CS 524: Introduction to Optimization
Lecture 20 : Multistage SP

Michael Ferris

Computer Sciences Department
University of Wisconsin-Madison

November 11, 2025

Two stage problems

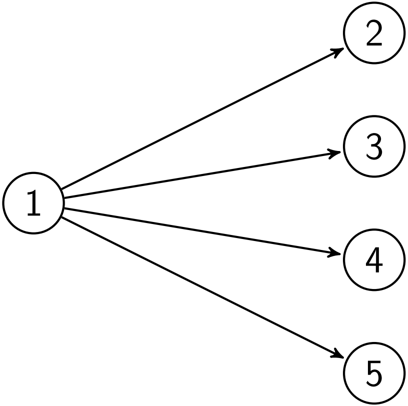
$$\min_{x \in X} c_1(x_1) + \mathbb{E}_1(Q_1(x_{\mathcal{S}(1)}, \omega_1))$$

- Expected Recourse Function:

$$\mathcal{Q}(x) \stackrel{\text{def}}{=} \mathbb{E}_\omega[Q(x, \omega)]$$

- Two-Stage Stochastic LP

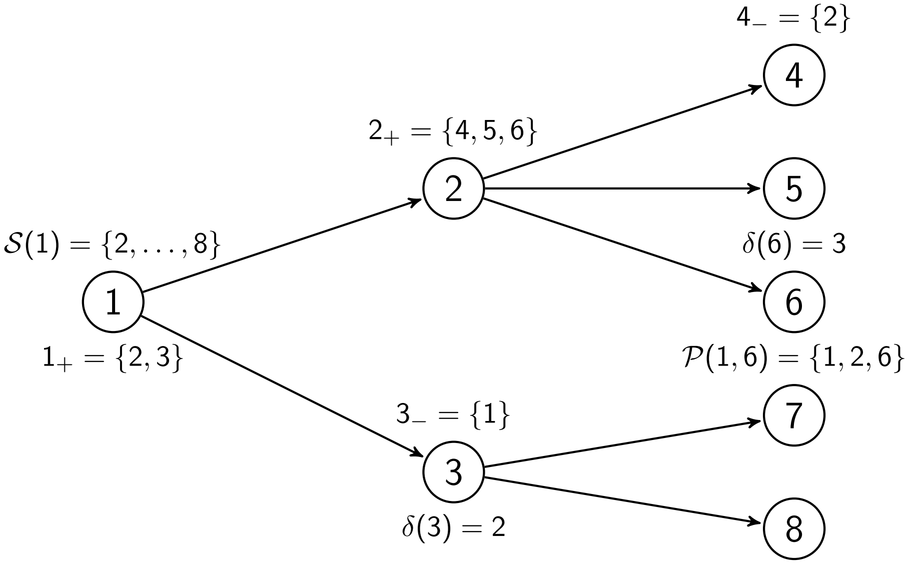
$$\min_{x \geq 0, Ax = b} c^T x + \mathcal{Q}(x)$$



Scenario tree

- Tree (directed graph) in which each node represents a state of the world
- Root node has no predecessors
- Depth of node (number of arcs on path back to root) could represent time
- Can represent using `n` (nodes in tree) and `succ` (pairs (i, j) such that j is a successor (child) of i), $\texttt{prob} = \phi_i$ for probabilites (on arcs) and data for the random variable realization
- Can use notation n_- to represent the predecessor (parent) of n and n_+ to represent the set of children of n
- $\mathcal{P}(n, m)$ is the set of nodes on path from n to m
- $\mathcal{S}(m)$ is the set of decendants of m

Scenario tree: $\mathcal{N} = \{1, \ldots, 8\}$, $\mathcal{L} = \{4, \ldots, 8\}$, $T = 3$



Example: Clear Lake Model

- Water levels $l(n)$ in dam for each node n
- Determine what to release normally $r(n)$, what then floods $f(n)$ and what to import $z(n)$
- Minimize cost of flooding and import
- Change in reservoir level between n_- and n is $\delta(n)$

$$\begin{aligned} &\min cost = c(f, z) \\ &\text{ s.t. } \; l(n) = l(n_-) + \delta(n) + z(n) - r(n) - f(n) \end{aligned}$$

- Random variables are δ, realized at node $n > 1$.
- Variables l,r,f,z at each node n.
- Balance constraint at each node n.

Example of a multi-stage stochastic program.

Multistage definition

$$
\begin{aligned}
\min_{x \in X} \; & c_1(x_1) + \mathbb{E}_1(Q_1(x_{\mathcal{S}(1)}, \omega_1)) \\
& = c_1(x_1) + \mathbb{E}_1([c_n(x_n) + \mathbb{E}_n(Q_n(x_{\mathcal{S}(n)}, \omega_n))]_{n \in 1_+}) \\
& \vdots \\
& = c_1(x_1) + \mathbb{E}_1\Bigg(\Bigg[c_n(x_n) + \mathbb{E}_n(\cdots \\
& \qquad\qquad + \cdots + c_l(x_l)]_{l \in \mathcal{L}} \cdots)\Bigg]_{n \in 1_+}\Bigg)
\end{aligned}
$$

Hard to write down: use recursion within ReSHOP

At leaf nodes l future cost is immediate cost. At parents m of leaf nodes:

$$\mathbb{E}_m\Big(\big[c_l(x_l)\big]_{l\in m_+}\Big) = \sum_{l\in m_+} \phi_l \cdot c_l(x_l)$$

Moving back another level to node n,

$$\begin{aligned}
\mathbb{E}_n\Big(\big[c_m(x_m)\big]_{m\in\mathcal{S}(n)}\Big) &= \sum_{m\in n_+} \phi_m \cdot \left(c_m(x_m) + \mathbb{E}_m\Big(\big[c_l(x_l)\big]_{l\in m_+}\Big)\right) \\
&= \sum_{m\in n_+} \phi_m \cdot \left(c_m(x_m) + \sum_{l\in m_+} \phi_l \cdot c_l(x_l)\right) \\
&= \sum_{m\in\mathcal{S}(n)} \Big(\prod_{l\in\mathcal{P}(n,m)\setminus\{n\}} \phi_l\Big) \cdot c_m(x_m)
\end{aligned}$$

The total cost the player wishes to minimize is:

$$\begin{aligned}
c_1(x_1) + \mathbb{E}_1\Big(\big[c_n(x_n)\big]_{n\in\mathcal{S}(1)}\Big) &= c_1(x_1) + \sum_{n\in\mathcal{S}(1)} \Big(\prod_{l\in\mathcal{P}(1,n)} \phi_l\Big) \cdot c_n(x_n) \\
&= c_1(x_1) + \sum_{n\in\mathcal{S}(1)} \pi_n \cdot c_n(x_n)
\end{aligned}$$

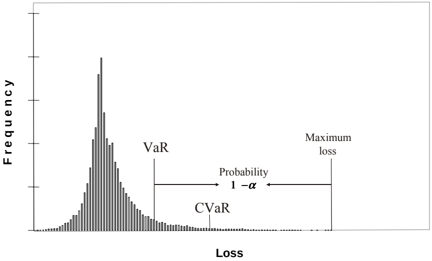
Risk Measures

$\overline{CVaR}_\alpha$: mean of upper tail beyond α-quantile (e.g. $\alpha = 0.95$)

- Classical: utility/disutility $u(\cdot)$:

$$\min_{x \in X} f(x) = \mathbb{E}[u(F(x,\xi))]$$

- Modern approach to modeling risk aversion uses concept of risk measures



- mean-risk, semi-deviations, mean deviations from quantiles, VaR, CVaR
- Römisch, Schultz, Rockafellar, Urasyev (in Math Prog literature)
- Much more in mathematical economics and finance literature
- Optimization approaches still valid, different objectives

Multistage risk-averse definition

$$
\begin{aligned}
\min_{x \in X} \; & c_1(x_1) + \rho_1(Q_1(x_{\mathcal{S}(1)}, \omega_1)) \\
& = c_1(x_1) + \rho_1([c_n(x_n) + \rho_n(Q_n(x_{\mathcal{S}(n)}, \omega_n))]_{n \in 1_+}) \\
& \vdots \\
& = c_1(x_1) + \rho_1\Big(\Big[c_n(x_n) + \rho_n(\cdots \\
& \qquad\qquad + \cdots + c_l(x_l)]_{l \in \mathcal{L}} \cdots)\Big]_{n \in 1_+}\Big)
\end{aligned}
$$

Hard to write down: use recursion within ReSHOP

CVaR as an example of coherent risk measure

- Problem transformation: theory states $\underline{CVaR}_\alpha(X)$ can be written as convex optimization using:

$$\underline{CVaR}_\alpha(X) = \max_{v \in \mathbb{R}} \left\{ v - \frac{1}{\alpha} \sum_{j=1}^{N} Prob_j * (v - X_j)_+ \right\}$$

  and this can then be put into a maximization problem or constraints like: $\underline{CVaR}_\alpha(X) \geq \gamma$

- Problem transformation: theory states $\overline{CVaR}_\alpha(X)$ can be written as convex optimization using:

$$\overline{CVaR}_\alpha(X) = \min_{v \in \mathbb{R}} \left\{ v + \frac{1}{1-\alpha} \sum_{j=1}^{N} Prob_j * (X_j - v)_+ \right\}$$

  and this can then be put into a minimization problem or constraints like: $\overline{CVaR}_\alpha(X) \leq \gamma$

Computation of SAA

- Can use CPLEX (or Gurobi) options to solve using barrier method, e,g,

```
advind 0
lpmethod 4
solutiontype 2
names no
threads 2
```

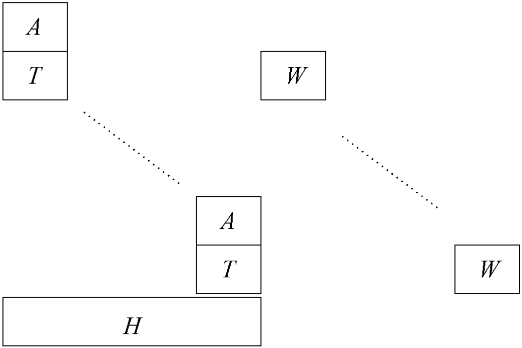
- Can generate samples - see `36sampling.ipynb` - to reduce size of problem

Summary and Other Extensions

- Sampling: SAA and out of sample testing (see newsvendor example)
- Probabilistic constraints
- Multi-stage problems: decisions at more than 2 stages
- Risk measures: techniques to value future outcomes differently
- Algorithms to exploit structure of problems:
  - Dynamic programming and approximations - rolling horizon, ADP, SDDP
  - Decomposition methods: Benders, L-shaped method, proximal algorithms
  - Large scale computational schemes: parallel implementations, etc

Key-idea: Non-anticipativity constraints

- Replace x with $x_1, x_2, \ldots, x_K$
- Non-anticipativity: $(x_1, x_2, \ldots, x_K) \in L$ (a subspace) - the H constraints



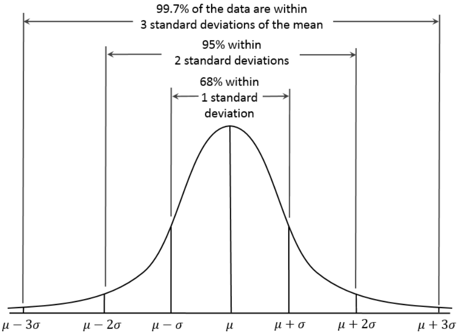
Computational methods exploit the separability of these constraints, essentially by dualization of the non-anticipativity constraints.

- Primal and dual decompositions (Lagrangian relaxation, progressive hedging , etc)
- L shaped method (Benders decomposition applied to det. equiv.)
- Trust region methods and/or regularized decomposition

Example: portfolio optimization

We must decide how to invest our money, and we can choose between $i = 1, 2, \ldots, N$ different assets.

- Each asset can be modeled as a random variable (RV) with an expected return μ_i and a standard deviation σ_i.



- Standard deviation is a measure of uncertainty.

Make Some Money in the Market

- R — A random variable representing portfolio return
- Z_i — The (random) return of stock i
- x_i — A decision variable. The amount invested in stock i

$$R = \sum_{i=1}^{N} x_i Z_i$$

- We can't optimize R (since it is random), but we can optimize $\mathbb{E}R$

Expectation is Linear

$$\begin{aligned}
\mathbb{E}R &= \mathbb{E}\left[\sum_{i=1}^{N} x_i Z_i\right] \\
&= \sum_{i=1}^{N} \mathbb{E}[x_i Z_i] \\
&= \sum_{i=1}^{n} x_i \mathbb{E} Z_i \stackrel{\text{def}}{=} \sum_{i=1}^{n} x_i \mu_i
\end{aligned}$$

Optimization Model

$$\max \sum_{i=1}^{N} \mu_i x_i$$

subject to

$$\begin{aligned} \sum_{i=1}^{N} x_i &= 1 \\ x_i &\geq 0 \end{aligned}$$

One Jillion \$ Question

- What is the optimal solution?

Risk

Which of these two yearly returns would you rather have?

Year	Portfolio #1	Portfolio #2
1	100	100
2	100	200
3	100	0
4	100	10
5	100	200
Avg	100	102

- Let's measure deviation from average
- Portfolio 1: 0
- Portfolio 2:

$$((102-100)^2+(102-200)^2+(102-0)^2+(102-10)^2+(102-200)^2)$$

- Variance is like average deviation from average

Example: portfolio optimization

If $Z_1 \sim (\mu_1, \sigma_1^2)$ and $Z_2 \sim (\mu_2, \sigma_2^2)$, what is $x_1 Z_1 + x_2 Z_2$?

Calculating the mean:

$$\begin{aligned} \mathbf{E}(x_1 Z_1 + x_2 Z_2) &= x_1 \mu_1 + x_2 \mu_2 \\ &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \end{aligned}$$

Calculating the variance:

$$\begin{aligned} \mathbf{var}(x_1 Z_1 + x_2 Z_2) &= \mathbf{E}\left(x_1(Z_1 - \mu_1) + x_2(Z_2 - \mu_2)\right)^2 \\ &= x_1^2\, \mathbf{var}(Z_1) + 2 x_1 x_2\, \mathbf{cov}(Z_1, Z_2) + x_2^2\, \mathbf{var}(Z_2) \\ &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{cov}(Z_1, Z_1) & \mathbf{cov}(Z_1, Z_2) \\ \mathbf{cov}(Z_2, Z_1) & \mathbf{cov}(Z_2, Z_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned}$$

More Ugly Math

$$
\begin{aligned}
\mathrm{Var}(R) &= \mathrm{Var}\left(\sum_{i=1}^{N} x_i Z_i\right) \\
&= \sum_{i=1}^{N} \mathrm{Var}(Z_i x_i) + 2\sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{Cov}(Z_i x_i, Z_j x_j) \\
&= \sum_{i=1}^{N} x_i^2 \mathrm{Var}(Z_i) + 2\sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j \mathrm{Cov}(Z_i, Z_j) \\
&= x^T \Sigma x
\end{aligned}
$$

- Where Σ is the variance-covariance matrix

Example: portfolio optimization: `27portfolio.ipynb`

Suppose we buy x_i of asset Z_i . We want:

- A high total return. Maximize $x^T \mu$.
- Low variance (risk). Minimize $x^T \Sigma x$.

Pose the optimization problem as a tradeoff:

$$\max \sum_{i=1}^{N} \mu_i x_i - \lambda x^T \Sigma x$$

$$\begin{aligned} \text{subject to } x_1 + \cdots + x_{225} &= 1 \\ x_i &\geq 0 \end{aligned}$$

Fun fact: This is the basic idea behind "Modern portfolio theory". Introduced by economist Harry Markowitz in 1952, for which he was later awarded the Nobel Prize.

Tradeoffs: objective or constraint

$$\max \sum_{i=1}^{N} \mu_i x_i - \lambda x^T \Sigma x$$

subject to

$$\begin{aligned} \sum_{i=1}^{N} x_i &= 1 \\ x_i &\geq 0 \end{aligned}$$

$$\min x^T \Sigma x$$

subject to

$$\begin{aligned} \sum_{i=1}^{N} \mu_i x_i &\geq K \\ \sum_{i=1}^{N} x_i &= 1 \\ x_i &\geq 0 \end{aligned}$$

$$x^T \Sigma x = \sum_{i=1}^{N} \sigma_i^2 x_i^2 + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j \mathrm{Cov}(Z_i, Z_j)$$

Example: Portfolio with CVaR

- Determine portfolio weights x_j for each of a collection of assets
- Asset returns V are random, but jointly distributed
- Portfolio return $R(x, V)$
- Optimize a "risk" measure

$$\begin{array}{rl} \max & 0.2 * \mathbb{E}(R) + 0.8 * \underline{CVaR}_{\alpha}(R) \\ \text{s.t.} & R = \sum_j V_j * x_j \\ & \sum_j x_j = 1,\ x \geq 0 \end{array}$$

- Jointly distributed random variables V, realized at stage 2
- Variables: portfolio weights x in stage 1, returns R in stage 2
- Coherent risk measures $\mathbb{E}$ and $\underline{CVaR}$