem

$$\begin{pmatrix} A & & & \\ B_1 & C_1 & & \\ \vdots & & \ddots & \\ B_S & & & C_S \end{pmatrix}$$

Recall

$$\begin{array}{llll} \max_x & a^T x & + & \sum_{k=1}^{S} P_k(x) \\ & Ax = b & & \\ & x \geqslant 0. & & \end{array}$$

with

$$\begin{array}{llll} P_k(x) = & \max_{y_k} & p_k c_k^T y_k & \\ & & C_k y_k & = d_k - B_k x \\ & & y_k \geqslant 0. & \end{array}$$

Cut generation

Obtain an initial solution for x, e.g. by solving

$$\begin{aligned} \max_x \quad & a^T x \\ & Ax = b \\ & x \geqslant 0. \end{aligned}$$

Given x^i of the i-th iteration, we compute

$$P_k(x^i) = \begin{aligned} \max_{y_k} \quad & p_k c_k^T y_k \\ & C_k y_k = d_k - B_k x^i \\ & y_k \geqslant 0 \end{aligned} \quad = \begin{aligned} \min_{u_k} \quad & u_k^T (d_k - B_k x^i) \\ & C_k^T u_k \geqslant p_k c_k \end{aligned}$$

- We iteratively compute solutions $x^0, x^1, x^2, \ldots$.
- The recourse problems $P_k(x^i)$ are independent for fixed x^i.
- Hence, they can be solved in parallel.
- If the dual is feasible, we obtain further constraints on x.

Optimality Cuts

Replace $P_k(x)$ by auxiliary variables z_k

$$\begin{array}{lrcl} \max_{x,z_1,\dots,z_S} & a^T x & + & \sum_{k=1}^S z_k \\ & Ax = b & & \\ & z_k & \leqslant & P_k(x^j) + (u_k^i)^T (B_k x^i - B_k x) \\ & x \geqslant 0. & & \end{array}$$

$$P_k(x^i) = \begin{array}{lc} \max_{y_k} & p_k c_k^T y_k \\ & C_k y_k = d_k - B_k x^i \\ & y_k \geqslant 0 \end{array} = \begin{array}{lc} \min_{u_k} & u_k^T (d_k - B_k x^i) \\ & C_k^T u_k \geqslant p_k c_k \end{array}$$

If $P_k(x^i)$ is finite with optimum dual solution u_k^i

$$\begin{aligned} P_k(x) \leqslant (u_k^i)^T (d_k - B_k x) &= (u_k^i)^T (d_k - B_k x^i + B_k x^i - B_k x) \\ &= P_k(x^i) + (u_k^i)^T (B_k x^i - B_k x) \end{aligned}$$

Feasibility Cuts

The refined first stage problem

$$\begin{array}{rcl} \max_{x,z_1,\dots,z_S} \quad a^T x & + & \sum_{k=1}^{S} z_k \\ Ax = b & & \\ z_k & \leqslant & P_k(x^j) + (u_k^i)^T (B_k x^i - B_k x) \\ 0 & \leqslant & (u_k^i)^T (d_k - B_k x) \\ x \geqslant 0. & & \end{array}$$

$$P_k(x^i) = \max_{y_k} \; p_k c_k^T y_k \quad C_k y_k = d_k - B_k x^i, \; y_k \geqslant 0 \quad = \min_{u_k} \; u_k^T (d_k - B_k x^i) \quad C_k^T u_k \geqslant p_k c_k$$

If the dual is unbounded in direction u_k^i, i.e.

$$(u_k^i)^T (d_k - B_k x^i) < 0 \qquad \text{and} \qquad C_k^T u_k^i \geqslant p_k c_k$$

Scenario Generation

- If the state space is too large or even infinite,
- we have to approximate by few samples, e.g.
- by random sampling,
- by tree fitting,
- such that the statistical properties of the sample are as close as possible to the ones of the original distribution (in particular the moments)
- Caution: The approximation might introduce modeling errors,
- e.g. create arbitrage opportunities.

Value-at-Risk (VaR)

Financial activities involve risk!

Popular risk measure by engineers at J.P. Morgan: **Value-at-Risk**

$$VaR_\alpha(X) := \inf\{\gamma : P(X > \gamma) \leqslant 1 - \alpha\}$$

where X is a random variable representing the loss from an investment portfolio.Continuous loss distribution:

$$P(X \leqslant VaR_\alpha(X)) = \alpha$$

It does not respect the paradigm "diversification reduces risk"

Mathematically: It lacks subadditivity $f(x_1 + x_2) \leqslant f(x_1) + f(x_2)$

loss X	2 CHF	−1 CHF	4 CHF	1 CHF	−2 CHF
$P(X)$	0.04	0.96	0.016	0.0768	0.9216
$VaR_{0.95}(X)$	−1 CHF			1 CHF	

Conditional Value-at-Risk (CVaR)

What about the magnitude of losses beyond VaR?

Modification: CVaR, a.k.a. mean expected loss, mean shortfall

$$\begin{aligned} CVaR_\alpha(X) &:= \frac{1}{1-\alpha}\int_\alpha^1 VaR_p(X)\,dp \\ &\geqslant \frac{1}{1-\alpha}\int_\alpha^1 VaR_\alpha(X)\,dp \\ &= \frac{VaR_\alpha(X)}{1-\alpha}\int_\alpha^1 dp \\ &= VaR_\alpha(X) \end{aligned}$$

Minimizing CVaR

Let $f(x,y)$ be the loss of a portfolio determined by $x \in X$ at the realization y of a random vector with probability $p(y)$.

$$CVaR_\alpha(x) = \frac{1}{1-\alpha}\int_{f(x,y)\geqslant VaR_\alpha(x)} f(x,y)p(y)dy$$

$$\begin{aligned} F_\alpha(x,\gamma) &:= \gamma + \tfrac{1}{1-\alpha}\int_{f(x,y)\geqslant\gamma}(f(x,y)-\gamma)p(y)dy \\ &= \gamma + \tfrac{1}{1-\alpha}\int \max\{0, f(x,y)-\gamma\}p(y)dy \end{aligned}$$

- $F_\alpha(x,\gamma)$ does not contain VaR_α.
- $F_\alpha(x,\gamma)$ is convex w.r.t. γ.
- $F_\alpha(x, VaR_\alpha(x)) = CVaR_\alpha(x)$ is minimum w.r.t. γ.

$$\min_{x\in X} CVaR_\alpha(x) = \min_{x\in X,\gamma} F_\alpha(x,\gamma)$$