An introduction to stochastic programming

H.I. Gassmann

DALHOUSIE UNIVERSITY

Inspiring Minds

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

$$
\begin{array}{lccccc}
\text{"min"} & c_0 x_0 & + c_1 x_1 & + \ldots & + c_T x_T & \\
\text{s.t.} & A_{00} x_0 & & & & \sim b_0 \\
 & R_1 x_0 & & & & \sim r_1 \\
 & A_{10} x_0 & + A_{11} x_1 & & & \sim b_1 \\
 & \vdots & \vdots & \ddots & & \vdots \\
 & A_{T0} x_0 & + A_{T1} x_1 & + \ldots & + A_{TT} x_T & \sim b_T
\end{array}
$$

$$l_0 \leq x_0 \leq u_0$$

$$l_t \leq x_t \leq u_t, t = 1, \ldots, T$$

Any data item with nonzero subscript may be random

(including dimensions where mathematically sensible)

~ stands for arbitrary relation (≤, =, ≥)

DALHOUSIE UNIVERSITY Inspiring Minds

Constraints involving random elements

$$A_{t0}x_0 + A_{t1}x_1 + \mathrm{K} + A_{tt}x_t \vartriangle b_t$$

Δ means ~ with probability 1

or with probability at least β

or with expected violation at most v

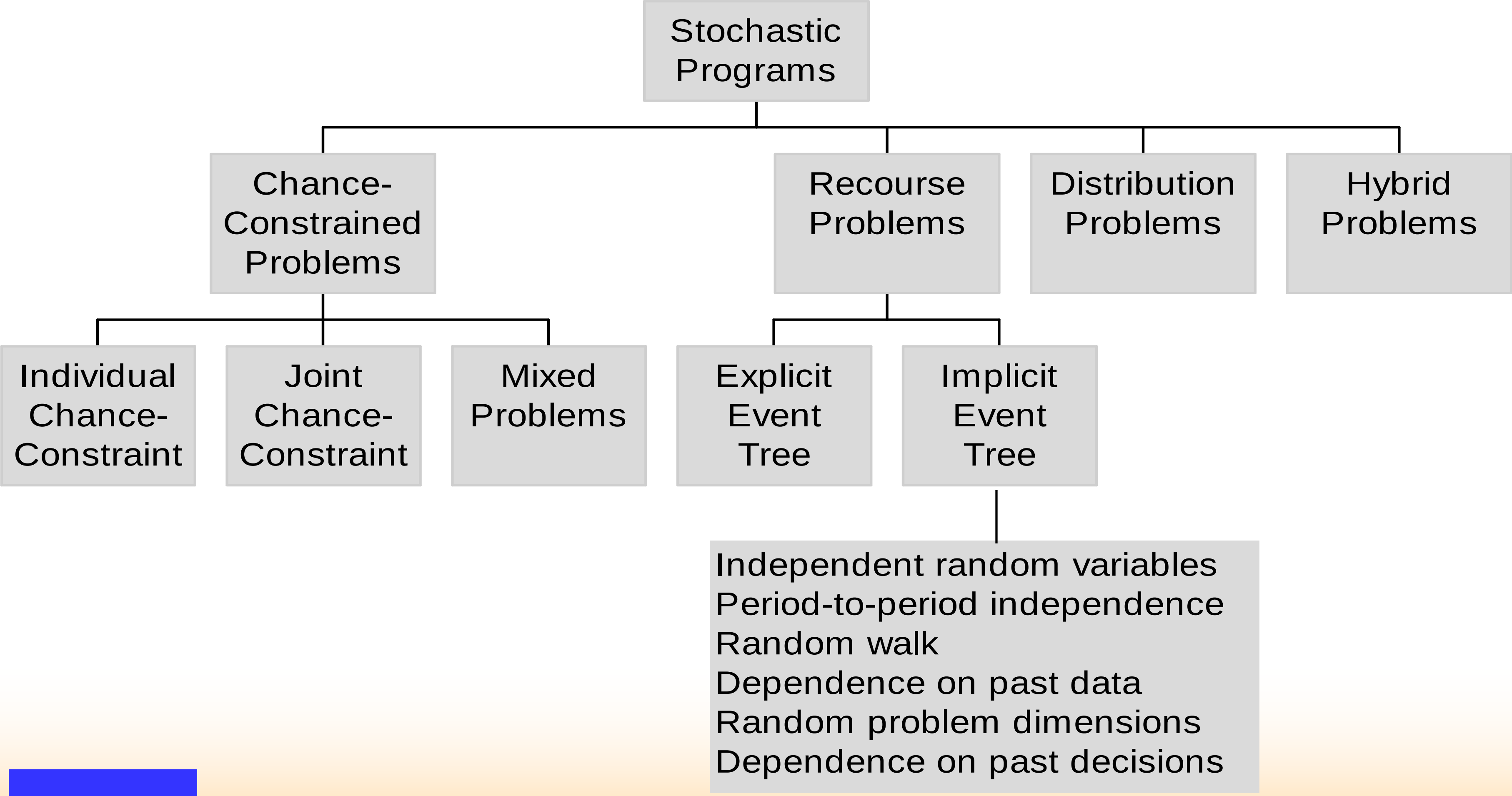
or …

DALHOUSIE UNIVERSITY
Inspiring Minds

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

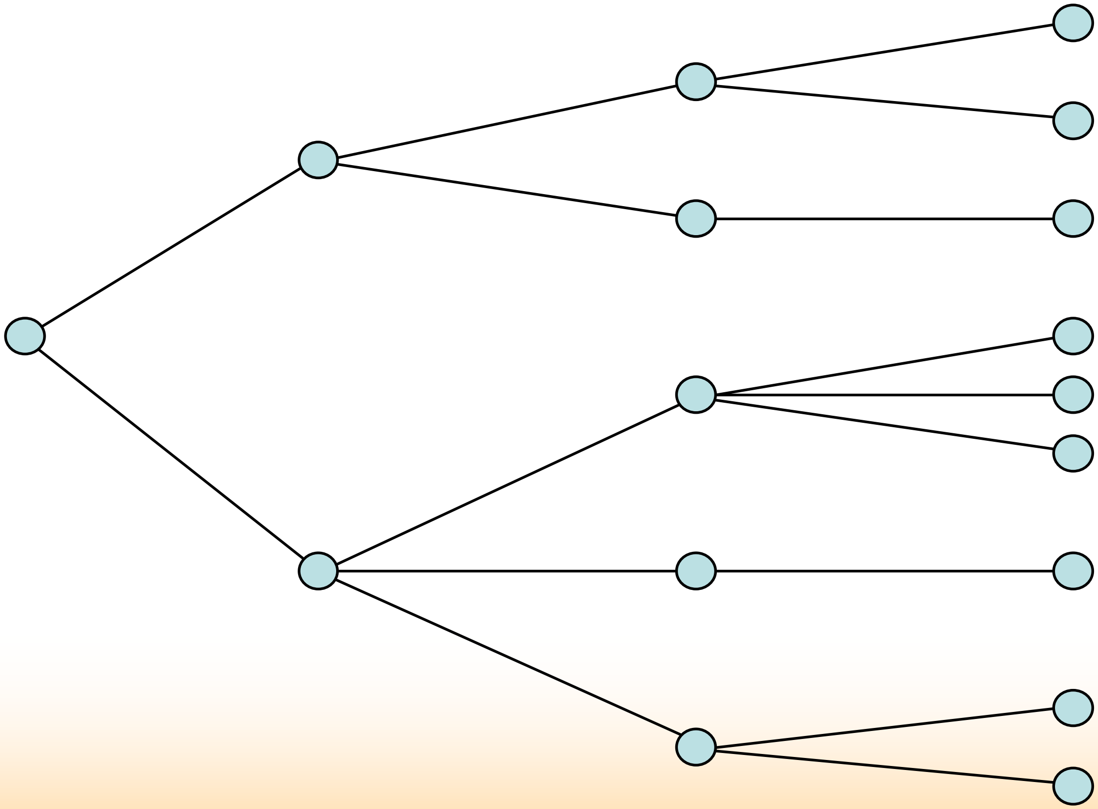
Taxonomy of stochastic programming problems



- Stochastic Programs
  - Chance-Constrained Problems
    - Individual Chance-Constraint
    - Joint Chance-Constraint
    - Mixed Problems
  - Recourse 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
  - Distribution Problems
  - Hybrid Problems

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:

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

- 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

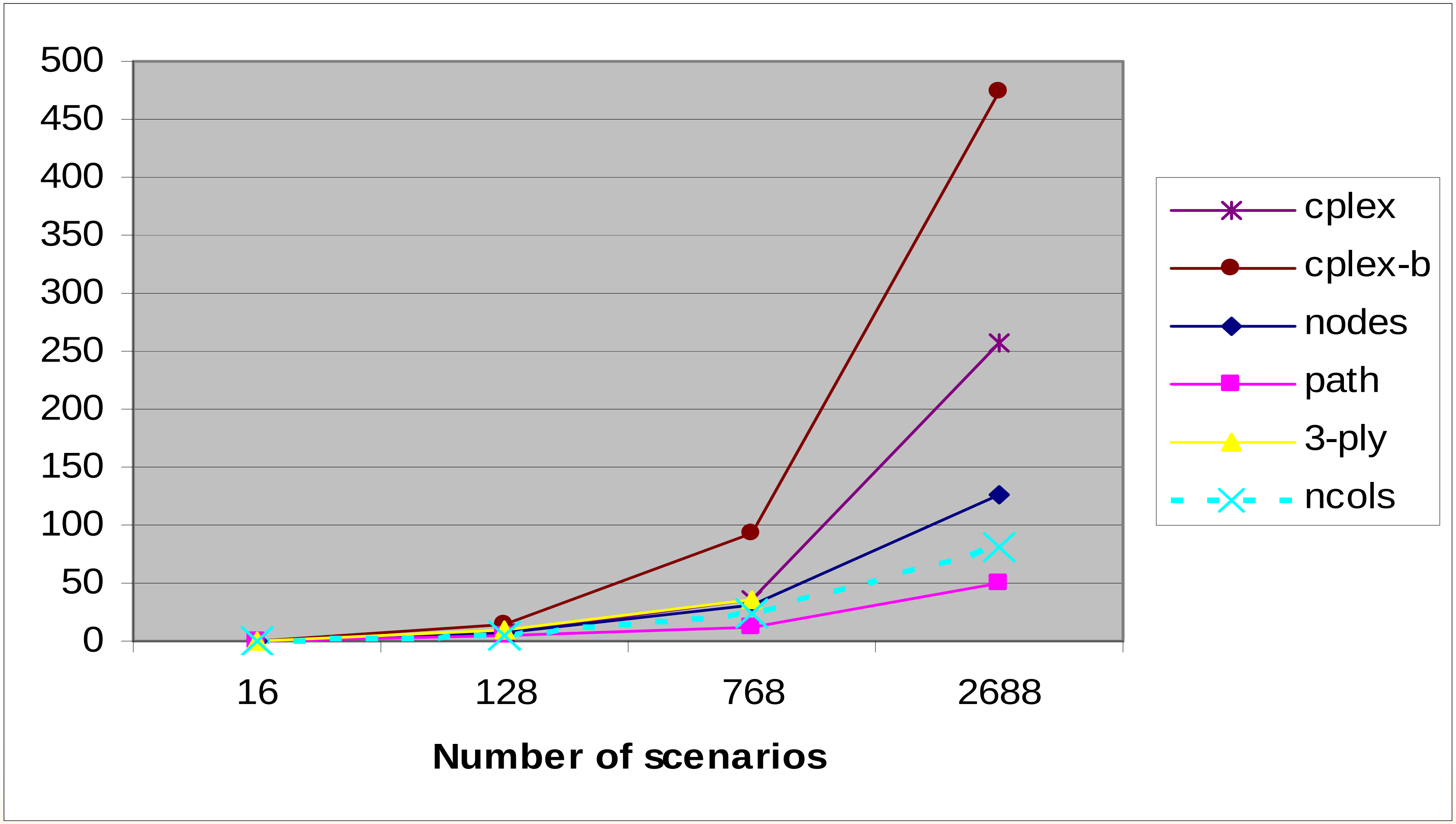
Problem characteristics

	StochFor				Watson			
stages	7	8	9	10	10	10	10	10
scenarios	729	2,187	6,561	19,683	16	128	768	2,688
nodes	1,093	3,280	9,841	29,524	111	511	1,534	5,363
nrows	19,672	59,038	177,136	531,430	4,684	26,748	102,132	357,376
ncols	17,487	52,479	157,455	472,383	8,401	49,153	191,994	671,861
nelem (DE)	76,467	229,557	688,827	2,066,637	21,368	128,648	526,078	1,841,028

Watson problem – CPU time

	scenarios			
configuration	16	128	768	2688
cplex	0.25	1.86	9.19	64.52
cplex-b	0.19	2.5	17.5	90.09
nodes	0.59	4.2	18.33	74.78
path	3.91	16.58	44.59	191.02
3-ply	1.41	12.55	50.59	N.C.

Watson problem - complexity

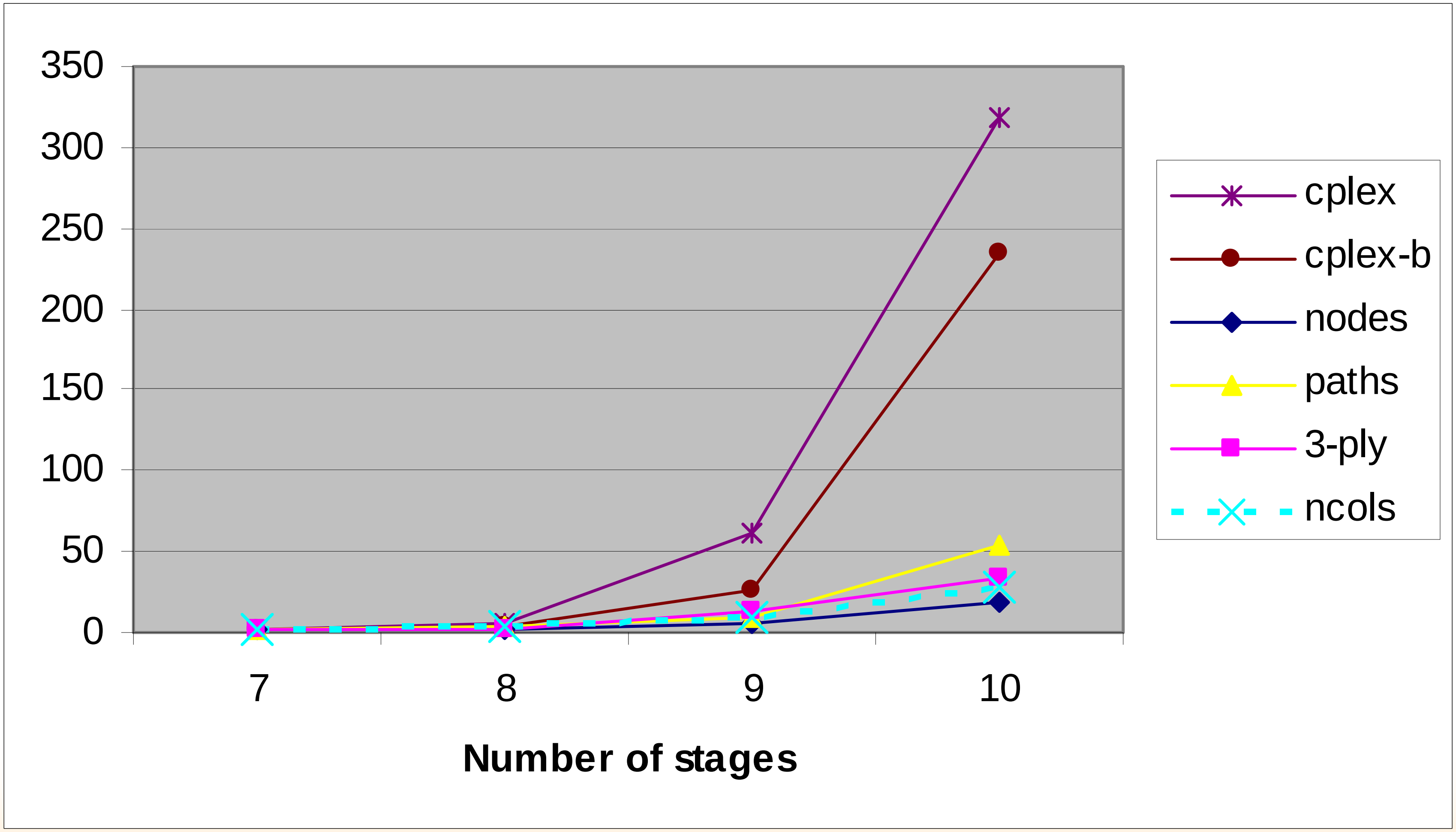


© 2006 H.I. Gassmann

StochFor problem – CPU time

	Periods			
configuration	7	8	9	10
cplex	1.67	8.61	101.97	530.89
cplex-b	1.27	5.81	33.03	298.64
nodes	2.03	3.39	11.64	38.56
path	2.42	7.09	20.72	129.28
3-ply	16.69	27.52	202.02	549.72

StochFor problem - complexity



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:

```
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

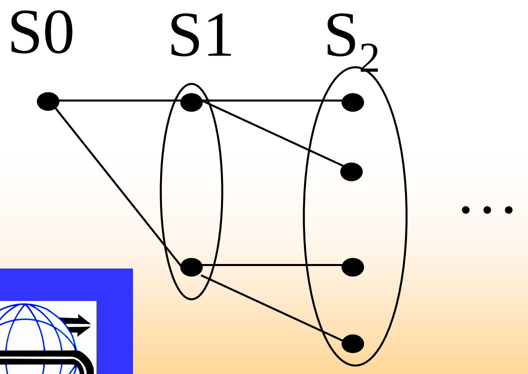
Instance representation

- SMPS format
- Algebraic modelling languages
- Internal representations
- XML format

Example (Birge)

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

$$\begin{aligned}
\text{s.t.}\quad & \sum_{i=1}^{I} x_{0i} && = B \\
& \sum_{i=1}^{I} \alpha_{0is} x_{0i} - \sum_{i=1}^{I} x_{1is} && = 0, s \in S_1 \\
& \sum_{i=1}^{I} \alpha_{t-1,i,s} x_{t-1,i,a(s)} - \sum_{i=1}^{I} x_{tis} && = 0, s \in S_t, t = 2,\mathrm{K}, T-1 \\
& \sum_{i=1}^{I} \alpha_{T-1,i,s} x_{T-1,i,a(s)} + u_s - w_s && = R, s \in S_T \\
& x_{tis}, u_s, w_s \geq 0
\end{aligned}$$



$I = 2$, $T = 3$, $B = 55$, $R = 80$,

α_{t1} = {1.25, 1.06},

α_{t2} = {1.14, 1.12}

DALHOUSIE UNIVERSITY *Inspiring Minds*

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           Budget0    1.0
    X01           Budget1    1.25
    ...
RHS
    rhs1          Budget0    55.
    rhs1          Budget3    80.
ENDATA
```

Stoch file

```
BLOCKS        DISCRETE
 BL Block1                0.5
    X01       Budget1     1.25
    X02       Budget1     1.14
 BL Block1                0.5
    X01       Budget1     1.06
    X02       Budget1     1.12
 BL Block2                0.5
    X11       Budget2     1.25
    X12       Budget2     1.14
    ...
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

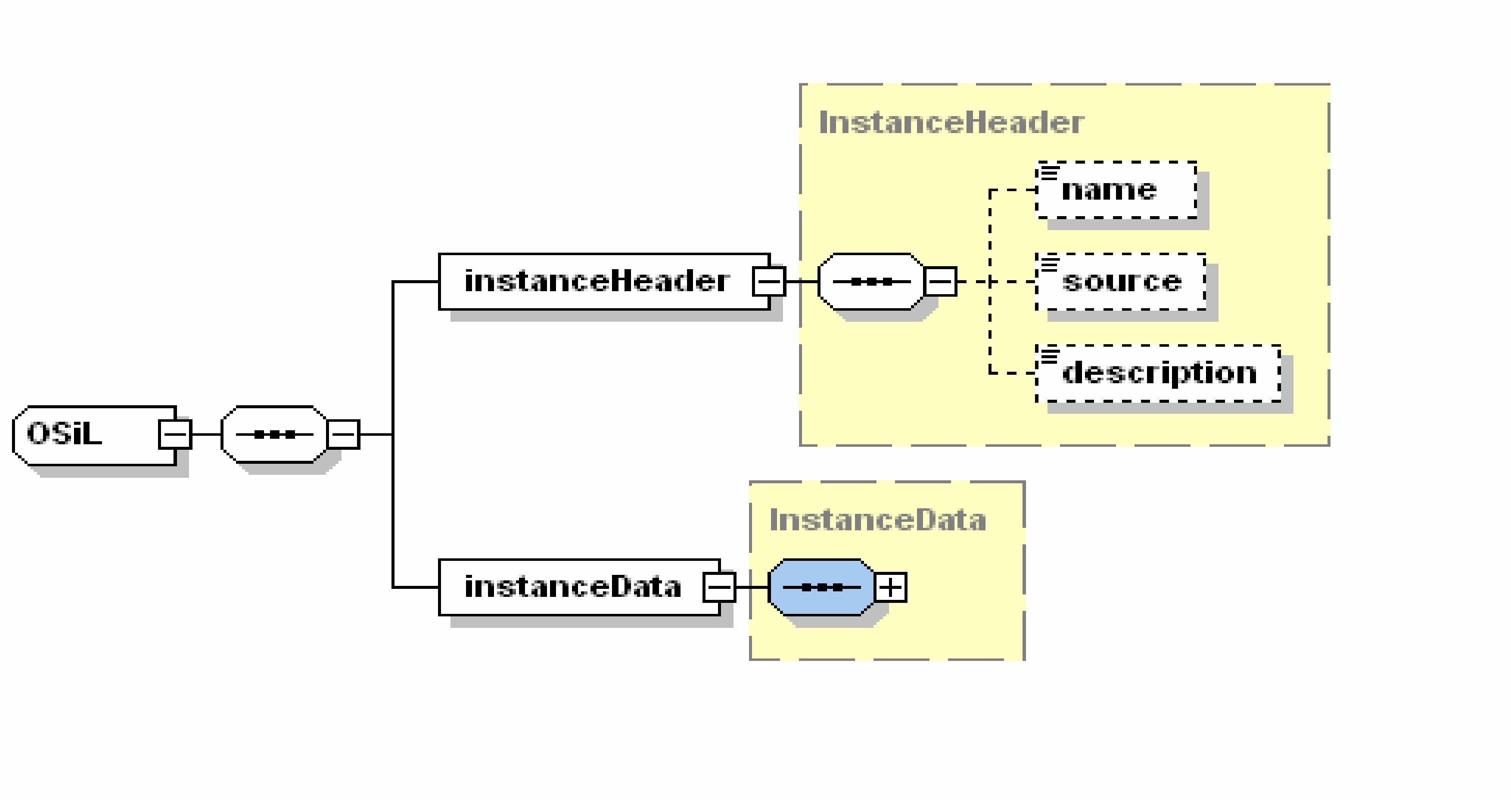
Storage requirements

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

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

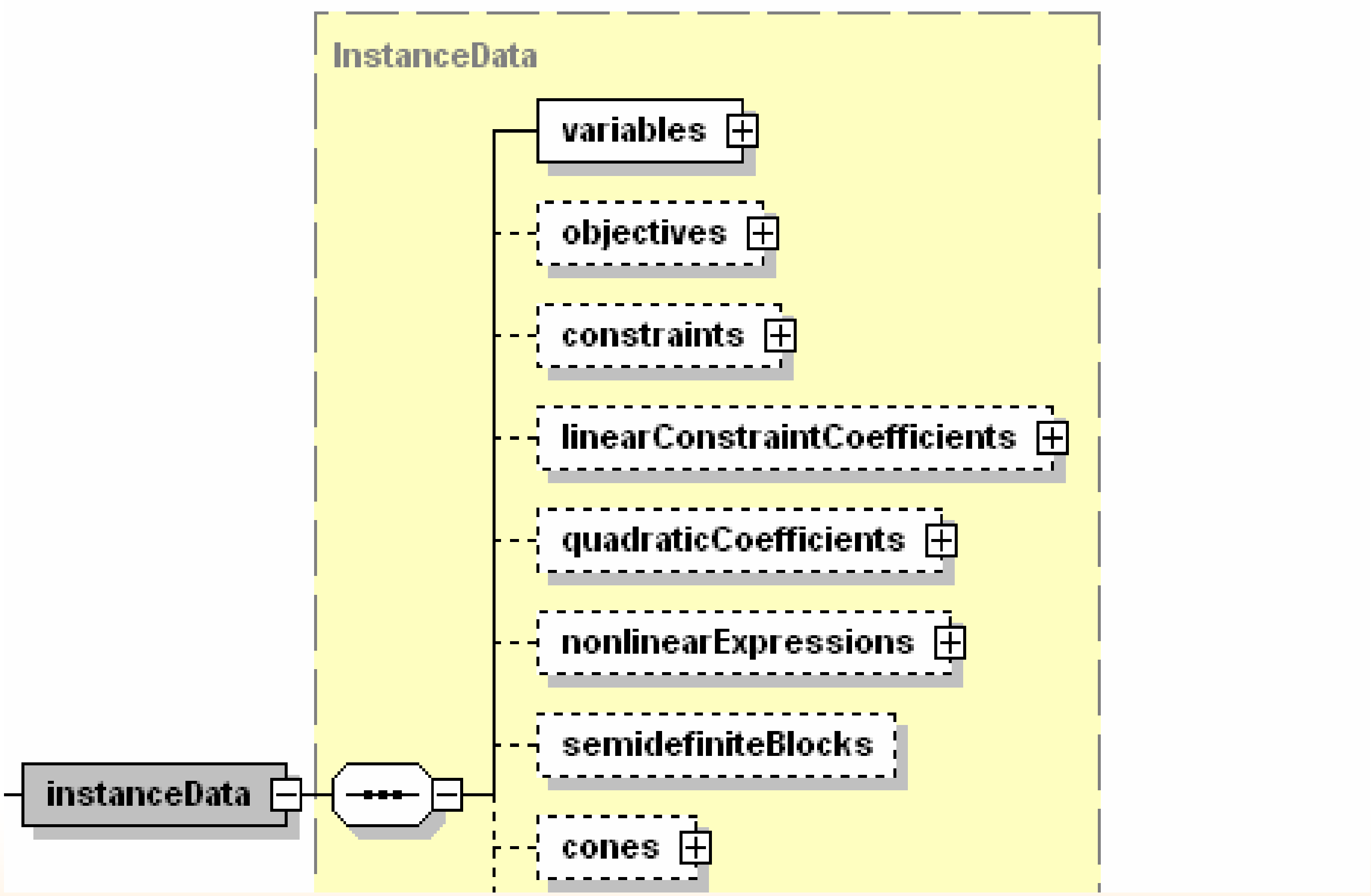
OSiL – Header information



OSiL – Header information

```
<?xmlversion="1.0"encoding="UTF8"?>
  <OSiL xmlns="os.optimizationservices.org“
    xmlns:xsi=http://www.w3.org/2001/XMLSchemainstance
    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>
```

OSiL – Deterministic information



. . .

OSiL – Program data – Constraints and variables

```
<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>
```

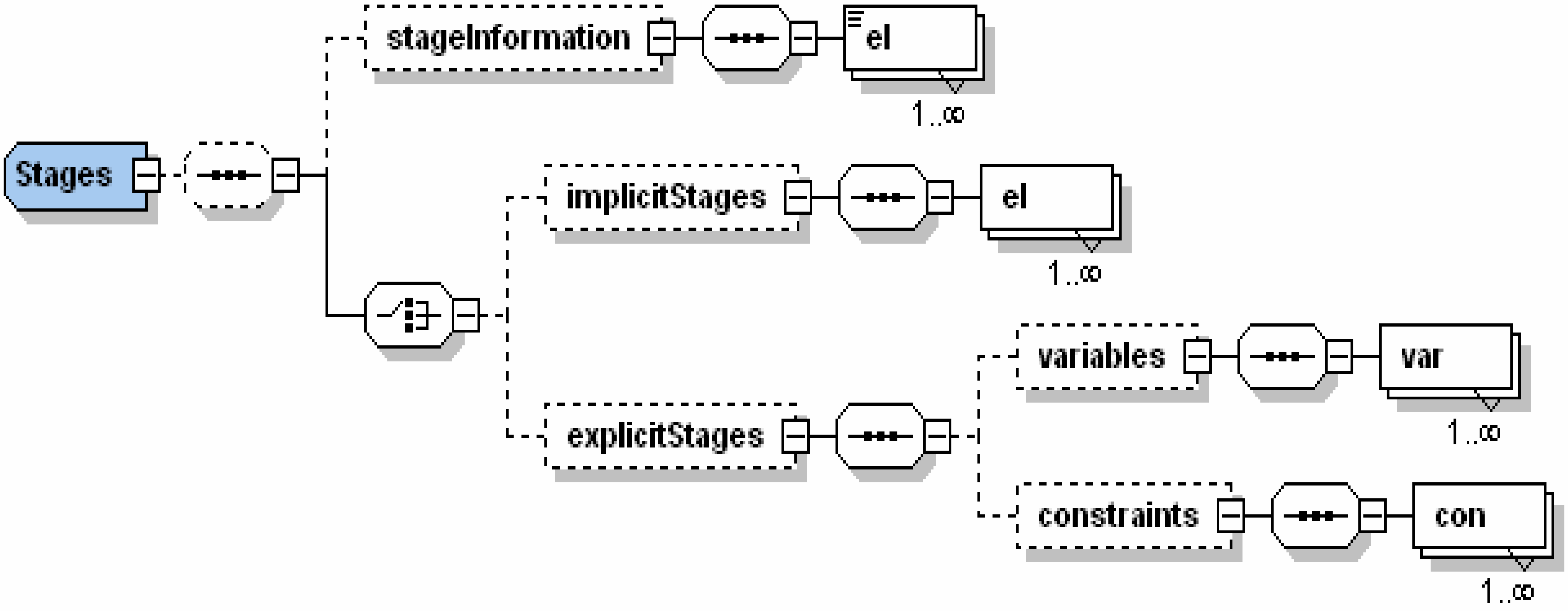
OSiL –
Core matrix (sparse matrix form)

```
<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>0</el>
  <el>1</el>
  <el>1</el>
  <el>2</el>
  <el>1</el>
  <el>2</el>
  <el>2</el>
  <el>3</el>
  <el>2</el>
  <el>3</el>
  <el>3</el>
  <el>3</el>
 </rowIdx>
```

```
<value>
  <el>1</el>
  <el>-1.25</el>
  <el>1</el>
  <el>-1.14</el>
  <el>1</el>
  <el>-1.25</el>
  <el>1</el>
  <el>-1.14</el>
  <el>1</el>
  <el>-1.25</el>
  <el>1</el>
  <el>-1.14</el>
  <el>1</el>
  <el>1</el>
 </value>
```

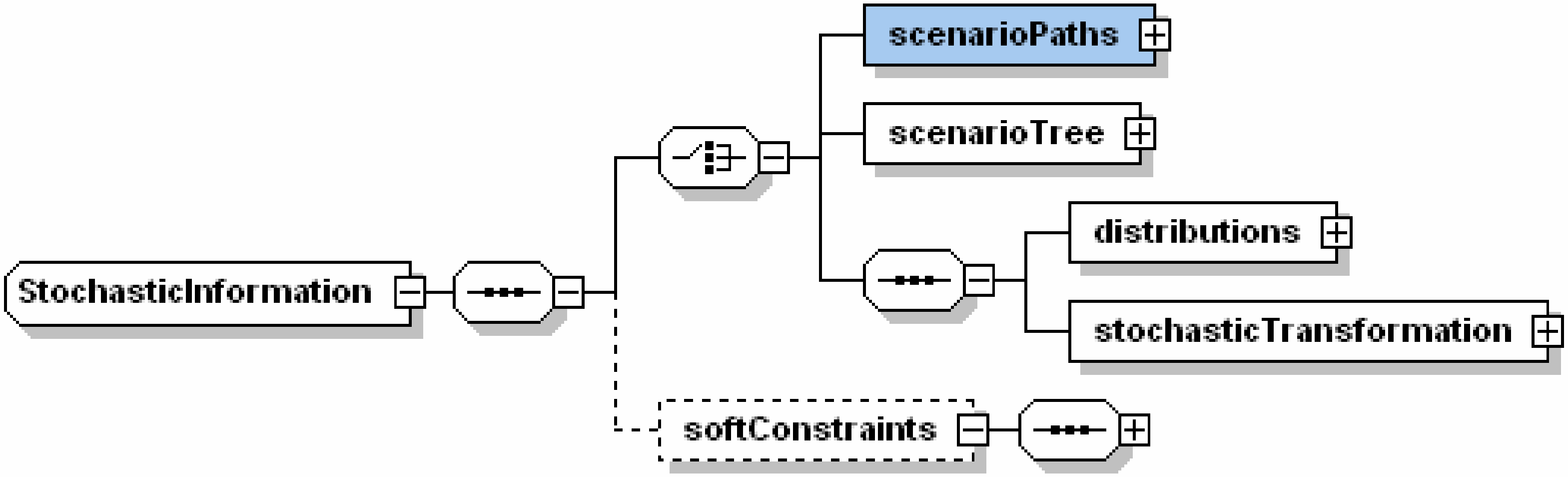
Dynamic structure



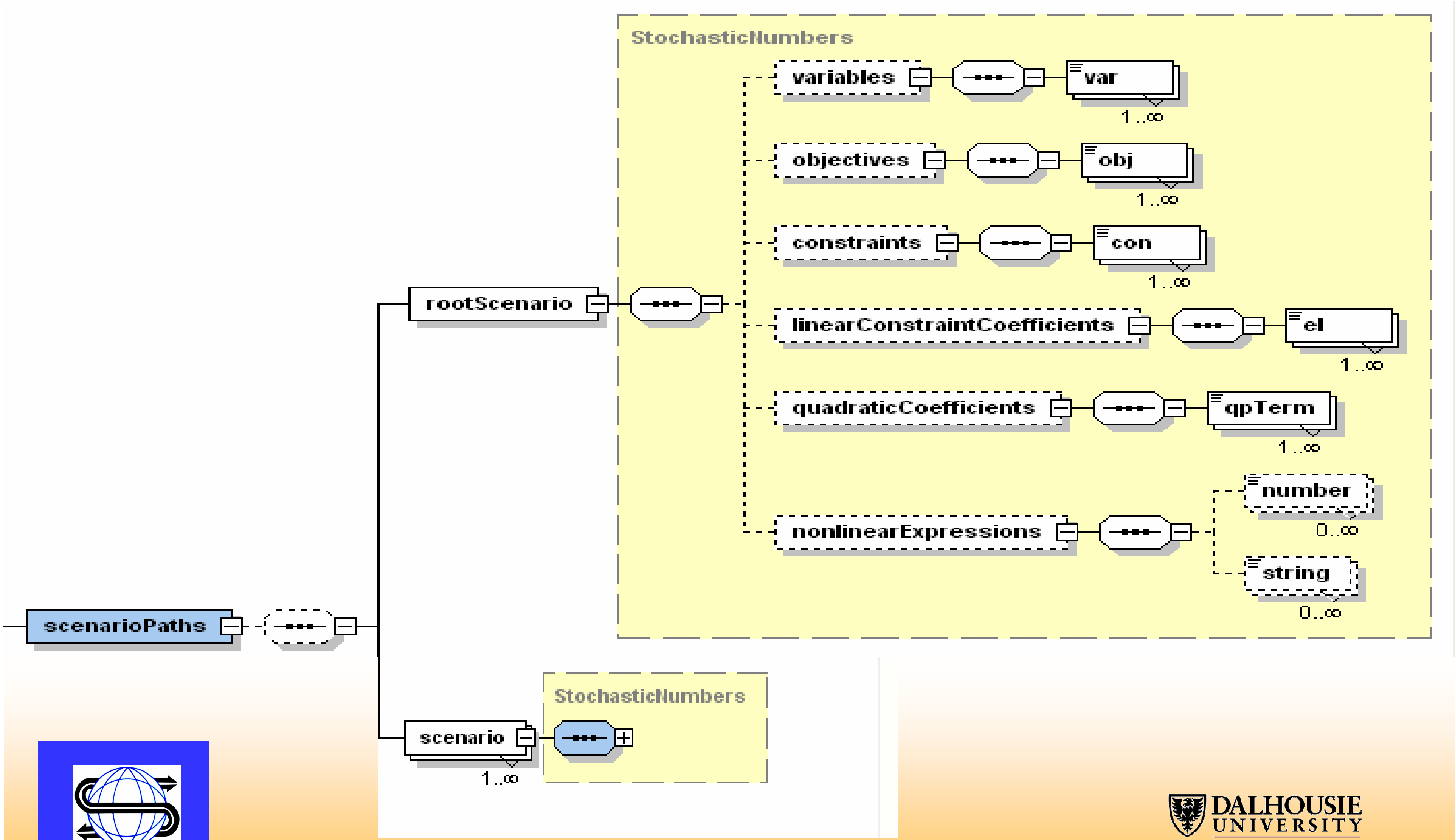
OSiL – dynamic information

```
<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



Scenario trees



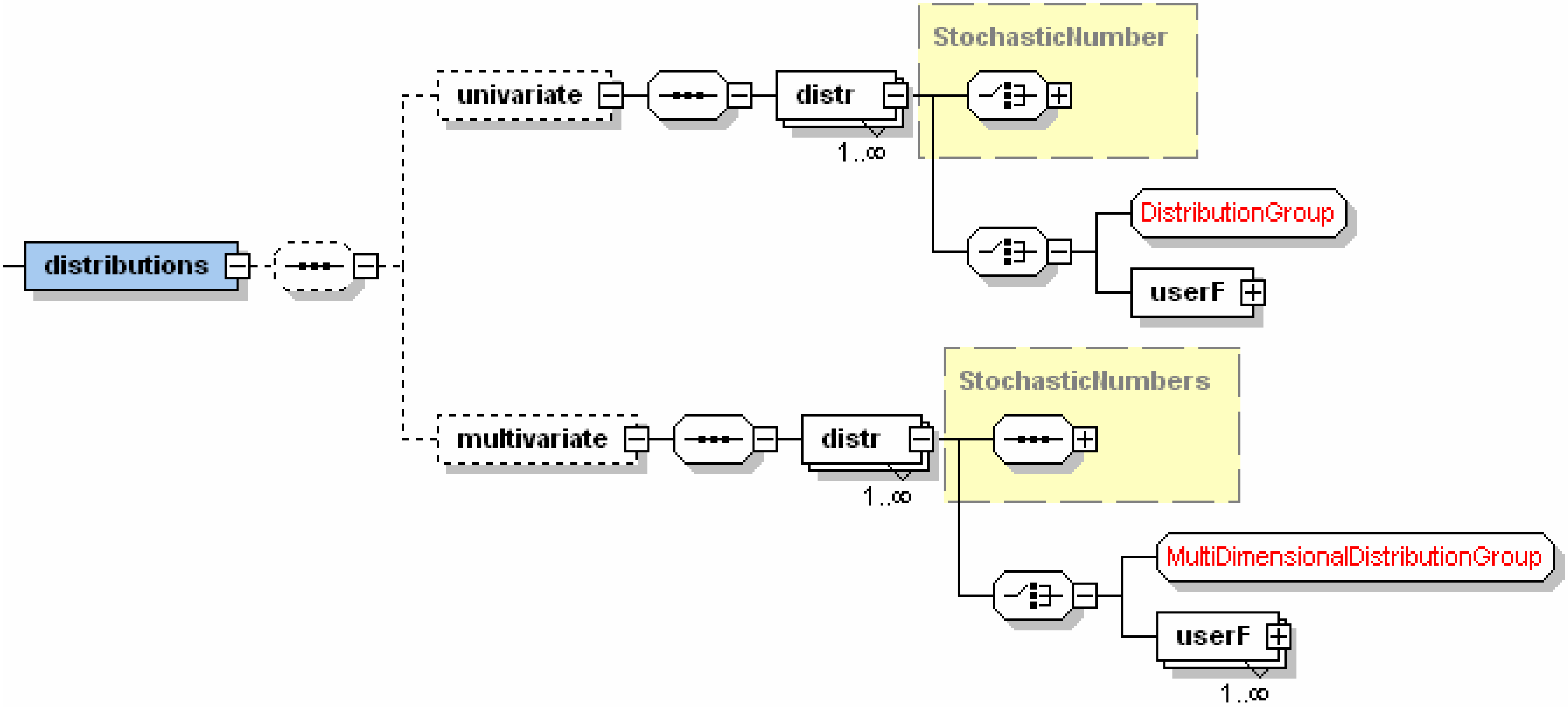
DALHOUSIE
UNIVERSITY
Inspiring Minds

OSiL – Stochastic information

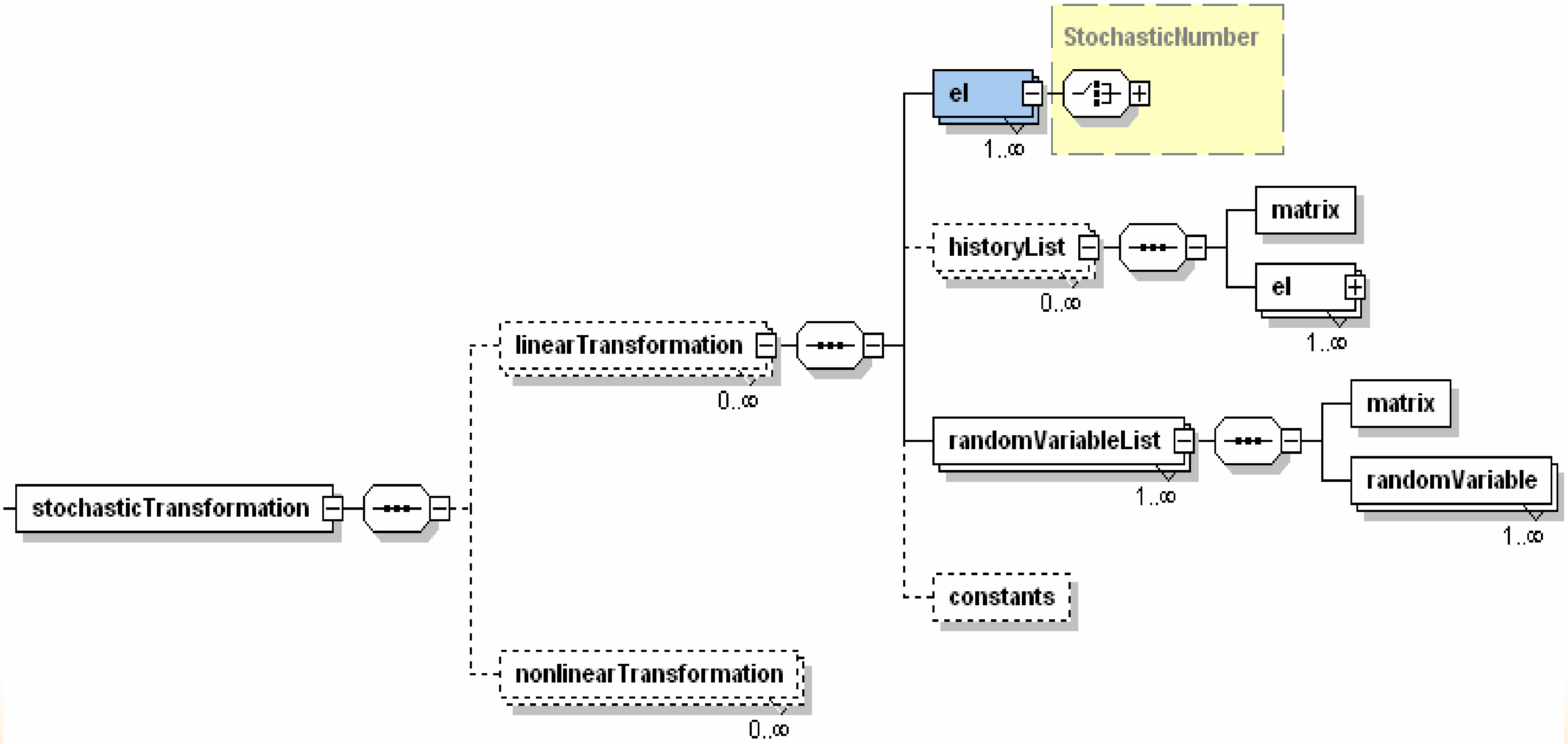
```
<stochastic>
 <explicitScenario>
  <scenarioTree>
   <sNode prob="1" base="coreProgram">
    <sNode prob="0.5" base="coreProgram">
     <sNode prob="0.5" base="coreProgram">
      <sNode prob="0.5" base="coreProgram"/>
      <sNode prob="0.5" base="firstSibling">
       <changes>
        <el rowIdx=“3"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

DALHOUSIE UNIVERSITY
Inspiring Minds

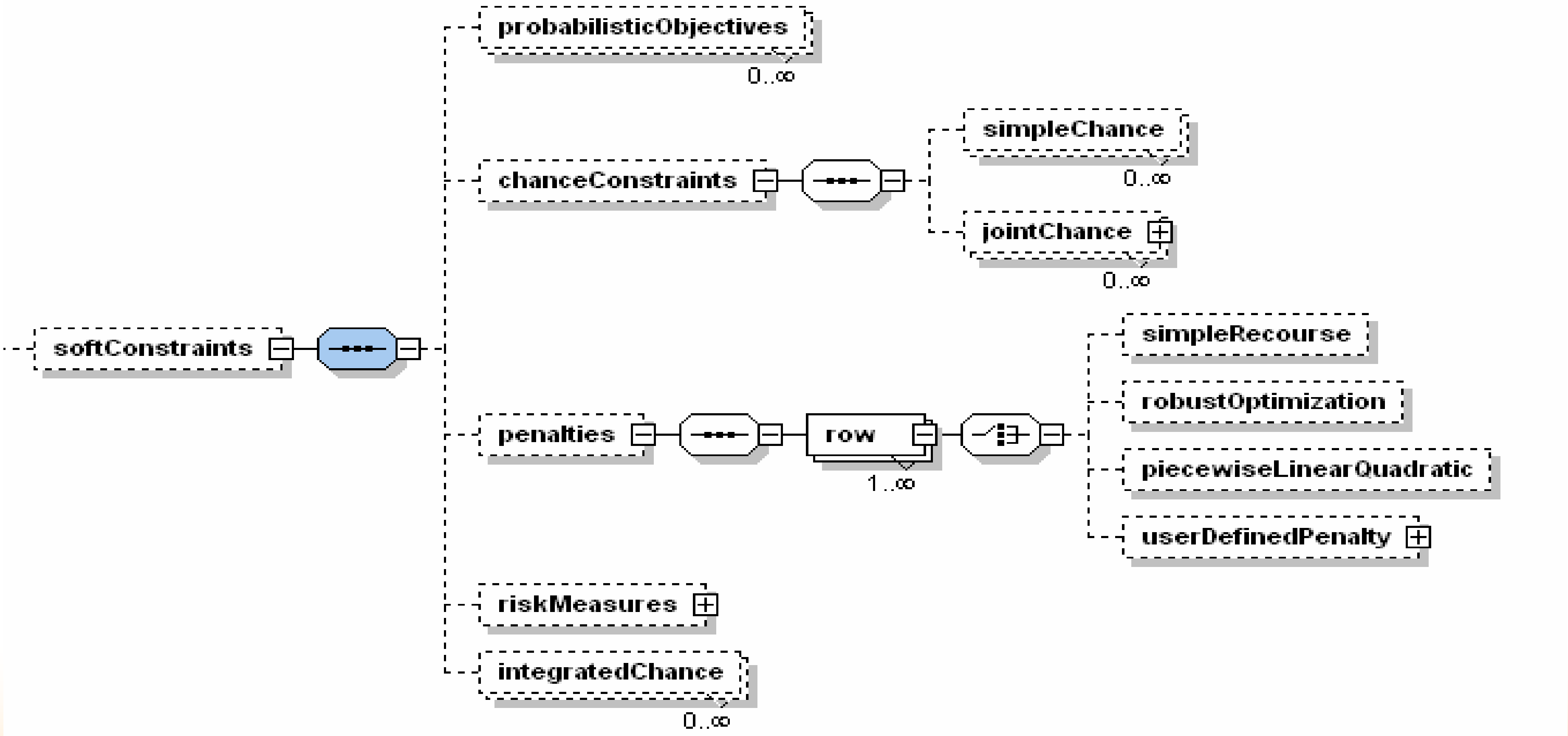
Distributions



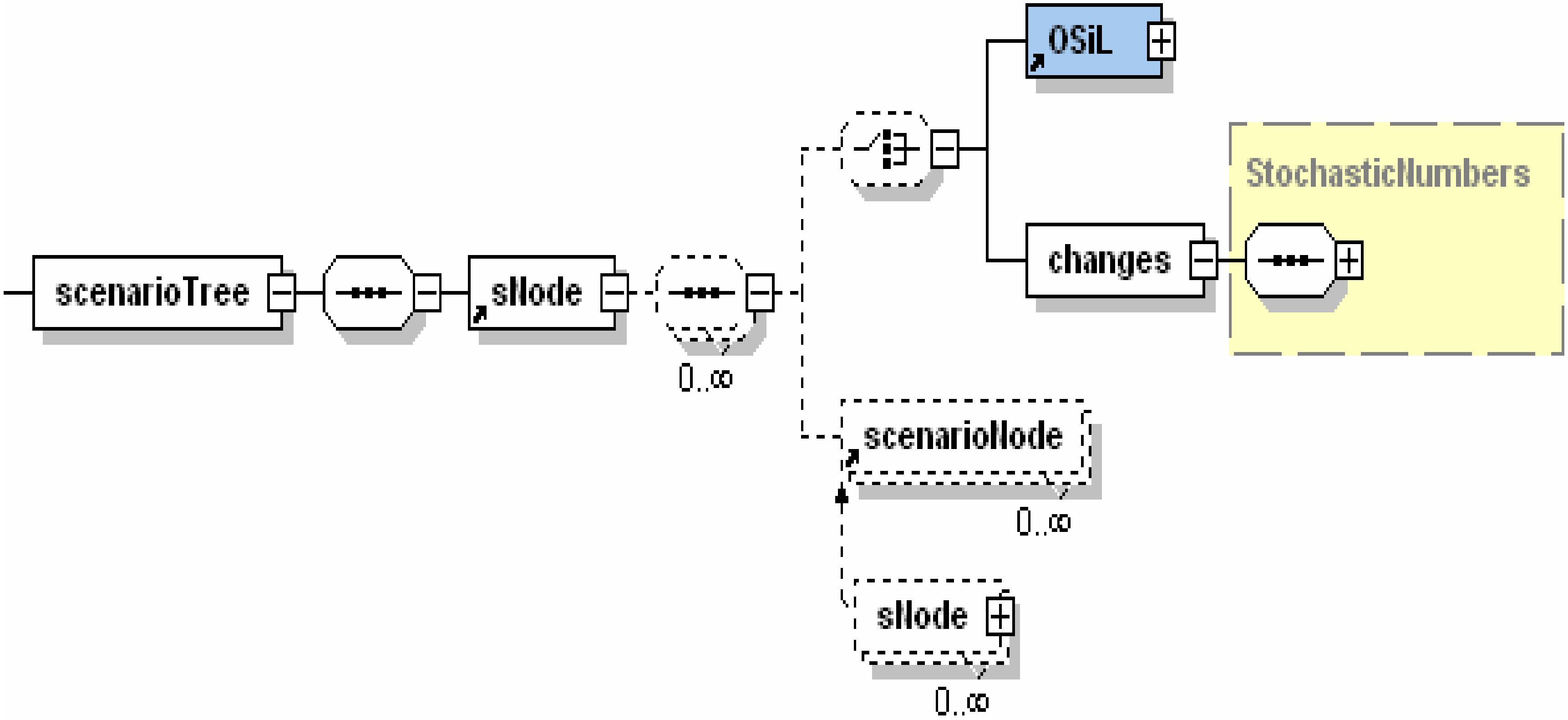
Transformations



Penalties and probabilistic constraints



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 – $(x_0 - x_1^2)^2 + (1 - x_0)^2$

```
<plus>
  <power>
    <minus>
      <var coef="1.0" idx="0"/>
      <power>
        <var coef="1.0" idx="1"/>
        <number value="2"/>
      </power>
    </minus>
    <number value="2"/>
  </power>
  <power>
    <minus>
      <number value="1"/>
      <var coef="1.0" idx="1"/>
    </minus>
    <number value="2"/>
  </power>
</plus>
```

DALHOUSIE UNIVERSITY
Inspiring Minds

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.optimizationservices.org

myweb.dal.ca/gassmann