# Stochastic Programming

An introduction

Giovanni Pantuso

Department of Mathematical Sciences
University of Copenhagen
Copenhagen, Denmark

# Table of Contents

# After this lecture you will know

Formulate general stochastic programs

# After this lecture you will know

Formulate general stochastic programs

What makes them difficult

# After this lecture you will know

Formulate general stochastic programs

What makes them difficult

How to solve them

# After this lecture you will know

Formulate general stochastic programs with recourse (two and multi-stage)

What makes them difficult

How to solve them

# After this lecture you will know

Formulate general stochastic programs with recourse (two and multi-stage) for a risk neutral decision maker

What makes them difficult

How to solve them

# After this lecture you will know

Formulate general stochastic programs with recourse (two and multi-stage) for a risk neutral decision maker

What makes them difficult and how to address the difficulty

How to solve them

# After this lecture you will know

Formulate general stochastic programs with recourse (two and multi-stage) for a risk neutral decision maker

What makes them difficult and how to address the difficulty

How to solve them *assuming a small number of decision stages...*

# The fence...

- Risk-aversion

# The fence...

- Risk-aversion
- Chance constraints

# The fence...

- Risk-aversion
- Chance constraints
- Stochastic dominance

# The fence...

- Risk-aversion
- Chance constraints
- Stochastic dominance
- Large number of stages

# The fence...

- Risk-aversion
- Chance constraints
- Stochastic dominance
- Large number of stages
- Robust

# The fence...

- Risk-aversion
- Chance constraints
- Stochastic dominance
- Large number of stages
- Robust
- Distributionally robust

# The fence...

- Risk-aversion
- Chance constraints
- Stochastic dominance
- Large number of stages
- Robust
- Distributionally robust
- Non-linear

# The fence...

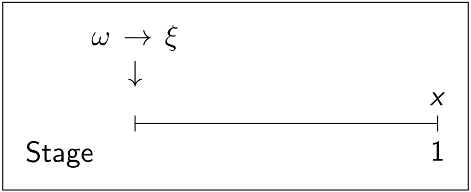
- Risk-aversion
- Chance constraints
- Stochastic dominance
- Large number of stages
- Robust
- Distributionally robust
- Non-linear
- Multi-objective

# The fence...

- Risk-aversion
- Chance constraints
- Stochastic dominance
- Large number of stages
- Robust
- Distributionally robust
- Non-linear
- Multi-objective
- Endogenous uncertainty

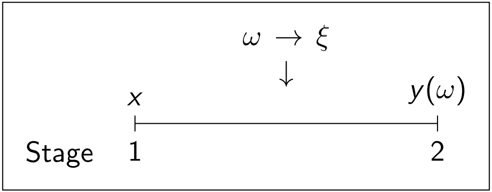
# Table of Contents

# Decision stages

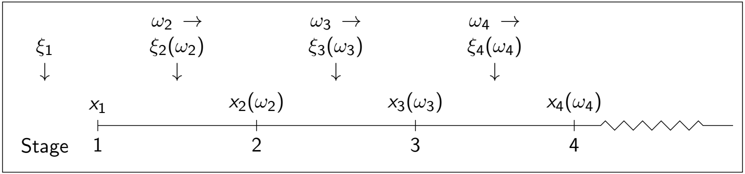


# Decision stages

# Decision stages



# Decision stages



# Two-Stage Stochastic Programs with Recourse

The decision maker:

1. Makes $x \in \mathcal{X} \subseteq \mathbb{R}^{n_1}$ (first stage)

# Two-Stage Stochastic Programs with Recourse

The decision maker:

1. Makes $x \in \mathcal{X} \subseteq \mathbb{R}^{n_1}$ (first stage)
2. Waits for the outcome $\omega \in \Omega$ of some random experiment. $\omega$ determines $\boldsymbol{\xi}(\omega)$ (our random data)

# Two-Stage Stochastic Programs with Recourse

The decision maker:

1. Makes $x \in \mathcal{X} \subseteq \mathbb{R}^{n_1}$ (first stage)
2. Waits for the outcome $\omega \in \Omega$ of some random experiment. $\omega$ determines $\boldsymbol{\xi}(\omega)$ (our random data)
3. Makes $y(\omega) \in \mathcal{Y} \subseteq \mathbb{R}^{n_2}$, given $\xi$ and $x$ (second stage)

# Two-Stage Stochastic Programs with Recourse

$$\min z = c^\top x + \mathbb{E}_{\boldsymbol{\xi}}[\min \boldsymbol{q}(\omega)^T y(\omega)]$$

# Two-Stage Stochastic Programs with Recourse

$$\min z = c^\top x + \mathbb{E}_{\boldsymbol{\xi}}[\min \boldsymbol{q}(\omega)^T y(\omega)]$$
$$\text{s.t. } Ax = b,$$

# Two-Stage Stochastic Programs with Recourse

$$
\begin{aligned}
\min z = & c^\top x + \mathbb{E}_{\boldsymbol{\xi}}[\min \boldsymbol{q}(\omega)^T y(\omega)] \\
\text{s.t. } & Ax = b, \\
& \boldsymbol{T}(\omega)x + \boldsymbol{W}(\omega)y(\omega) = \boldsymbol{h}(\omega), \\
& x \in \mathcal{X}, y(\omega) \in \mathcal{Y}
\end{aligned}
$$

# Two-Stage Stochastic Programs with Recourse

Parameters $c \in \mathbb{R}^{n_1}$, $b \in \mathbb{R}^{m_1}$, and $A \in \mathbb{R}^{n_1 \times m_1}$ are known.

# Two-Stage Stochastic Programs with Recourse

Parameters $c \in \mathbb{R}^{n_1}$, $b \in \mathbb{R}^{m_1}$, and $A \in \mathbb{R}^{n_1 \times m_1}$ are known.

Parameters $\boldsymbol{q}(\omega) \in \mathbb{R}^{n_2}$, $\boldsymbol{h}(\omega) \in \mathbb{R}^{m_2}$, $\boldsymbol{W}(\omega) \in \mathbb{R}^{m_2 \times n_2}$ and $\boldsymbol{T}(\omega) \in \mathbb{R}^{m_2 \times n_1}$ are uncertain.

# Two-Stage Stochastic Programs with Recourse

Parameters $c \in \mathbb{R}^{n_1}$, $b \in \mathbb{R}^{m_1}$, and $A \in \mathbb{R}^{n_1 \times m_1}$ are known.

Parameters $\boldsymbol{q}(\omega) \in \mathbb{R}^{n_2}$, $\boldsymbol{h}(\omega) \in \mathbb{R}^{m_2}$, $\boldsymbol{W}(\omega) \in \mathbb{R}^{m_2 \times n_2}$ and $\boldsymbol{T}(\omega) \in \mathbb{R}^{m_2 \times n_1}$ are uncertain.

$\boldsymbol{\xi}(\omega) = \left(\boldsymbol{q}(\omega)^\top, \boldsymbol{h}(\omega)^\top,\ \boldsymbol{W}^{\mathbf{1}}(\omega), \ldots, \boldsymbol{W}(\omega)^{m_2}, \boldsymbol{T}(\omega)^1, \ldots, \boldsymbol{T}(\omega)^{m2}\right)$.
$\xi$ is a realization.

# Two-Stage Stochastic Programs with Recourse

Parameters $c \in \mathbb{R}^{n_1}$, $b \in \mathbb{R}^{m_1}$, and $A \in \mathbb{R}^{n_1 \times m_1}$ are known.

Parameters $\boldsymbol{q}(\omega) \in \mathbb{R}^{n_2}$, $\boldsymbol{h}(\omega) \in \mathbb{R}^{m_2}$, $\boldsymbol{W}(\omega) \in \mathbb{R}^{m_2 \times n_2}$ and $\boldsymbol{T}(\omega) \in \mathbb{R}^{m_2 \times n_1}$ are uncertain.

$\boldsymbol{\xi}(\omega) = \left(\boldsymbol{q}(\omega)^\top, \boldsymbol{h}(\omega)^\top,\ \boldsymbol{W}^{\mathbf{1}}(\omega), \ldots, \boldsymbol{W}(\omega)^{m_2}, \boldsymbol{T}(\omega)^1, \ldots, \boldsymbol{T}(\omega)^{m2}\right)$.
$\xi$ is a realization.

$$(\Omega, \mathcal{F}, \mathbb{P})$$

We only need a specification of $\boldsymbol{\xi}$ (e.g., probability density/mass function and support $\Xi$).

# Two-Stage Stochastic Programs with Recourse

$$\begin{aligned} \min z &= c^\top x + Q(x) \\ \text{s.t.} Ax &= b \\ x &\in \mathcal{X} \end{aligned}$$

# Two-Stage Stochastic Programs with Recourse

$$\begin{aligned} \min z &= c^\top x + Q(x) \\ \text{s.t.} Ax &= b \\ x &\in \mathcal{X} \end{aligned}$$

where

$$Q(x) = \mathbb{E}_{\boldsymbol{\xi}}[Q(x, \xi)]$$

# Two-Stage Stochastic Programs with Recourse

$$\min z = c^\top x + \mathcal{Q}(x)$$
$$\text{s.t.} Ax = b$$
$$x \in \mathcal{X}$$

where

$$\mathcal{Q}(x) = \mathbb{E}_{\boldsymbol{\xi}}[Q(x, \xi)]$$

$$Q(x, \xi) = \min_{y} \{ q(\omega)^\top y | W(\omega) y = h(\omega) - T(\omega) x, y \in \mathcal{Y} \}.$$

# Two-Stage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Consider $\Xi = \{\xi_1, \ldots, \xi_S\}$ with probabilities $\pi_s$, $s = 1 \ldots, S$.

# Two-Stage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Consider $\Xi = \{\xi_1, \ldots, \xi_S\}$ with probabilities $\pi_s$, $s = 1 \ldots, S$.

$\xi_s \implies q_s^\top, T_s, W_s, h_s.$

# Two-Stage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Consider $\Xi = \{\xi_1, \ldots, \xi_S\}$ with probabilities $\pi_s$, $s = 1 \ldots, S$.

$\xi_s \implies q_s^\top, T_s, W_s, h_s$.

$y(\omega)$ becomes $y_1, \ldots, y_S$.

# Two-Stage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

$$
\begin{aligned}
\min z = & c^\top x + \sum_{s=1}^{S} \pi_s q_s^T y_s \\
\text{s.t. } & Ax = b, \\
& T_s x + W_s y_s = h_s, && s = 1, \dots, S \\
& x \in \mathcal{X}, \\
& y_s \in \mathcal{Y}, && s = 1, \dots, S.
\end{aligned}
$$

# Multistage Stochastic Programs with Recourse

The decision maker:

1. Makes decisions $x_1$ based on $\xi_1$

# Multistage Stochastic Programs with Recourse

The decision maker:

1. Makes decisions $x_1$ based on $\xi_1$
2. Waits for outcome $\boldsymbol{\xi}_2(\omega_2)$

# Multistage Stochastic Programs with Recourse

The decision maker:

1. Makes decisions $x_1$ based on $\xi_1$
2. Waits for outcome $\boldsymbol{\xi}_2(\omega_2)$
3. Makes decisions $x_2(\omega_2)$ based on $x_1$ and realization $\xi_1, \xi_2$

# Multistage Stochastic Programs with Recourse

The decision maker:

1. Makes decisions $x_1$ based on $\xi_1$
2. Waits for outcome $\boldsymbol{\xi}_2(\omega_2)$
3. Makes decisions $x_2(\omega_2)$ based on $x_1$ and realization $\xi_1, \xi_2$
4. Waits for $\boldsymbol{\xi}_3(\omega_3)$

# Multistage Stochastic Programs with Recourse

The decision maker:

1. Makes decisions $x_1$ based on $\xi_1$
2. Waits for outcome $\boldsymbol{\xi}_2(\omega_2)$
3. Makes decisions $x_2(\omega_2)$ based on $x_1$ and realization $\xi_1, \xi_2$
4. Waits for $\boldsymbol{\xi}_3(\omega_3)$
5. Makes decisions $x_3(\omega_3)$ based on $x_1, x_2$ and realization $\xi_1, \xi_2, \xi_3$
6. ....
7. Waits for $\boldsymbol{\xi}_T(\omega_T)$
8. Makes decisions $x_T(\omega_T)$

# Multistage Stochastic Programs with Recourse

$$\min z = c_1^\top \; x_1$$

# Multistage Stochastic Programs with Recourse

$$\min z = c_1^\top x_1 + \mathbb{E}_{\boldsymbol{\xi}_2|\xi_{[1]}}\left[ \min \boldsymbol{c}_2(\omega_2)^\top x_2(\omega_2) + \cdots \right]$$

# Multistage Stochastic Programs with Recourse

$$\min z = c_1^\top \; x_1 + \mathbb{E}_{\boldsymbol{\xi}_2|\xi_{[1]}}\left[ \min \boldsymbol{c}_2(\omega_2)^\top x_2(\omega_2) + \mathbb{E}_{\boldsymbol{\xi}_3|\xi_{[2]}}\left[ \boldsymbol{c}_3(\omega_3) x_3(\omega_3) + \cdots \right]\right]$$

# Multistage Stochastic Programs with Recourse

$$\min z = c_1^\top \; x_1 + \mathbb{E}_{\boldsymbol{\xi}_2|\xi_{[1]}}\Bigg[\min \boldsymbol{c}_2(\omega_2)^\top x_2(\omega_2) + \mathbb{E}_{\boldsymbol{\xi}_3|\xi_{[2]}}\bigg[\cdots$$

$$+ \mathbb{E}_{\boldsymbol{\xi}_{T-1}|\xi_{[T-2]}}\bigg[\min \boldsymbol{c}_{T-1}(\omega_{T-1})^\top x_{T-1}(\omega_{T_1}) + \mathbb{E}_{\boldsymbol{\xi}_T|\xi_{[T-1]}}\big[\min \boldsymbol{c}_T(\omega_T)^\top x_T(\omega_T)\big]\bigg]\cdots\bigg]\Bigg]$$

# Multistage Stochastic Programs with Recourse

$$\min z = c_1^\top \; x_1 + \mathbb{E}_{\boldsymbol{\xi}_2|\xi_{[1]}}\Bigg[\min \boldsymbol{c}_2(\omega_2)^\top x_2(\omega_2) + \mathbb{E}_{\boldsymbol{\xi}_3|\xi_{[2]}}\Big[\cdots$$

$$+ \mathbb{E}_{\boldsymbol{\xi}_{T-1}|\xi_{[T-2]}}\Big[\min \boldsymbol{c}_{T-1}(\omega_{T-1})^\top x_{T-1}(\omega_{T_1}) + \mathbb{E}_{\boldsymbol{\xi}_T|\xi_{[T-1]}}\big[\min \boldsymbol{c}_T(\omega_T)^\top x_T(\omega_T)\big]\Big]\cdots\Big]\Bigg]$$

$$\text{s.t. } W_1 x_1 = h_1,$$

# Multistage Stochastic Programs with Recourse

$$\min z = c_1^\top x_1 + \mathbb{E}_{\boldsymbol{\xi}_2|\xi_{[1]}}\left[\min \boldsymbol{c}_2(\omega_2)^\top x_2(\omega_2) + \mathbb{E}_{\boldsymbol{\xi}_3|\xi_{[2]}}\Bigg[\cdots\right.$$

$$\left. + \mathbb{E}_{\boldsymbol{\xi}_{T-1}|\xi_{[T-2]}}\Bigg[\min \boldsymbol{c}_{T-1}(\omega_{T-1})^\top x_{T-1}(\omega_{T_1}) + \mathbb{E}_{\boldsymbol{\xi}_T|\xi_{[T-1]}}\big[\min \boldsymbol{c}_T(\omega_T)^\top x_T(\omega_T)\big]\cdots\Bigg]\right]$$

$$\text{s.t. } W_1 x_1 = h_1,$$

$$\boldsymbol{T}_1(\omega_2)x_1 + \boldsymbol{W}_2(\omega_2)x_2(\omega_2) = \boldsymbol{h}_2(\omega_2),$$

# Multistage Stochastic Programs with Recourse

$$\min z = c_1^\top x_1 + \mathbb{E}_{\boldsymbol{\xi}_2|\xi_{[1]}}\Bigg[\min \boldsymbol{c}_2(\omega_2)^\top x_2(\omega_2) + \mathbb{E}_{\boldsymbol{\xi}_3|\xi_{[2]}}\Big[\cdots$$

$$+ \mathbb{E}_{\boldsymbol{\xi}_{T-1}|\xi_{[T-2]}}\Big[\min \boldsymbol{c}_{T-1}(\omega_{T-1})^\top x_{T-1}(\omega_{T_1}) + \mathbb{E}_{\boldsymbol{\xi}_T|\xi_{[T-1]}}\big[\min \boldsymbol{c}_T(\omega_T)^\top x_T(\omega_T)\big]\Big]\cdots\Big]\Bigg]$$

$$\text{s.t. } W_1 x_1 = h_1,$$

$$\boldsymbol{T}_1(\omega_2)x_1 + \boldsymbol{W}_2(\omega_2)x_2(\omega_2) = \boldsymbol{h}_2(\omega_2),$$

$$\vdots$$

$$\boldsymbol{T}_{T-1}(\omega_T)x_{T-1}(\omega_{T-1}) + \boldsymbol{W}_T(\omega_T)x_T(\omega_T) = \boldsymbol{h}_T(\omega_T),$$

$$x_1 \in \mathcal{X}_1, x_t(\omega_t) \in \mathcal{X}_t, t = 2, \ldots, T.$$

# Multistage Stochastic Programs with Recourse

$$
\begin{aligned}
Q_T(x_{T-1}, \xi_{[T]}) = \min\ & c_T x_T \\
\text{s.t. } & W_T x_T = h_T - T_{T-1} x_{T-1} \\
& x_T \in \mathcal{X}_T
\end{aligned}
$$

# Multistage Stochastic Programs with Recourse

$$\begin{aligned} Q_T(x_{T-1},\xi_{[T]}) = \min\ & c_T x_T \\ \text{s.t. } & W_T x_T = h_T - T_{T-1} x_{T-1} \\ & x_T \in \mathcal{X}_T \end{aligned}$$

for $t = 2, \ldots, T-1$

$$\begin{aligned} Q_t(x_{t-1}, \xi_t) = \min\ & c_t x_t + \mathcal{Q}_{t+1}(x_t) \\ \text{s.t. } & W_t x_t = h_t - T_{t-1} x_{t-1} \\ & x_t \in \mathcal{X}_t \end{aligned}$$

where $\mathcal{Q}_{t+1}(x_t) = \mathbb{E}_{\boldsymbol{\xi}_{t+1}|\xi_{[t]}}[Q_{t+1}(x_t, \xi_{t+1})]$.

# Multistage Stochastic Programs with Recourse

Finally

$$
\begin{aligned}
\min z &= c_1 x_1 + Q_2(x_1) \\
\text{s.t. } W_1 x_1 &= h_1, \\
x_1 &\in \mathcal{X}_1.
\end{aligned}
$$

# Multistage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Special structure called a *Scenario Tree*.

# Multistage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Special structure called a *Scenario Tree*.

Example: Assume a three-stage random process $\boldsymbol{\xi} = (\xi_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$.
$\xi_1$ is known (assume 10) and at every $t$

- the value is doubled with a probability of 0.5
- the value is halved with a probability of 0.5

# Multistage Stochastic Programs with Recourse with Discrete $\xi$

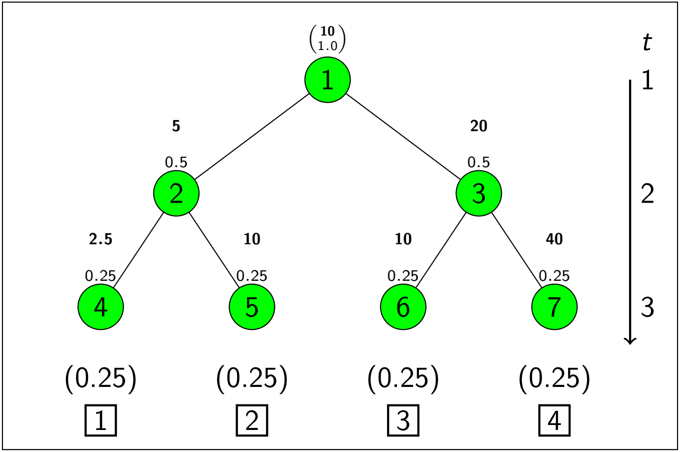



Figure 1:

# Multistage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Example: Assume a three-stage random process $\boldsymbol{\xi} = (\xi_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$.
$\xi_1$ is known (assume $(10, 50)$) and at every $t$

- the value is doubled with a probability of 0.5
- the value is halved with a probability of 0.5

# Multistage Stochastic Programs with Recourse with Discrete $\xi$

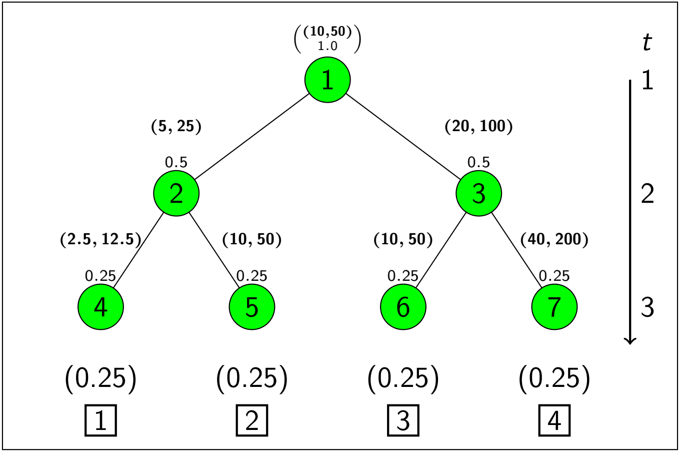



Figure 2:

# Multistage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Node formulation

$$
\begin{aligned}
\min z = \sum_{n \in \mathcal{N}} \pi_n c_n^\top x_n \\
\text{s.t.} \quad & W_1 x_1 = h_1, \\
& W_n x_n + T_{a(n)} x_{a(n)} = h_n && \forall n \in \mathcal{N} \setminus \{1\}, \\
& x_n \in \mathcal{X}_{t(n)} && \forall n \in \mathcal{N}.
\end{aligned}
$$

# Multistage Stochastic Programs with Recourse with Discrete $\xi$

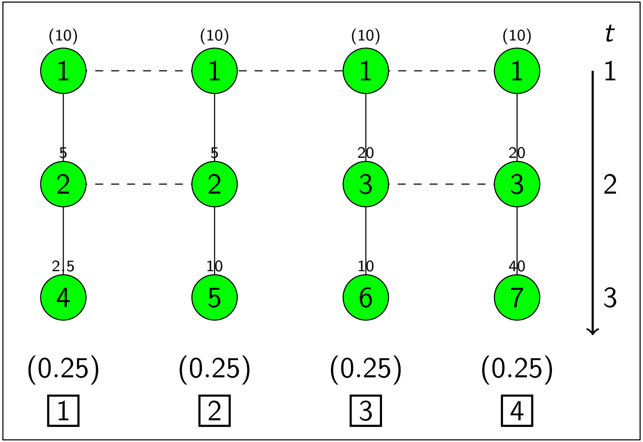



Figure 3:

# Multistage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Scenario formulation

$$\begin{aligned}
\min \quad & z = \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} \pi_s c_{ts}^\top x_{ts} & \\
\boldsymbol{s}.t. \quad & W_{1s} x_{1s} = h_{1s}, & s \in \mathcal{S}, \\
& W_{ts} x_{ts} + T_{t-1,s} x_{t-1,s} = h_{ts}, & s \in \mathcal{S}, t \in \mathcal{T} \setminus \{1\}, \\
& \texttt{non-anticipativity constraints} & \\
& x_{ts} \in \mathcal{X}_t & s \in \mathcal{S}, t \in \mathcal{T}.
\end{aligned}$$

# Multistage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Non-anticipativity constraints

$$x_{ts} - x_{ts'} = 0 \;\; \forall t \in \mathcal{T}, s, s' \in \mathcal{S} : \xi_{1s}, \ldots, \xi_{ts} = \xi_{1s'}, \ldots, \xi_{ts'}$$

# Multistage Stochastic Programs with Recourse with Discrete $\boldsymbol{\xi}$

Scenario formulation or Node formulation?

# Table of Contents

# A closer look at $\xi$

Where do we get $\xi$?

# A closer look at $\xi$

Where do we get $\xi$?

- Number of failures $\approx$ Weibull
- Wind speed $\approx$ Weibull,Rayleigh
- Forecast error (linear regression) $\approx$ Normal
- Hospitalization in certain epidemics $\approx$ LogNormal
- Repair times $\approx$ LogNormal
- Choice model $\approx$ Gumbel, Normal, EV Type I
- Waiting times $\approx$ Beta
- ...

# A closer look at $\xi$

Some randomness is discrete

# A closer look at $\xi$

Some randomness is discrete

- Number of occurrences $\approx$ Poisson
- Number of trials before success $\approx$ Geometric
- Number of successes $\approx$ HyperGeometric
- ...

# A closer look at $\xi$

High dimensions and mixes are problematic

# A closer look at the constraints

If $\xi$ is continuous...

# A closer look at the constraints

If $\xi$ is continuous...

Constraints must hold a.s. ...

# A closer look at the constraints

If $\xi$ is continuous...

Constraints must hold a.s. ...

Possibly $\infty$ constraints

# A closer look at $Q(x)$

The recourse function ...

$$Q(x) = \mathbb{E}_{\boldsymbol{\xi}}\left[Q(x,\xi)\right] = \int_{\Omega} Q(x,\xi(\omega))\mathbb{P}(d\omega)$$

Why is this difficult?

# A closer look at $Q(x)$

The recourse function ...

$$Q(x) = \mathbb{E}_{\boldsymbol{\xi}}\big[Q(x,\xi)\big] = \int_{\Omega} Q(x,\xi(\omega))\mathbb{P}(d\omega)$$

Why is this difficult?

Ingredient 1: a closed form expression $Q(x,\xi)$

# A closer look at $Q(x)$

The recourse function ...

$$Q(x) = \mathbb{E}_{\boldsymbol{\xi}}\big[Q(x,\xi)\big] = \int_{\Omega} Q(x,\xi(\omega))\mathbb{P}(d\omega)$$

Why is this difficult?

Ingredient 1: a closed form expression $Q(x,\xi)$

Ingredient 2: an antiderivative

# A closer look at $Q(x)$

The recourse function ...

$$Q(x) = \mathbb{E}_{\boldsymbol{\xi}}\left[Q(x,\xi)\right] = \int_{\Omega} Q(x,\xi(\omega))\mathbb{P}(d\omega)$$

Why is this difficult?

Ingredient 1: a closed form expression $Q(x,\xi)$

Ingredient 2: an antiderivative

Observe: $Q(x) = \int\int\cdots\int Q(x,\xi)\mathbb{D}(\xi)d\xi_1 d\xi_2\cdots d\xi_N$

# A closer look at $Q(x)$

The recourse function ...

$$Q(x) = \mathbb{E}_{\boldsymbol{\xi}}\left[Q(x,\xi)\right] = \int_{\Omega} Q(x,\xi(\omega))\mathbb{P}(d\omega)$$

Idea! Numerical integration!

# A closer look at $Q(x)$

In one dimension (i.e., $N = 1$): Riemann sums, Trapezoidal rule, Simpson's rule

Ex. Riemann Sums

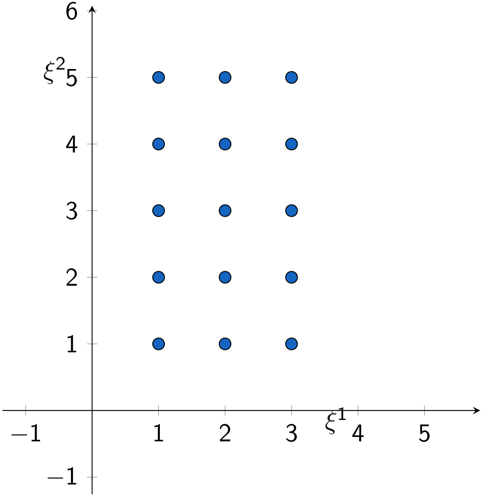
$$\int_a^b f(x)dx$$

- Partition $[a, b]$ using $K$ points $x_0 = a$, $x_1, \ldots, x_K = b$ equally spaced $\Delta x$
- $\int_a^b f(x)dx \approx \sum_k f(x_k)\Delta x$
- As $K$ increases we improve the approximation.

# A closer look at $Q(x)$

In multiple dimensions: *Quadrature* methods.

Same principle, harder partition



# A closer look at $Q(x)$

Numerical integration: it is already an approximation

# A closer look at $Q(x)$

Numerical integration: it is already an approximation

all this work for one $x$...

# A closer look at $Q(x)$

Numerical integration: it is already an approximation

all this work for one $x$...

we still have to solve the stochastic program..

# A closer look at $Q(x)$

When $\boldsymbol{\xi}$ is discrete...

$$Q(x) = \sum_{s=1}^{S} \pi_s Q(x, \xi_s)$$

# A closer look at $Q(x)$

When $\boldsymbol{\xi}$ is discrete...

$$Q(x) = \sum_{s=1}^{S} \pi_s Q(x, \xi_s)$$

Finitely many linear constraints

# A closer look at $Q(x)$

When $\boldsymbol{\xi}$ is discrete...

$$Q(x) = \sum_{s=1}^{S} \pi_s Q(x, \xi_s)$$

Finitely many linear constraints

When $\boldsymbol{\xi}$ is discrete ... but large ...

# Approximations

Continuous/discrete but large $\rightarrow$ discrete and small

# Approximations

Continuous/discrete but large $\rightarrow$ discrete and small

Three categories of methods (loosely speaking)

- Monte Carlo
- Probability Metrics
- Property Matching

# Monte Carlo in a nutshell

Make $K$ identical copies $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_K$ of $\boldsymbol{\xi}$

# Monte Carlo in a nutshell

Make $K$ identical copies $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_K$ of $\boldsymbol{\xi}$

From each take, independently, a realization $\xi_k$.

# Monte Carlo in a nutshell

Make $K$ identical copies $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_K$ of $\boldsymbol{\xi}$

From each take, independently, a realization $\xi_k$.

Write the SAA

$$
\begin{aligned}
\min z^K = & c^\top x + \sum_{k=1}^{K} \frac{1}{K} q_k^T y_k \\
\text{s.t. } & Ax = b, \\
& T_k x + W_k y_k = h_k, \qquad k = 1, \ldots, K \\
& x \in \mathcal{X}, \\
& y_k \in \mathcal{Y}, \qquad k = 1, \ldots, K.
\end{aligned}
$$

# Monte Carlo in a nutshell

Advantages

# Monte Carlo in a nutshell

Advantages

$f^K(x)$ is an unbiased estimator (pointwise $x = \bar{x}$)

# Monte Carlo in a nutshell

Advantages

$f^K(x)$ is an unbiased estimator (pointwise $x = \bar{x}$)

$z^K \to z^*$ as $k \to \infty$ (exponentially fast!)

# Monte Carlo in a nutshell

Advantages

$f^K(x)$ is an unbiased estimator (pointwise $x = \bar{x}$)

$z^K \to z^*$ as $k \to \infty$ (exponentially fast!)

$\mathbb{E}[z^K]$ gives a statistical lower bound! (obs! $z^K$ is biased)

# Monte Carlo in a nutshell

Advantages

$f^K(x)$ is an unbiased estimator (pointwise $x = \bar{x}$)

$z^K \to z^*$ as $k \to \infty$ (exponentially fast!)

$\mathbb{E}[z^K]$ gives a statistical lower bound! (obs! $z^K$ is biased)

$\mathbb{E}\big[c^\top \bar{x} + \frac{1}{K}\sum_{k=1}^{K}[Q(\bar{x}, \xi_k)]\big]$ gives a statistical upper bound!

See, e.g., [Sha91, MMW99, Sha03].

# Monte Carlo in a nutshell

However:

$K = (\frac{n_1}{\epsilon^2})$ samples to have an $\epsilon$-approximation.

# Monte Carlo in a nutshell

However:

$K = (\frac{n_1}{\epsilon^2})$ samples to have an $\epsilon$-approximation.

Improvements exist: variance reduction techniques.

# Method based on Probability Metrics in a nutshell

Results from research on stability, see,e.g., [Dup90, Pfl01, RR02, Röm03].

$$|z(\mathbb{P}) - z(\mathbb{Q})| \leq Ld(\mathbb{P}, \mathbb{Q})$$

# Method based on Probability Metrics in a nutshell

- Start from large $N$ scenarios (e.g., sampled)

# Method based on Probability Metrics in a nutshell

- Start from large $N$ scenarios (e.g., sampled)
- Remove one scenario at a time to minimize the distanced between the new and old distribution

# Method based on Probability Metrics in a nutshell

- Start from large $N$ scenarios (e.g., sampled)
- Remove one scenario at a time to minimize the distanced between the new and old distribution
- Add the probability of the deleted scenarios to the closest scenarios (in the sense of the probability metric)

Scenario reduction/generation, see, e.g., [HR03, DGKR03].

# Property Matching in a nutshell

Idea: Replicate only the statistical properties that are important for the problem [HW01].

Create a small distribution that replicates only those properties.

# Property Matching in a nutshell

Not always necessary to increase the size of the distribution.

# Property Matching in a nutshell

Not always necessary to increase the size of the distribution.

Problem driven.

# Property Matching in a nutshell

Not always necessary to increase the size of the distribution.

Problem driven.

Observe: requires an NLP (heuristics exist [HKW03])

# Property Matching in a nutshell

Not always necessary to increase the size of the distribution.

Problem driven.

Observe: requires an NLP (heuristics exist [HKW03])

Which properties?

# Approximations

Getting a good solution vs Estimating its value

# Mathematical properties of discrete stochastic programs

$$\mathcal{K}_2(\xi) = \{x | \exists y \geq 0, \text{s.t.} W(\omega)y = h(\omega) - T(\omega)x\}$$

# Mathematical properties of discrete stochastic programs

$$\mathcal{K}_2(\xi) = \{x | \exists y \geq 0, \text{s.t.} W(\omega)y = h(\omega) - T(\omega)x\}$$

Convex and polyhedral!

# Mathematical properties of discrete stochastic programs

$$\mathcal{K}_2 = \bigcap_{\xi \in \Xi} \mathcal{K}_2(\xi)$$

# Mathematical properties of discrete stochastic programs

$$\mathcal{K}_2 = \bigcap_{\xi \in \Xi} \mathcal{K}_2(\xi)$$

Convex and polyhedral!

# Mathematical properties of discrete stochastic programs

Useful jargon

**Complete recourse**

$$\mathcal{K}_2 = \mathbb{R}^{n_1}$$

# Mathematical properties of discrete stochastic programs

Useful jargon

**Complete recourse**

$$\mathcal{K}_2 = \mathbb{R}^{n_1}$$

**Relatively complete recourse**

$$\mathcal{K}_2 \subseteq \mathcal{K}_1 = \{x | Ax = b, x \geq 0\}$$

.

# Mathematical properties of discrete stochastic programs

$Q(x, \xi)$ is:

a. piece-wise linear convex in $h$, $T$ and $x$,

b. piece-wise linear concave in $q$.

# Mathematical properties of discrete stochastic programs

$$Q(x) = \mathbb{E}_{\boldsymbol{\xi}} Q(x, \xi) = \sum_{s=1}^{S} \pi_s Q(x, \xi_s)$$

piece-wise linear convex in $x$.

# Table of Contents

# References I


[DGKR03] Jitka Dupačová, Nicole Gröwe-Kuska, and Werner Römisch. Scenario reduction in stochastic programming. *Mathematical programming*, 95(3):493–511, 2003.

[Dup90] Jitka Dupačová. Stability and sensitivity-analysis for stochastic programming. *Annals of operations research*, 27(1):115–142, 1990.

[HKW03] Kjetil Høyland, Michal Kaut, and Stein W Wallace. A heuristic for moment-matching scenario generation. *Computational optimization and applications*, 24(2):169–185, 2003.

[HR03] Holger Heitsch and Werner Römisch. Scenario reduction algorithms in stochastic programming. *Computational optimization and applications*, 24(2):187–206, 2003.

# References II

[HW01] Kjetil Høyland and Stein W Wallace. Generating scenario trees for multistage decision problems. *Management science*, 47(2):295–307, 2001.

[MMW99] Wai-Kei Mak, David P Morton, and R Kevin Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations research letters*, 24(1-2):47–56, 1999.

[Pfl01] G Ch Pflug. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical programming*, 89(2):251–271, 2001.

[Röm03] Werner Römisch. Stability of stochastic programming problems. *Handbooks in operations research and management science*, 10:483–554, 2003.

# References III

[RR02] Svetlozar T Rachev and Werner Römisch. Quantitative stability in stochastic programming: The method of probability metrics. *Mathematics of Operations Research*, 27(4):792–818, 2002.

[Sha91] Alexander Shapiro. Asymptotic analysis of stochastic programs. *Annals of Operations Research*, 30(1):169–186, 1991.

[Sha03] Alexander Shapiro. Monte carlo sampling methods. In *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 353–425. Elsevier, 2003.