Stochastic Programming using GAMS

Steven Dirkse

GAMS Development Corporation

Washington DC

GAMS

About GAMS

- What is GAMS?
- Why are we doing SP?
  - Existing client applications
  - Potential new applications
- What do we hope to achieve?
  - Efficiency in modeling
  - Efficiency in “problem solution”

Current Collaborations

- DECIS
  - DECompositon (Benders, multistage)
  - Importance Sampling
- IBM Stochastic Solutions
  - General purpose SP tool
  - Contains nested Benders solver
- Structure Exploiting Tool

DECIS: Problem Formulation

- Initial time-staged LP model
  - SMPS: core & time files
  - GAMS: Standard LP
- Augmentation to describe stochastic nature
  - SMPS: stoch file
  - GAMS: auxiliary file
  - Complete Cartesian product

DECIS: Problem Solution

- Eval, universe, expected value
- Monte Carlo pre-sampling, regularization
- Monte Carlo Sampling
  - Estimates recourse costs, cuts, bounds
  - Student-*t* test determines convergence
  - Importance Sampling, control variates modes

SPOSL: Problem Formulation

- Initial time-staged *core* model
- Event tree to represent stochastic structure
  - nodes: system state at each stage
  - directed arcs: movement to subsequent stage
  - each *scenario* a path (leaf --> root)
  - scenarios specified as branches off existing tree
- Models specified as node sets

GAMS/SPOSL Formulation

- Initial time-staged formulation
- Augmentation to allow uncertainty
  - time T --> time-node pair (T,N)
  - ancestor relationship required
- Generating the uncertainty structure
  - done via dynamic sets
  - multiple trees for one model

SPOSL: Problem Solution

- Extract node-arc structure from DE
  - Full DE model formulated in GAMS
  - Sparse scenario “deltas” passed to SPOSL
- All nodes included in the solution
- Solution via nested Benders code
- Parallel implementation on SP2

Structure Exploiting Tool

- Model structure extracted with SET
- Interior-point decomposition solver
- Implementation on a cluster of Linux PCs
- Computational results:
  - 1,111,112 rows, 2,555,556 vars
  - < 3 hours

The ALM Project

- Ongoing World Bank project
- Strategic Asset Liability Management
  - Addresses fundamental problem
  - Uncertainty in interest & exchange rates, prices
- Lead to development of RAMS
  - Risk Analysis & Management System

Problem Formulation

- Initial time-staged formulation
- Augmentation to allow uncertainty
  - time T --> time-node pair (T,N)
  - ancestor relationship required
- Generating the uncertainty structure
  - accuracy / computability tradeoff
  - requires the use of NLP (least-squares)

Risk Minimization

- Many definitions of risk considered
- PDF of portfolio estimated
  - Measures value likelihood at some fixed time
  - Done via NLP
- Is the PDF acceptable?
  - Use a different risk function
  - Add more scenario

Problem Solution & Reporting

- Solver depends on formulation used
  - linear --> SP/OSL
  - quadratic --> SP/OSL
  - nonlinear --> CONOPT, MINOS
- Reporting
  - estimation of density function
  - output via MATLAB linkage

Conclusions

- Opportunities to employ SP abound
  - Potential user base is quite large
  - Computational power is there
  - Application to other model types
- Formulation is still a hurdle
  - Better educated modelers
  - Better integration of existing tools
  - New developments?

Time-Staged Model

```
sets
   C          'commodities'              / rice, corn /,
   T          'time'                     / spring, fall, winter /;

parameters
   price(C,T),
   demand;

positive variables
   stock(C,T),
   x(C,T)                                'purchase quantity';

variable
   z                                     'overall cost';

equations
   stockdef(C,T),
   demdef,
   zdef;

zdef..        z  =e=  sum {(C,T), price(C,T)*x(C,T)};

stockdef(C,T)..
              stock(C,T)  =e=  stock(C,T-1) + x(C,T);

demdef..
              sum {C, stock(C,'winter')}  =g=  demand;

model timeonly / zdef, stockdef, demdef /;

solve timeonly using lp minimizing z;
```

Augmented Model

```
sets
   N          'nodes'      / 0, 1 * 8 /,
   TN(T,N)  /              spring.0,
                           fall.1,
                           winter.2   /,
   ANC(N,N)                /               1.0,
                           2.1      / ;
alias(N,NN);

parameter
   price(C,T,N),
   demand(N)
   prob(N);

positive variables
   stock(C,T,N),
   x(C,T,N)                                'purchase quantity';

equations
   stockdef(C,T,N),
   demdef(T,N);

zdef..          z =e= sum {(C,TN(T,N)), price(C,T,N)*prob(N)*x(C,T,N)};

stockdef(C,TN(T,N))..
                stock(C,T,N) =e= sum {ANC(N,NN), stock(C,T-1,NN)} + x(C,T,N)};

demdef(TN(T,N))$[TLAST(T)]..
                sum {C, stock(C,T,N)} =g= demand(N);

model stoch / zdef, stockdef, demdef /;

solve stoch using lp minimizing z;
```

Node Generation

```
* cheap fall corn
price('corn','fall','1') = 0.2;
prob('1') = .1;
prob('2') = .1;
demand('2') = 1.3;

* expensive fall corn
TN('fall','3') = YES;
anc('3','0') = YES;
price('corn','fall','3') = 1.2;
prob('3') = .9;

TN('winter','4') = YES;
anc('4','3') = YES;
prob('4') = .36;
demand('4') = .9;

TN('winter','5') = YES;
anc('5','3') = YES;
prob('5') = .36;
demand('5') = 1.1;

TN('winter','6') = YES;
anc('6','3') = YES;
prob('6') = .18;
demand('6') = 1.3;
```