An introduction to stochastic programming

H.I. Gassmann
Overview

- Introduction
- A taxonomy of stochastic programming problems
- Algorithms
- Instance representations
- An XML format for stochastic programs
- Conclusions
Stochastic programming

• Decision making under uncertainty
• Very general class of problems:
  – How to create and manage a portfolio
    • Optimal investment sequences, given
      – Historic distribution of returns and covariances
      – Horizon, financial goals, regulatory constraints, etc.
  – How to harvest a forest
    • Optimal harvest sequence, given
      – Random incidence of forest fires, pest, etc.
  – How to generate power
    • Random data on demand, rates, parameters
  – etc.
Common characteristics

• Large-scale optimization models
• Some problem parameters unknown
• Assume distribution of parameters known
• (Otherwise: Optimization under risk)
Multistage stochastic linear program

"min" \[ c_0 x_0 + c_1 x_1 + K + c_T x_T \]
\[ \text{s.t.} \quad A_{00} x_0 \sim b_0 \]
\[ R_1 x_0 \Delta r_1 \]
\[ A_{10} x_0 + A_{11} x_1 \Delta b_1 \]
\[ M \quad M \quad O \quad M \]
\[ A_{T0} x_0 + A_{T1} x_1 + K + A_{TT} x_T \Delta b_T \]
\[ l_0 \leq x_0 \leq u_0 \]
\[ l_i \leq x_i \leq u_i, t = 1, K, T \]

Any data item with nonzero subscript may be random
(including dimensions where mathematically sensible)

~ stands for arbitrary relation (\(\leq, =, \geq\))

© 2006 H.I. Gassmann
Constraints involving random elements

\[ A_{t0}x_0 + A_{t1}x_1 + K + A_{tt}x_t \Delta b_t \]

\( \Delta \) means \( \sim \) with probability 1

or with probability at least \( \beta \)

or with expected violation at most \( v \)

or \( \ldots \)
Problem classes

• Recourse problems
  – All constraints hold with probability 1
  – Minimize expected objective value

• Chance-constrained problems
  – Typically single stage

• Hybrid problems
  – Recourse problems including probabilistic constraints (VaR) or integrated chance constraints (CVaR)
  – Regulatory necessity
  – Often modelled using integer variables and/or linking constraints

• Distribution problems
  – Determine distribution of optimum objective and/or decisions
Taxonomy of stochastic programming problems

- Stochastic Programs
  - Chance-Constrained Problems
  - Recourse Problems
  - Distribution Problems
  - Hybrid Problems
    - Individual Chance-Constraint
    - Joint Chance-Constraint
    - Mixed Problems
    - Explicit Event Tree
    - Implicit Event Tree

- Independent random variables
- Period-to-period independence
- Random walk
- Dependence on past data
- Random problem dimensions
- Dependence on past decisions
Event trees for finite distributions

- Display evolution of information
Algorithms for recourse problems

• Direct solution of the deterministic equivalent
  – “Curse of dimensionality”

• Decomposition
  – Recognize structure
  – Repeated calls to solver with different data
  – Configuration and sequencing of subproblems
Benders Decomposition

- Decompose event tree into nodal problems:

\[
\begin{align*}
\min & \quad c_n x_n + \vartheta_n \\
\text{s.t.} & \quad A_n x_n = b_n - B_n x_{a(n)} \\
& \quad D_n x_n = d_n \quad \text{(feasibility cuts)} \\
& \quad E_n x_n + \vartheta_n = e_n \quad \text{(optimality cuts)}
\end{align*}
\]

- In sequence solve each problem repeatedly
- Pass primal information to successors
- Pass dual information to ancestors (cuts)
Algorithm variants

• Different decomposition schemes
  – Path by path
  – Several stages at once
  – etc.

• Stochastic decomposition
  – Sequential sampling of subproblems
  – Suitable for continuous distributions
  – Convergence in probability
Numerical results

• Problem 1: WATSON
  – Ten-stage financial investment problem
  – Various numbers of scenarios
  – Largest DE: around 700,000 variables

• Problem 2: STOCHFOR
  – Stochastic forestry problem
  – Varying number of time stages
  – Largest DE: around 500,000 variables
## Problem characteristics

<table>
<thead>
<tr>
<th></th>
<th>StochFor</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>stages</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>scenarios</td>
<td>729</td>
<td>2,187</td>
<td>6,561</td>
<td>19,683</td>
<td>16</td>
<td>128</td>
<td>768</td>
</tr>
<tr>
<td>nodes</td>
<td>1,093</td>
<td>3,280</td>
<td>9,841</td>
<td>29,524</td>
<td>111</td>
<td>511</td>
<td>1,534</td>
</tr>
<tr>
<td>nrows</td>
<td>19,672</td>
<td>59,038</td>
<td>177,136</td>
<td>531,430</td>
<td>4,684</td>
<td>26,748</td>
<td>102,132</td>
</tr>
<tr>
<td>ncols</td>
<td>17,487</td>
<td>52,479</td>
<td>157,455</td>
<td>472,383</td>
<td>8,401</td>
<td>49,153</td>
<td>191,994</td>
</tr>
<tr>
<td>nelem (DE)</td>
<td>76,467</td>
<td>229,557</td>
<td>688,827</td>
<td>2,066,637</td>
<td>21,368</td>
<td>128,648</td>
<td>526,078</td>
</tr>
</tbody>
</table>

© 2006 H.I. Gassmann
## Watson problem – CPU time

<table>
<thead>
<tr>
<th>configuration</th>
<th>scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td>cplex</td>
<td>0.25</td>
</tr>
<tr>
<td>cplex-b</td>
<td>0.19</td>
</tr>
<tr>
<td>nodes</td>
<td>0.59</td>
</tr>
<tr>
<td>path</td>
<td>3.91</td>
</tr>
<tr>
<td>3-ply</td>
<td>1.41</td>
</tr>
</tbody>
</table>
StochFor problem – CPU time

<table>
<thead>
<tr>
<th>configuration</th>
<th>Periods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td>cplex</td>
<td>1.67</td>
</tr>
<tr>
<td>cplex-b</td>
<td>1.27</td>
</tr>
<tr>
<td>nodes</td>
<td>2.03</td>
</tr>
<tr>
<td>path</td>
<td>2.42</td>
</tr>
<tr>
<td>3-ply</td>
<td>16.69</td>
</tr>
</tbody>
</table>
StochFor problem - complexity

Number of stages

© 2006 H.I. Gassmann
Modelling support

- Often $O(10^6)$ variables and constraints
- Need computer support
  - Algebraic modelling language (AML)
    - How to express random entities?
    - How to work with random entities?
  - Databases
    - How to link to AMLs?
  - Visualization
    - How to present solution and other problem components?
What is an instance?

- Role and number of constraints, objectives, parameters and variables must be known
- Every parameter’s value must be known
- Continuous entities vs. discretization
  - Decision variables
  - Objective and constraints
  - Distribution of random variables
  - Time domain
What is a stage?

- Stages form a subset of the time structure
- Stages comprise both decisions and events
- Events must either precede all decisions or follow all decisions
- Should a stage be decision – event or event – decision?
Why is there a problem?

• AMPL-like declarations:

```AMPL
set time ordered;
param demand{t in time} random;
Production_balance {t in time}:
  Inv[t-1] + product[t] >= demand[t] + Inv[t];
```

• Is the constraint well-posed?

• At least two possible interpretations
  – Inv[t] set after demand[t] known: recourse form, well-posed
  – Inv[t] set before demand[t] known: undeclared chance constraint
Instance representation

- SMPS format
- Algebraic modelling languages
- Internal representations
- XML format
Example (Birge)

\[
\text{max} \sum_{s=1}^{S} p_s (w_s - \beta u_s)
\]

s.t. \[
\sum_{i=1}^{I} x_{0i} - \sum_{i=1}^{I} x_{lis} = -B
\]

\[
\sum_{i=1}^{I} \alpha_{0is} x_{0i} - \sum_{i=1}^{I} \alpha_{lis} x_{lis} = 0, s \in S_1
\]

\[
\sum_{i=1}^{I} \alpha_{t-1,i,s} x_{t-1,i,a(s)} - \sum_{i=1}^{I} x_{tis} = 0, s \in S_t, t = 2, K, T - 1
\]

\[
\sum_{i=1}^{I} \alpha_{T-1,i,s} x_{T-1,i,a(s)} + u_s - w_s = R, s \in S_T
\]

\[x_{tis}, u_s, w_s \geq 0\]

I = 2, T = 3, B = 55, R = 80, 
\[\alpha_{t1} = \{1.25, 1.06\},\]
\[\alpha_{t2} = \{1.14, 1.12\}\]
SMPS format

• Three files based on MPS format
  – Core file for deterministic problem components
  – Time file for dynamic structure
  – Stoch file for stochastic structure

• Disadvantages
  – Old technology
  – Limited precision (12 digits, including sign)
  – Limited name space (8 characters)
  – Direction of optimization (min/max) ambiguous
  – Linear constraints, quadratic objective only
## Example (Birge)

### Core file

**ROWS**
- Budget0
- Object
- Budget1
- Budget2
- Budget3

**COLS**
- X01
- ...  
- ...  
- RHS
- rhs1
- rhs1

**ENDATA**

### Stoch file

**BLOCKS**
- BL Block1
- BL Block1
- BL Block1

**DISCRETE**
- 0.5
- ...  
- ENDATA
Algebraic modelling languages

• Characteristics
  – Similar to algebraic notation
  – Powerful indexing capability
  – Data verification possible

• Disadvantages
  – Discrete distributions only
  – Limited consistency checks for stochastic structure
AMPL model

param T;
param penalty;
param budget;
param target;
set instruments;
set scenarios;
param prob{scenarios};
set slice{t in 0..T} within scenarios;
param ancestor {t in 1..T, s in slice[t]};
var over {slice[T]};
var under{slice[T]};
param return {t in 1..T, i in instruments,s in slice[t]};
var invest {t in 0..T-1,i in instruments,s in slice[t]};

maximize net_profit:
    sum{s in scenarios} prob[s]*(over[s] - penalty*under[s]);

subject to wealth{t in 0..T, s in slice[t]}:
    (if t < T then sum{i in instruments} invest[t,i,s]) =
    (if t = 0 then budget 
        else sum {i in instruments}
            return[t,i,s]*invest[t-1,i,ancestor[t,s]])
    + if t = T then (under[s] - over[s] + target);
Internal representations

• Seek most compact representation possible
  – Sparse matrix format is insufficient
  – Blocks corresponding to nodes in the event tree
  – Change blocks if problem dimensions are deterministic
    \[ A_{stj} = A_{st0} + \Delta A_{stj} \] (addition or replacement)
  – Exploit period-to-period independence
### Storage requirements

<table>
<thead>
<tr>
<th></th>
<th>Watson 1</th>
<th>Watson 2</th>
<th>Watson 3</th>
<th>Watson 4</th>
<th>Watson 5</th>
<th>Forest 1</th>
<th>Forest 2</th>
<th>Forest 3</th>
<th>Forest 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>stages</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>scenarios</td>
<td>16</td>
<td>128</td>
<td>768</td>
<td>2,688</td>
<td></td>
<td>729</td>
<td>2,187</td>
<td>6,561</td>
<td>19,683</td>
</tr>
<tr>
<td>nodes</td>
<td>111</td>
<td>511</td>
<td>1,534</td>
<td>5,363</td>
<td></td>
<td>1,093</td>
<td>3,280</td>
<td>9,841</td>
<td>29,524</td>
</tr>
<tr>
<td>nrows</td>
<td>4,684</td>
<td>26,748</td>
<td>102,132</td>
<td>357,376</td>
<td></td>
<td>19,672</td>
<td>59,038</td>
<td>177,136</td>
<td>531,430</td>
</tr>
<tr>
<td>ncols</td>
<td>8,401</td>
<td>49,153</td>
<td>191,994</td>
<td>671,861</td>
<td></td>
<td>17,487</td>
<td>52,479</td>
<td>157,455</td>
<td>472,383</td>
</tr>
<tr>
<td>nelem (DE)</td>
<td>21,368</td>
<td>128,648</td>
<td>526,078</td>
<td>1,841,028</td>
<td></td>
<td>76,467</td>
<td>229,557</td>
<td>688,827</td>
<td>2,066,637</td>
</tr>
<tr>
<td>nelem (blk)</td>
<td>21,368</td>
<td>128,648</td>
<td>526,078</td>
<td>1,841,028</td>
<td></td>
<td>43,887</td>
<td>131,397</td>
<td>393,867</td>
<td>1,181,217</td>
</tr>
<tr>
<td>ch_blocks</td>
<td>8,283</td>
<td>45,695</td>
<td>184,800</td>
<td>638,695</td>
<td></td>
<td>30,783</td>
<td>92,049</td>
<td>275,787</td>
<td>826,941</td>
</tr>
<tr>
<td>indep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>927</td>
<td>1,077</td>
<td>1,227</td>
<td>1,377</td>
</tr>
</tbody>
</table>

© 2006 H.I. Gassmann
OSiL – Optimization Services instance Language

• Written in XML
  – Easy to accommodate new features
  – Existing parsers to check syntax
• Easy to generate automatically
• Trade-off between verbosity and human readability
OSiL – Header information
OSiL – Header information

<?xml version="1.0" encoding="UTF8"?>
<OSiL xmlns="os.optimizationservices.org"
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="OSiL.xsd">
<programDescription>
<source>FinancialPlan_JohnBirge</source>
<maxOrMin>max</maxOrMin>
<objConstant>0.</objConstant>
<numberObjectives>1</numberObjectives>
<numberConstraints>4</numberConstraints>
<numberVariables>8</numberVariables>
</programDescription>
<programData>
...
</programData>
</OSiL>

© 2006 H.I. Gassmann
OSiL – Deterministic information
<constraints>
  <con name="budget0" lb="55" ub="55"/>
  <con name="budget1" lb="0" ub="0"/>
  <con name="budget2" lb="0" ub="0"/>
  <con name="budget3" lb="80" ub="80"/>
</constraints>

<variables>
  <var name="invest01" type="C" lb="0.0"/>
  <var name="invest02"/>
  <var name="invest11"/>
  <var name="invest12"/>
  <var name="invest21"/>
  <var name="invest22"/>
  <var objCoef="1" name="w"/>
  <var objCoef="-4" name="u"/>
</variables>
OSiL –
Core matrix (sparse matrix form)

<coefMatrix>
  <listMatrix>
    <start>
      <el>0</el>
      <el>2</el>
      <el>4</el>
      <el>6</el>
      <el>8</el>
      <el>10</el>
      <el>12</el>
      <el>13</el>
    </start>
    <rowIdx>
      <el>0</el>
      <el>1</el>
      <el>1</el>
      <el>2</el>
      <el>1</el>
      <el>2</el>
      <el>2</el>
      <el>3</el>
    </rowIdx>
    <value>
      <el>1</el>
      <el>-1.25</el>
      <el>-1.14</el>
      <el>-1.25</el>
      <el>-1.14</el>
      <el>-1.25</el>
      <el>-1.14</el>
      <el>-1.25</el>
    </value>
  </listMatrix>
</coefMatrix>
Dynamic structure
OSiL – dynamic information

<stages number="4">
  <implicitOrder>
    <el startRowIdx="0" startColIdx="0"/>
    <el startRowIdx="1" startColIdx="2"/>
    <el startRowIdx="2" startColIdx="4"/>
    <el startRowIdx="3" startColIdx="6"/>
  </implicitOrder>
</stages>
Explicit and implicit event trees

© 2006 H.I. Gassmann
Scenario trees
OSiL – Stochastic information

<stochastic>
  <explicitScenario>
    <scenarioTree>
      <sNode prob="1" base="coreProgram">
        <sNode prob="0.5" base="coreProgram">
          <sNode prob="0.5" base="coreProgram">
            <sNode prob="0.5" base="firstSibling">
              <changes>
                <el rowIdx="3" colIdx="4">-1.06</el>
                <el rowIdx="3" colIdx="5">-1.12</el>
              </changes>
            </sNode>
          </sNode>
        </sNode>
      </sNode>
    </scenarioTree>
  </explicitScenario>
</stochastic>
Distributions
Transformations

© 2006 H.I. Gassmann
Penalties and probabilistic constraints

- probabilisticObjectives
  - simpleChance
  - jointChance
  - softConstraints
    - penalties
      - row
      - riskMeasures
      - integratedChance
    - simpleRecourse
      - robustOptimization
      - piecewiseLinearQuadratic
      - userDefinedPenalty
Node-by-node representation for stochastic problem dimensions
Capabilities

- Arbitrary nonlinear expressions
- Arbitrary distributions
- Scenario trees
- Stochastic problem dimensions
- Simple recourse
- Soft constraints with arbitrary penalties
- Probabilistic constraints
- Arbitrary moment constraints
Nonlinear expression –
\[(x_0 - x_1^2)^2 + (1 - x_0)^2\]
Example – discrete random vector

<distributions>
  <multivariate>
  <dist name="dist1">
    <multivariateDiscrete>
      <scenario>
        <prob>0.5</prob>
        <el>-1.25</el>
        <el>-1.14</el>
      </scenario>
      <scenario>
        <prob>0.5</prob>
        <el>-1.06</el>
        <el>-1.12</el>
      </scenario>
    </multivariateDiscrete>
  </dist>
  </multivariate>
</distributions>
Further work

- Readers
- Internal data structures
- Solver interfaces
- Library of problems
- Buy-in
Thank you!

www.optimizatiionservices.org

myweb.dal.ca/gassmann