Stochastic Programming

in GAMS

Lutz Westermann
lwestermann@gams.com

GAMS Software GmbH
GAMS Development Corporation
www.gams.com
2010
Example Model: Gas Price Model

Natural Gas NYMEX Weekly Price Chart

Inject/Buy
Withdraw/Sell

Gas Storage
n-Stage Stochastic Programs

• Construct Scenario Tree:
  – Start with today’s price and use a (discrete) distribution
  – Realizations: up, down

• Stochastic Linear Program (block structure)
  – Nested Bender’s Decomposition (OSLSE, FortSP, AIMMS)
  – In practice Deterministic Equivalent with Barrier method

\[
Z_{RN} = \min_{x_i} \left\{ c_1 x_1 + E_{x_2} \left[ \min_{x_3} c_2 x_2 + E_{x_3} \left[ \min_{x_4} c_3 x_3 + \ldots + E_{x_T} \left[ \min_{x_T} c_T x_T \right] \right] \right] \right\}
\]

subject to:

\[
\begin{align*}
A_{t1} x_1 & = b_1 \\
A_{t1} x_1 + A_{t2} x_2 & = b_2 \\
A_{t1} x_1 + A_{t2} x_2 + A_{t3} x_3 & = b_3 \\
\vdots & \vdots \\
A_{T1} x_1 + A_{T2} x_2 + A_{T3} x_3 + \ldots + A_{TT} x_T & = b_T \\
\ell_i & \leq x_i \leq u_i
\end{align*}
\]
ScenRed (Römisch et. al., HU Berlin)

• Find good approximation of original scenario tree of significant smaller size
• Available since 2002
• Integrated in GAMS system
• No extra cost
Tree Generation: ScenRed2

- Construct a true scenario tree from independent scenarios:

- Reconstruct underlying distribution from a set of scenarios
2-Stage Stochastic Programs

- SP Solver DECIS (Gerd Infanger, Stanford, USA)
  - Stores only one instance of the problem and generates scenario sub-problems as needed
  - Solution Strategies
    - Deterministic Equivalent (all scenarios)
    - Sampling: Crude Monte Carlo/Importance sampling
AML and Stochastic Programming (SP)

• Algebraic Modeling Languages/Systems good way to represent optimization problems
  – Algebra is a universal language
  – Hassle free use of optimization solvers
  – Simple connection to data sources (DB, Spreadsheets, …) and analytic engines (GIS, Charting, …)

• Large number of (deterministic) models in production
  – Opportunity for seamless introduction of new technology like Global Optimization, Stochastic Programming, …
  – AML potential framework for SP
Simple Example
Simple Example: Newsboy (NB) Problem

• Data:
  – A newsboy faces a certain demand for newspapers \( d = 45 \)
  – He can buy newspapers for fixed costs per unit \( c = 30 \)
  – He can sell newspapers for a fixed price \( r = 60 \)
  – For hold units he has to pay a disposal fee \( h = 10 \)
  – He has to satisfy his customers demand or has to pay a penalty \( p = 5 \)

• Decisions:
  – How many newspapers should he buy: \( X \) 45
  – How many newspapers should he sell: \( S \) 45

• Derived Outcomes:
  – How many newspapers need to be disposed: \( I \) 0
  – How many customers are lost: \( L \) 0
Simple NB Problem – GAMS Formulation

* LostSales = demand - UnitsSold
lSales.. L =e= d - S;

* Inventory = UnitsBought - UnitsSold
Inv.. I =e= X - S;

* Profit, to be maximized
Profit.. Z =e= r*S - c*X - h*I - p*L;

Model nb / all /;

solve nb max z use lp;
NB Problem – Add Uncertainty

• Uncertain demand $d$

• Decisions to make:
  – How much newspaper should he buy “here and now” (without knowing the outcome of the uncertain demand)?
    → *First-stage decision*
  – How many customers are lost after the outcome becomes known?
    → *Second-stage or recourse decision*
  – Recourse decisions can be seen as
    • penalties for bad first-stage decisions
    • variables to keep the problem feasible
* Make d uncertain
randvar d discrete 0.2 40
                        0.7 45
                        0.1 50

* Define non-default stages
stage 2 d I L S
stage 2 lSales Inv
New GAMS (EMP) Keywords
Excursus: EMP, what?

With new modeling and solution concepts do not:
• overload existing GAMS notation right away!
• attempt to build new solvers right away!

But:
• Use existing language features to specify additional model features, structure, and semantics
• Express extended model in symbolic (source) form and apply existing modeling/solution technology
• Package new tools with the production system

→ Extended Mathematical Programming (EMP)
JAMS: a GAMS EMP Solver

EMP Information  
Original Model  
Translation  
Reformulated Model  
Solving using established Algorithms  
Solution  
Mapping Solution into original space  
Viewable
Random Variables

Discrete Distribution

Normal Distribution

Poisson Distribution

Exponential Distribution
Random Variables (RV) \([\text{randVar}]\)

-Defines both discrete and parametric random variables:

\[
\text{randVar rv discrete prob val \{prob val\}}
\]

- The distribution of discrete random variables is defined by pairs of the probability \(\text{prob}\) of an outcome and the corresponding realization \(\text{val}\)

\[
\text{randVar rv distr par \{par\}}
\]

- The name of the parametric distribution is defined by \(\text{distr}, \text{par}\) defines a parameter of the distribution
Joint Random Variables

Demand

- Prob: 0.2, d: 40
- Prob: 0.7, d: 45
- Prob: 0.1, d: 50

Demand / Price

- Prob: 0.04, d: 40 / p: 55
- Prob: 0.14, d: 40 / p: 60
- Prob: 0.02, d: 40 / p: 65

Price

- Prob: 0.2, p: 55
- Prob: 0.7, p: 60
- Prob: 0.1, p: 65

Demand / Price

- Prob: 0.04, d: 40 / p: 55
- Prob: 0.14, d: 40 / p: 60
- Prob: 0.02, d: 40 / p: 65

vs.

Demand / Price

- Prob: 0.2, d: 40
- Prob: 0.7, d: 45
- Prob: 0.1, d: 50

vs.

- Prob: 0.04, d: 40 / p: 55
- Prob: 0.14, d: 40 / p: 60
- Prob: 0.02, d: 40 / p: 65

- Prob: 0.07, d: 45 / p: 55
- Prob: 0.07, d: 50 / p: 60
- Prob: 0.01, d: 50 / p: 65
Joint RVs \([j\text{RandVar}]\)

- Defines discrete random variables and their joint distribution:

  \[
  j\text{RandVar} \text{ rv rv \{rv\} prob val val \{val\} \{prob val val \{val\}\}}
  \]

- At least two discrete random variables \(\text{rv}\) are defined and the outcome of those is coupled
- The probability of the outcomes is defined by \(\text{prob}\) and the corresponding realization for each random variable by \(\text{val}\)
Correlation between RVs [correlation]

- Defines a correlation between a pair of random variables:

  \texttt{correlation rv rv val}

- \texttt{rv} is a random variable which needs to be specified using the \texttt{randvar} keyword and \texttt{val} defines the desired correlation (-1 \leq \texttt{val} \leq 1)
Stages

Stage 1: Decision X

Observation of d

Stage 2: Decisions S, I, L

...
Stages [stage]

• Defines the stage of random variables (rv), equations (equ) and variables (var):

  stage stageNo rv | equ | var {rv | equ | var}

• StageNo defines the stage number
• The default StageNo for the objective variable and objective equation is the highest stage mentioned
• The default StageNo for all the other random variables, equations and variables not mentioned is 1
Chance Constraints

OBJ.. Z =e= X1 + X2;
E1.. om1*X1 + X2 =g= 7;
E2.. om2*X1 + 3*X2 =g= 12;
Model sc / all /;
solve sc min z use lp;

om1

Prob: 0.25
val: 1
Prob: 0.25
val: 2
Prob: 0.25
val: 3
Prob: 0.25
val: 4

Om2

Prob: 0.33
val: 1
Prob: 0.33
val: 2
Prob: 0.33
val: 3

chance E1 0.6
chance E2 0.6
Chance Constraints

3 out of 4 must be true
[0.75 \geq 0.6]

- \[ 1\times X_1 + X_2 \geq 7; \]
- \[ 2\times X_1 + X_2 \geq 7; \]
- \[ 3\times X_1 + X_2 \geq 7; \]
- \[ 4\times X_1 + X_2 \geq 7; \]

2 out of 3 must be true
[0.66 \geq 0.6]

- \[ 1\times X_1 + 3\times X_2 \geq 12; \]
- \[ 2\times X_1 + 3\times X_2 \geq 12; \]
- \[ 3\times X_1 + 3\times X_2 \geq 12; \]

Just in case: \( X_1 = 2 \) and \( X_2 = 3 \) are optimal.
Chance Constraints \([\text{chance}]\)

- Defines individual or joint chance constraints (CC):
  \[
  \text{chance equ } \{\text{equ}\} \ [\text{holds}] \ \text{minRatio} \ [\text{weight} \vert \text{varName}]
  \]

- Individual CC: A single constraint \(\text{equ}\) has to hold for a certain ratio \((0 \leq \text{minRatio} \leq 1)\) of the possible outcomes
- Joint CC: A set of constraints \(\text{equ}\) has to hold for a certain ratio \((0 \leq \text{minRatio} \leq 1)\) of the possible outcomes
- If \text{weight} is defined, the violation of a CC gets penalized in the objective \((\text{weight \ violationRatio})\)
- If \text{varName} is defined the violation get multiplied by this existing variable
Expected Value [ExpectedValue]

- This is the default objective:

  \[ \text{ExpectedValue} \ [x \ EV_x] \]

- If only \text{ExpectedValue} is defined, the expect value of the GAMS objective variable will be optimized (same as if it would be omitted at all)

- If the variable pair \( x \ EV_x \) is defined, GAMS will replace its objective variable by \( EV_x \), which will become the expected value of \( x \)
Conditional Value at Risk [cVaR]

- As an alternative to the expected value, the conditional value at risk (cVaR) can be optimized:

  \[ \text{cVaR} \ [x \ \text{cVaR}_x] \ \theta \]

- If only \( \text{cVaR} \ \theta \) is defined, the cVaR of the GAMS objective variable to the quantile level \( \theta \) will be optimized.

- If the variable pair \( x \ \text{cVaR}_x \) is defined, GAMS will replace its objective variable by \( \text{cVaR}_x \), which will become the cVaR of \( x \) to the quantile level \( \theta \).
Combining EV and cVaR

It is also possible optimize a combination of the expected value and the conditional value at risk like this:

```plaintext
... 
defobj..
    obj =e= lambda*EV_r + (1-lambda)*CVaR_r;

ExpectedValue r EV_r
cvarlo r CVaR_r 0.1
... 
```
Output Extraction

• The expected value of the solution can be accessed via the regular .L and .M fields

• In addition, the following information can be stored in a parameter by scenario:
  - level: Levels of variables or equations
  - marginal: Marginals of variables or equations
  - randvar: Realization of a random variable
  - opt: Probability of each scenario

• This needs to be stored in a separate dictionary:

```gams
Set dict / scen.scenario .''
  x .level .s_x
  '' .opt .srep /;
```
Adding Uncertainty to Transport

...  

Model transport /all/ ;

file emp / '%emp.info%' /; put emp '* problem %gams.i%'/
$onput
randvar b('new-york') normal 325 50
randvar b('chicago')  normal 300 50
randvar b('topeka')   normal 275 50
stage 2 b demand
$offput
putclose emp;

Set scen scenarios / s1*s6 /;
Parameter
  s_b(scen,j)       demand realization by scenario
  s_x(scen,i,j)    shipment per scenario
  s_s(scen) ;

Set dict / scen .scenario.'''
  b   .randvar .s_b
  x   .level   .s_x /;

Solve transport using emp minimizing z scenario dict;
Summary
<table>
<thead>
<tr>
<th></th>
<th>DE</th>
<th>DECIS</th>
<th>LINDO</th>
</tr>
</thead>
<tbody>
<tr>
<td>chance</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>correlation</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>cVaR</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>expectedValue</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>jrandVar</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>randVar (discrete)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>randVar (parametric)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Conclusion

• Deterministic examples from all kind of application areas exist already (e.g. ~400 in the GAMS Model Library)

• Easy to add uncertainty to existing deterministic models, to
  – ... either use specialized algorithms (DECIS, LINDO)
  – ... or create Deterministic Equivalent and select from wide range of existing GAMS solver links (DE, free)

• New SP examples in the GAMS EMP Library

• More work to be done:
  – Scenario tree support
  – Sampling
  – ...
## Contacting GAMS

<table>
<thead>
<tr>
<th>Europe</th>
<th>USA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GAMS Software GmbH</strong></td>
<td><strong>GAMS Development Corp.</strong></td>
</tr>
<tr>
<td>Eupener Str. 135-137</td>
<td>1217 Potomac Street, NW</td>
</tr>
<tr>
<td>50933 Cologne</td>
<td>Washington, DC 20007</td>
</tr>
<tr>
<td>Germany</td>
<td>USA</td>
</tr>
<tr>
<td>Phone: +49 221 949 9170</td>
<td>Phone: +1 202 342 0180</td>
</tr>
<tr>
<td>Fax: +49 221 949 9171</td>
<td>Fax: +1 202 342 0181</td>
</tr>
</tbody>
</table>

- [sales@gams.com](mailto:sales@gams.com)
- [support@gams.com](mailto:support@gams.com)

http://www.gams.com