Scenario tree generation for stochastic programming models using GAMS/SCENRED

Holger Heitsch\textsuperscript{1} and Steven Dirkse\textsuperscript{2}

\textsuperscript{1} Humboldt-University Berlin, Department of Mathematics, Germany
\textsuperscript{2} GAMS Development Corp., Washington D.C., United States of America

INFORMS Annual Meeting
Overview

What is GAMS/SCENRED about

- GAMS/SCENRED is a link between the well-known General Algebraic Modeling System (GAMS) and the software tool SCENRED
- SCENRED provides a collection of software routines dealing with recent scenario tree manipulation algorithms in stochastic programming
- It is developed at the Department of Mathematics at Humboldt University Berlin by the research group of Prof. Werner Römisch
- A first version of GAMS/SCENRED has been available since 2002
- Now we offer a basically extended version SCENRED2

What is new in SCENRED2

- SCENRED has been extended by scenario tree construction tools
- Available scenario reduction methods are improved by new metrics
- A lot of visualization functions (connected to GNUPLOT) are integrated
Introduction

- Stochastic programs deal with finite sets of scenarios to model the probabilistic information on random data
- The number of scenarios could be very large
- Scenario reduction becomes important to reduce the high sized scenario based models to make them numerical tractable

⇒ Scenario reduction aims to reduce the number of scenarios and to maintain the probability information as good as possible!
About Scenario Reduction

**Probability metrics**

- To control the probability information probability metrics are needed
- Optimal values behave stable with respect to small perturbations of the underlying distribution in terms of probability metrics of the form

\[ d_F(P, Q) = \sup_{f \in F} \left| \int_{\Xi} f(\xi) P(d\xi) - \int_{\Xi} f(\xi) Q(d\xi) \right| \]

*Linear case:*

\[ F_c := \left\{ f : \Xi \to \mathbb{R} \mid f(\xi) - f(\tilde{\xi}) \leq c(\xi, \tilde{\xi}) \text{ for all } \xi, \tilde{\xi} \in \Xi \right\} \]

\[ c(\xi, \tilde{\xi}) := \max \left\{ 1, \|\xi - \xi_0\|^{r-1}, \|\tilde{\xi} - \xi_0\|^{r-1} \right\} \|\xi - \tilde{\xi}\| \]

Metrics of this type are called *Fortet-Mourier metrics* of order \( r \)
About Scenario Reduction

Dual representation

- The dual representation of probability metrics are of the form
  \[ \mu_c(P, Q) = \inf \left\{ \int_{\Xi \times \Xi} c(\xi, \tilde{\xi}) \eta(d\xi, d\tilde{\xi}) : \eta \in M(P, Q) \right\} \]

  These are the Monge-Kantorovich transport functionals

- It holds
  \[ d_{\mathcal{F}_c}(P, Q) \leq \mu_c(P, Q) \quad \text{and} \quad d_{\mathcal{F}_c}(P, Q) = \mu_\hat{c}(P, Q) \]

  for the so-called reduced costs \( \hat{c} \) with

  \[ \hat{c}(\xi, \tilde{\xi}) := \inf \left\{ \sum_{j=1}^{n+1} c(z_{j-1}, z_j) : z_0 = \xi, z_{n+1} = \tilde{\xi}, z_j \in \Xi, n \in \mathbb{N} \right\} \]

  \[ \Rightarrow \text{SCENRED2 allows to control the scenario reduction w.r.t. both} \]
  \[ \text{the Fortet-Mourier metric and the Monge-Kantorovich functional!} \]
About Scenario Reduction

- The dual reformulation allows to compute the scenario reduction without solving the underlying transport problem.
- The problem of optimal scenario reduction is to find convenient scenarios for removing and it can be stated as:

\[
\min \left\{ D_J := \sum_{i \in J} p_i \min_{j \notin J} \hat{c}(\xi^i, \xi^j) \mid J \subset \{1, \ldots, N\}, \#J = N - n \right\}
\]

Approximative solutions by fast (heuristic) algorithms

**Backward Reduction:**
Delete scenario \( u^k \) such that
\[
D_{J_k-1 \cup \{u^k\}} = \min_{u \notin J_{k-1}} D_{J_k-1 \cup \{u\}}
\]

**Forward Selection:**
Select scenerien \( u^k \) such that
\[
D_{J_k-1 \setminus \{u^k\}} = \min_{u \in J_{k-1} \setminus \{u\}} D_{J_k-1 \setminus \{u\}}
\]
About Scenario Reduction

Example 2-dimensional normal distribution

- Scenario reduction of the normal distribution from 10,000 scenarios to 20
About Scenario Reduction

Example 2-dimensional normal distribution

- Scenario reduction of the normal distribution from 10 000 scenarios to 20
Scenario Tree Construction

Introduction

- We consider a multiperiod decision problem of the form

- We have a time discrete stochastic input process \( \xi = (\xi_1, \ldots, \xi_T) \)
- We introduce a decision process \( x = (x_1, \ldots, x_T) \), where the stage decision \( x_t \) only depends on outcomes \( \xi_1, \ldots, \xi_t \)

⇒ Observation: A multistage stochastic program implies a certain information structure: \( \mathcal{F}_1(\xi) \subseteq \ldots \subseteq \mathcal{F}_t(\xi) \subseteq \ldots \subseteq \mathcal{F}_T(\xi) \)

\( (\mathcal{F}_t(\xi) \) denotes the \( \sigma \)-field generated by \( (\xi_1, \ldots, \xi_t) \) \)
Scenario Tree Construction

Assuming that the support of $\xi$ is infinite:

- We have an infinite dimensional optimization problem
- Optimization problem is intractable in general

Replace $\xi$ by a scenario tree approximation $\xi_{tr}$ (finite distribution)

How does the optimal value change when $\xi$ is replaced by $\xi_{tr}$?
Scenario Tree Construction

**Theorem (Stability – Heitsch/Römisch/Strugarek 06)**

Under some regularity assumptions it holds for \( \| \xi - \tilde{\xi} \|_r < \delta \):

\[
|v(\xi) - v(\tilde{\xi})| \leq L \left( \| \xi - \tilde{\xi} \|_r + D_f(\xi, \tilde{\xi}) \right),
\]

where \( v(\cdot) \) denotes the optimal value and \( D_f(\xi, \tilde{\xi}) \) is a distance of the filtrations defined by \( \xi \) and \( \tilde{\xi} \), respectively.

The filtration (information) distance is defined by

\[
D_f(\xi, \tilde{\xi}) := \inf_{x \in S(\xi)} \sum_{t=2}^{T-1} \max \left\{ \| x_t - \mathbb{E}[x_t|\mathcal{F}_t(\tilde{\xi})] \|_r, \| \tilde{x}_t - \mathbb{E}[\tilde{x}_t|\mathcal{F}_t(\xi)] \|_r \right\}
\]

Here \( S(\xi) \) denotes the solution set of the model with input \( \xi \).
Scenario Tree Construction

General approach

1. Providing of scenarios $\xi^i$ with probabilities $p_i$, $i = 1, \ldots, N$:
   - Adaption of a statistic model for the underlying data process (decomposition of historical data, cluster analysis, time series models, stress scenarios)
   - Simulation of scenarios out of the statistic model (may be a large number of scenarios)

2. Construction of the scenario tree out of scenarios $\xi^i$ based on stagewise approximations:
   - Choose a construction $\varepsilon$-percentage (should depend on the number of scenarios)
   - Determine a scenario tree $\xi_{tr}$ by recursive scenario reduction (both the probability distance and the filtration distance can be controlled by this approach)
Scenario Tree Construction

Recursive forward scenario reduction:

![Scenario Tree Diagrams]

1. $t = 1$
2. $t = 2$
3. $t = 3$
4. $t = 4$
5. $t = 5$

Scenario tree generation for stochastic programming models using GAMS/SCENRED Holger Heitsch and Steven Dirkse INFORMS – 2008
Scenario Tree Construction

Recursive **backward** scenario reduction:

Scenario tree generation for stochastic programming models using GAMS/SCENRED

Holger Heitsch and Steven Dirkse

INFORMS – 2008
Example Problem

Stochastic purchase problem

\[
\begin{align*}
\min \left\{ \mathbb{E} \left[ \sum_{t=1}^{3} \xi_t x_t \right] \middle| x_t \geq 0, \\
x_t \text{ is } \mathcal{F}_t(\xi)-\text{measurable}, \\
x_1 + x_2 + x_3 \geq 1
\right\}
\end{align*}
\]

Assumptions

- \( \xi_1 \) is deterministic and \( \xi_1 \equiv 1 \)
- \( \xi_2 \sim U([0, 1]) \) (uniformly distributed)
- \( \xi_3 \sim L([0, 1]) \) (linear distributed) with slope depending on \( \xi_2 \):

\[
\mathbb{P}(\xi_3 \in [a, b] \mid \xi_2 = x) = \int_a^b \left[ 2(1 - x) - 2(1 - 2x)y \right] dy
\]
Example Problem

Probability distribution

Joint density function of the stochastic components \((\xi_2, \xi_3)\)
Example Problem

Analytical solution

Optimal decision

\[ x_1 \equiv 0, \quad x_2 = \begin{cases} 1 & \text{if } \xi_2 \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}, \quad x_3 = \begin{cases} 1 & \text{if } \xi_2 > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \]

Optimal value (OPT) / Value of perfect information (VOPI)

OPT = 0.4167  VOPI = 0.3667
Example Problem

Construction of the scenario tree

![Scenario tree generation for stochastic programming models using GAMS/SCENRED](image-url)
Example Problem

Construction of the scenario tree

- Simulation of a scenario sample $\xi^1, \ldots, \xi^N$ based on the distribution
Example Problem

Construction of the scenario tree

- Simulation of a scenario sample $\xi^1, \ldots, \xi^N$ based on the distribution
- GAMS/SCENRED2 allows to generate a scenario tree

![Scenario tree generation for stochastic programming models using GAMS/SCENRED](image-url)
Example Problem

\[ \text{OPT}^* = 0.4167 \quad / \quad \text{VOPI}^* = 0.3667 \]

Numerical results

<table>
<thead>
<tr>
<th>Sample</th>
<th>Scenarios</th>
<th>Tree size</th>
<th>OPT</th>
<th>VOPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>500</td>
<td>200</td>
<td>0.4156</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>0.4179</td>
<td>0.3629</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>0.4164</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>500</td>
<td>200</td>
<td>0.4162</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>0.4197</td>
<td>0.3640</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>0.4179</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>500</td>
<td>200</td>
<td>0.4245</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>0.4261</td>
<td>0.3749</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>0.4246</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>500</td>
<td>200</td>
<td>0.4092</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>0.4121</td>
<td>0.3632</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>0.4114</td>
<td></td>
</tr>
</tbody>
</table>
Using GAMS/SCENRED

Organization of the GAMS program

1. Data:
   - set & parameter declarations and definitions
   - include SCENRED symbols by: $libinclude scenred.gms
   - setup scenarios (by nodes) and options for SCENRED run

2. SCENRED call:
   - export data from GAMS to SCENRED (using GDX unload)
   - execute SCENRED or SCENRED2
   - import data from SCENRED to GAMS (using GDX load)

3. Model:
   - variable & equation declarations and definitions
   - model definitions using node subsets of reduced/constructed tree
   - solve the model

Example: An implementation of the stochastic purchase example problem is available as GAMS program (‘srpurchase.gms’)