Stochastic Programming in GAMS

Lutz Westermann

lwestermann@gams.com

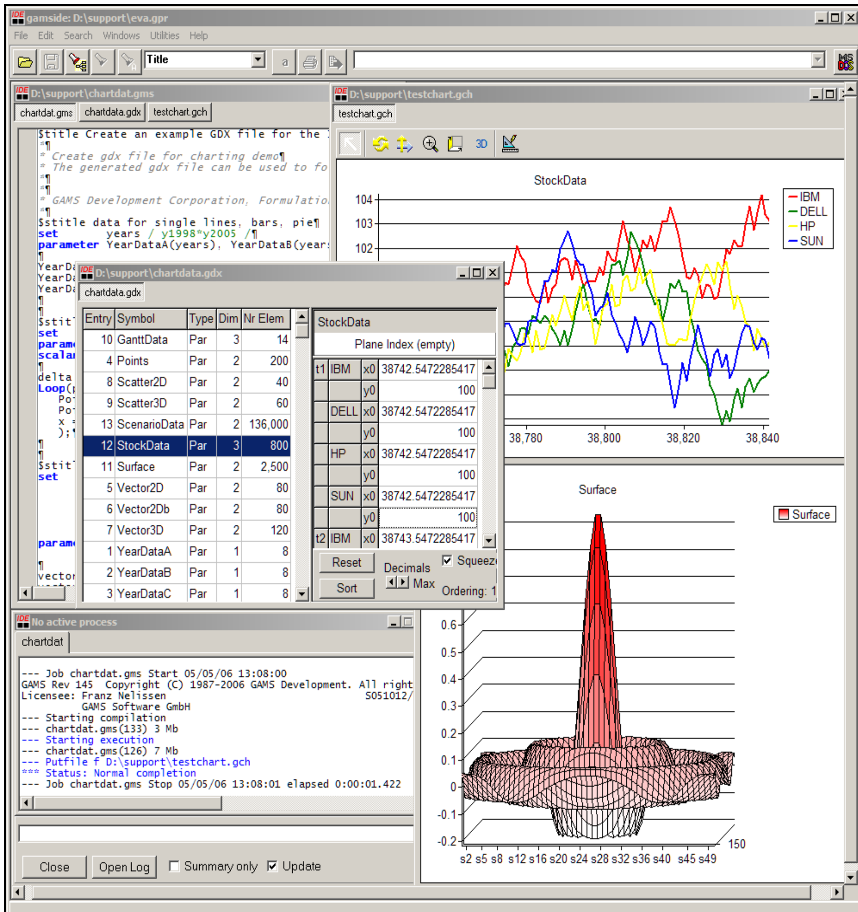
GAMS Software GmbH

GAMS Development Corporation

www.gams.com

INFORMATICS RISING

GAMS at a Glance



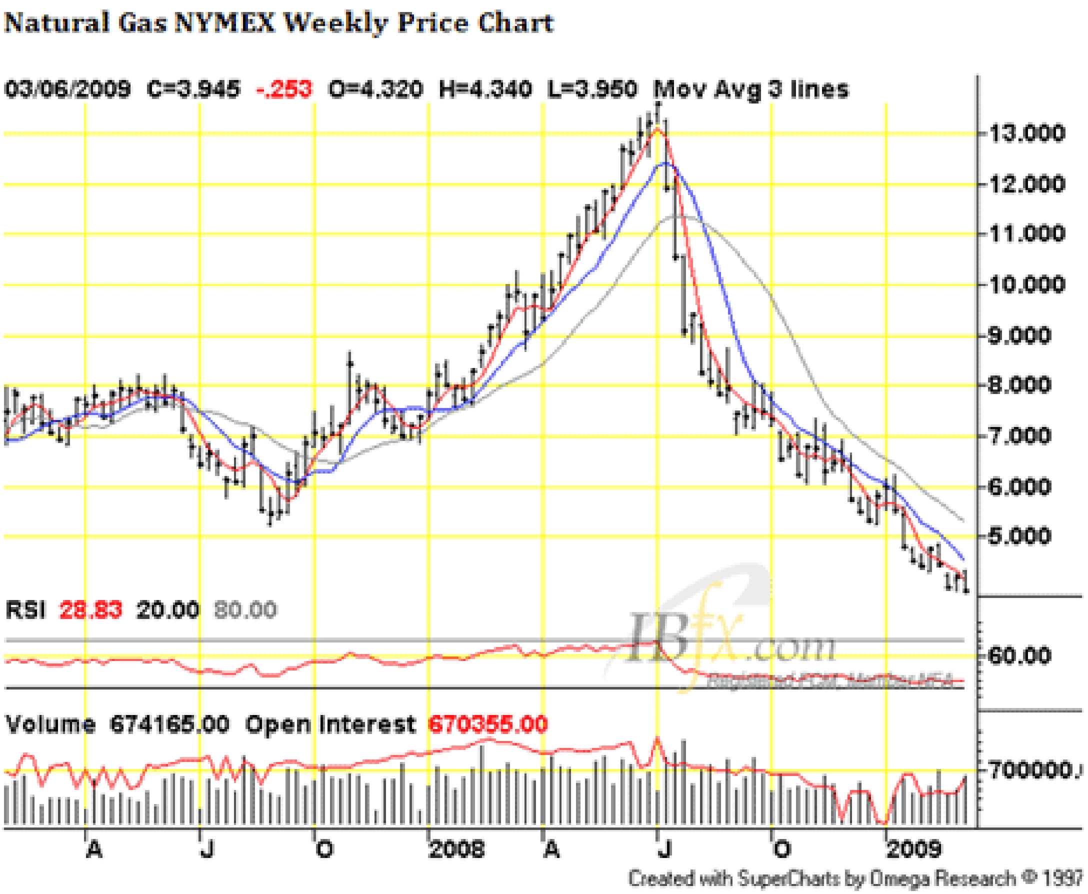
General **A**lgebraic **M**odeling **S**ystem

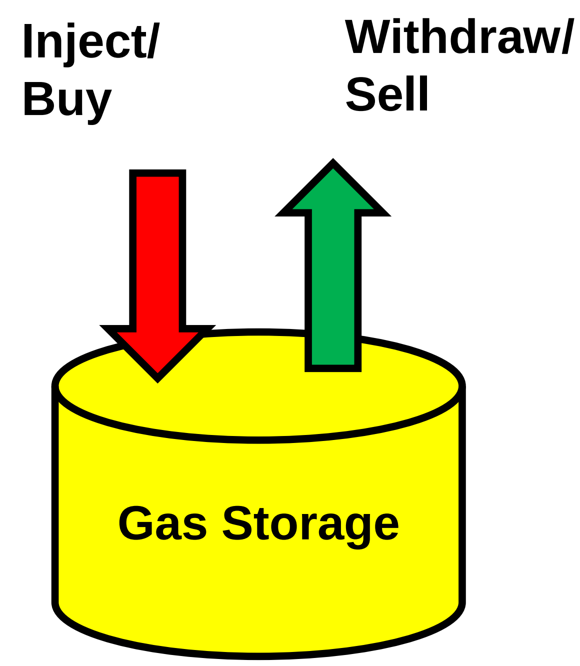
- Algebraic Modeling Language
- 30+ Integrated Solvers
- 10+ Supported MP classes
- 10+ Supported Platforms
- Connectivity- & Productivity Tools
  - IDE
  - Model Libraries
  - GDX, Interfaces & Tools
  - Grid Computing
  - Benchmarking
  - Compression & Encryption
  - Deployment System
  - APIs (C, Fortran, Java, .Net ...)
  - …

2010

Example Model: Gas Price Model

2010





n-Stage Stochastic Programs

2010

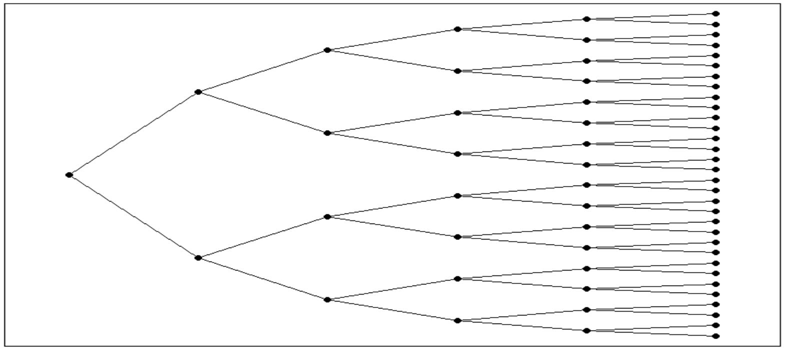
- Construct Scenario Tree:
  - Start with today’s price and use a (discrete) distribution
  - Realizations: up, down
- Stochastic Linear Program (block structure)
  - Nested Bender’s Decomposition (OSLSE, FortSP, AIMMS)
  - In practice Deterministic Equivalent with Barrier method

$$Z_{HN} = \min_{x_1} \left\{ c_1 x_1 + E_{\xi_2}\left[\min_{x_2} c_2 x_2 + E_{\xi_3|\xi_2}\left[\min_{x_3} c_3 x_3 + \ldots + E_{\xi_T|\xi_{T-1}|\ldots|\xi_2} \min_{x_T} c_T x_T \right]\right]\right\}$$

subject to:

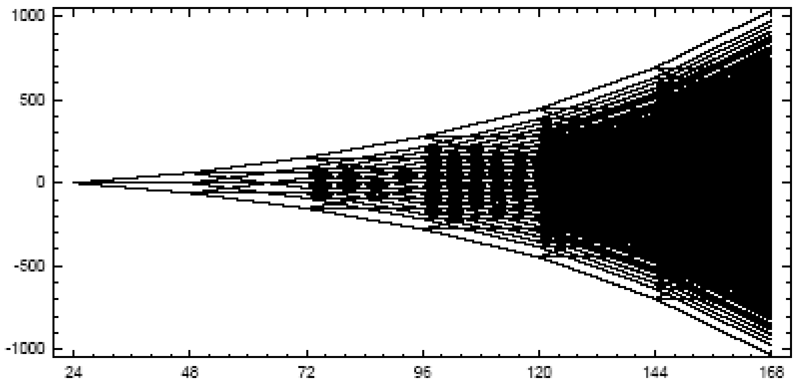
$$\begin{array}{lllll} A_{11}x_1 & & & & = b_1 \\ A_{21}x_1 + & A_{22}x_2 & & & = b_2 \\ A_{31}x_1 + & A_{32}x_2 + & A_{33}x_3 & & = b_3 \\ \vdots & & \ddots & & \vdots \\ A_{T1}x_1 + & A_{T2}x_2 + & A_{T3}x_3 + & \ldots & + A_{TT}x_T = b_T \end{array}$$

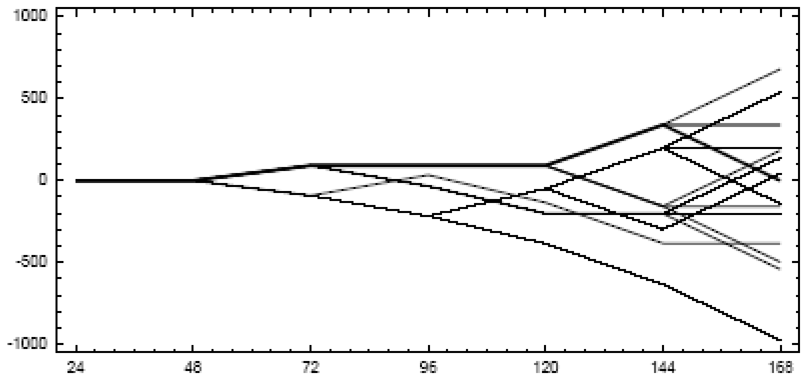
$$\ell_t \le x_t \le u_t;$$



ScenRed (Römisch et. al., HU Berlin)

2010



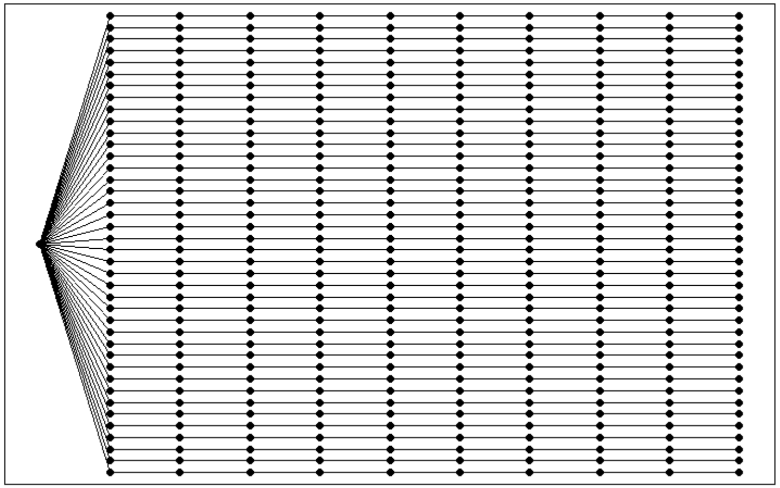


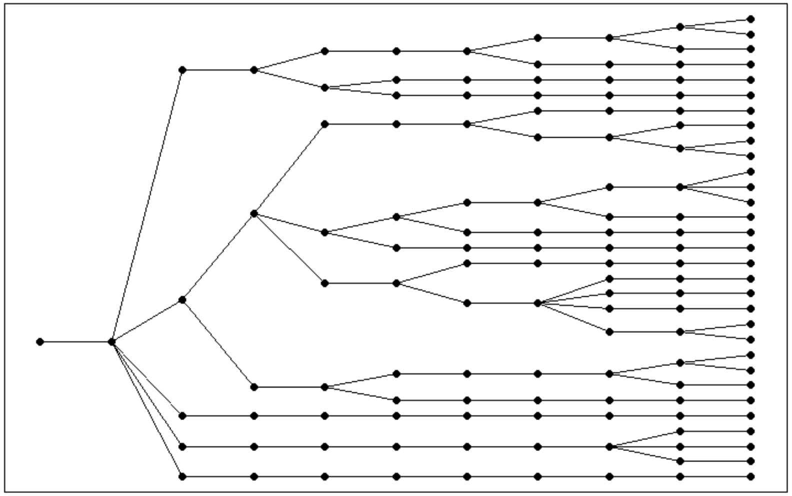
- Find good approximation of original scenario tree of significant smaller size
- Available since 2002
- Integrated in GAMS system
- No extra cost

Tree Generation: ScenRed2

2010

- Construct a true scenario tree from independent scenarios:



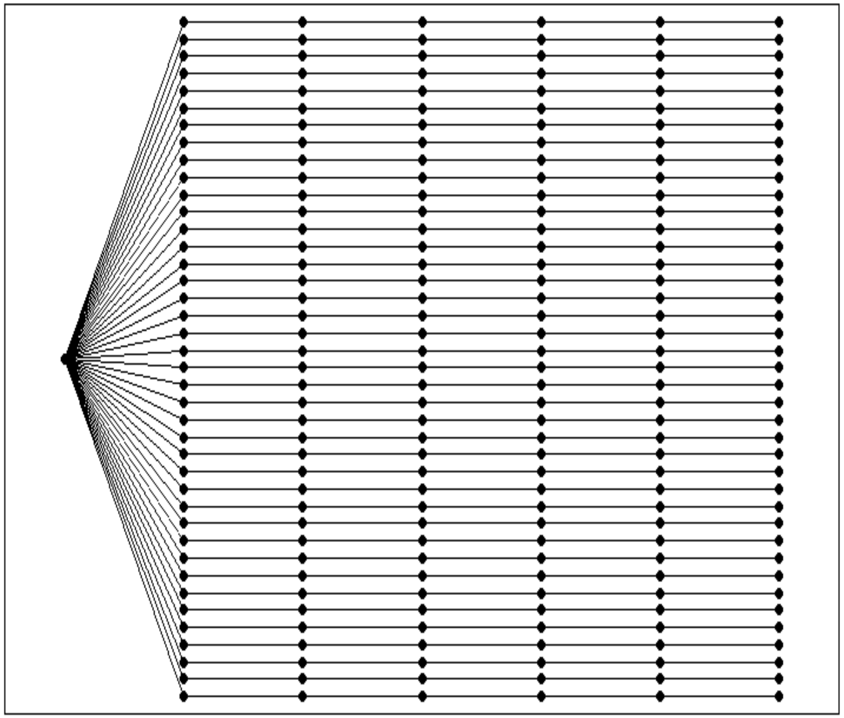


- Reconstruct underlying distribution from a set of scenarios

2-Stage Stochastic Programs

2010

- 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



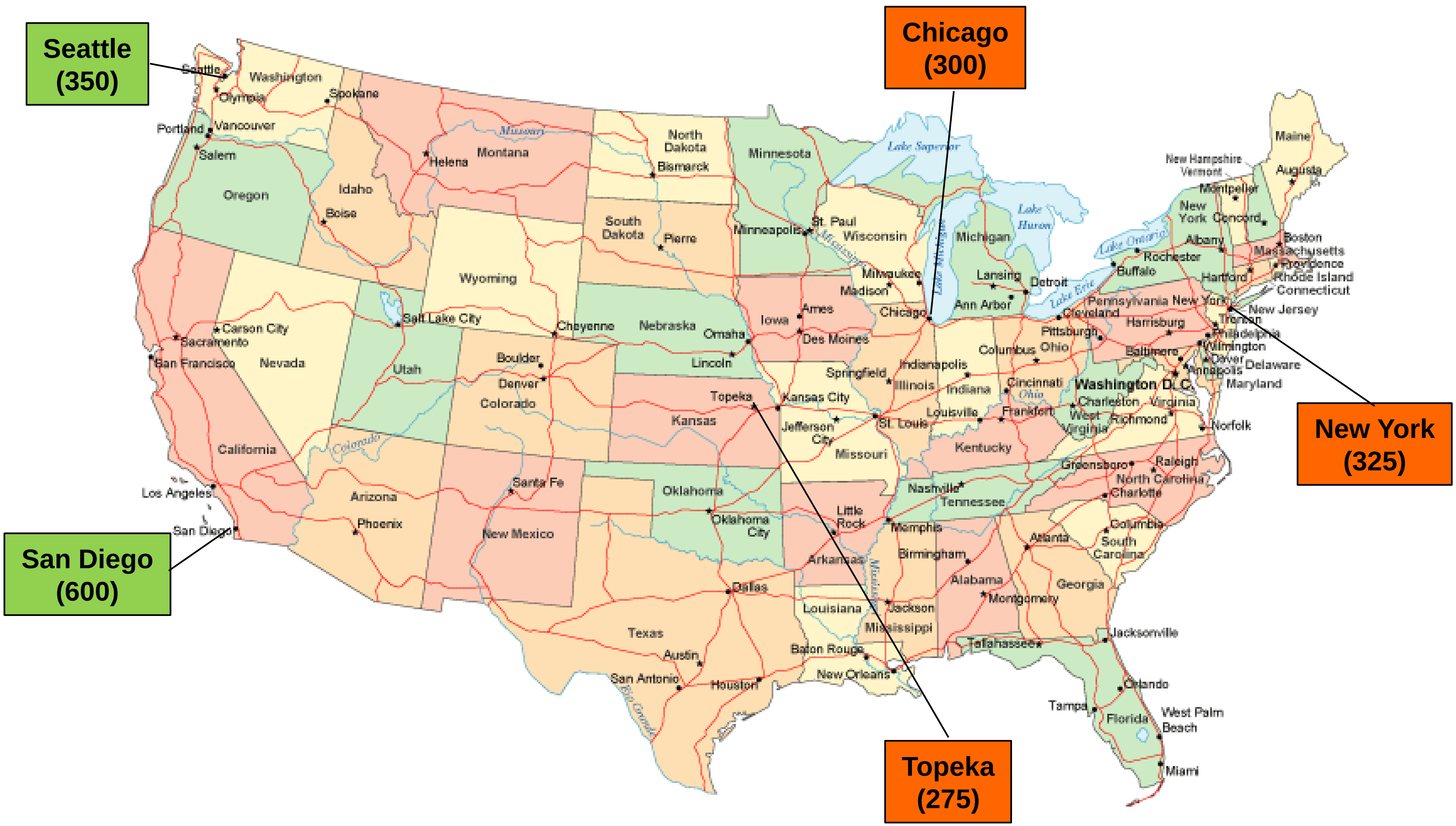
AML and Stochastic Programming (SP)

2010

- 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

A Transportation Model



Simple Example: Transportation Model

- Data:
  - Certain capacity at plants
```
a(i) / seattle  350, san-diego 600             /
```
  - Certain demand at markets
```
b(j) / new-york 325, chicago   300, topeka 275 /
```
  - Given transportation cost
```
c(i,j)          new-york      chicago      topeka
       seattle     0.225        0.153       0.162
     san-diego     0.225        0.162       0.126
```
  - Units can also be bought at markets directly for a fixed price
```
p    / 1 /
```

- Decisions:
  - How many units to ship: **X(i,j)**
  - How many units to buy: **U(j)**

… in order to minimize total cost

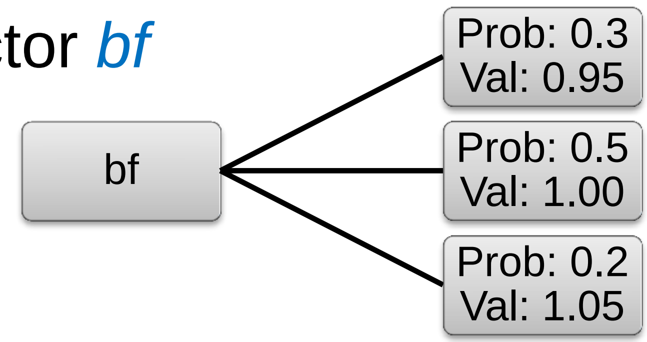
GAMS

Transportation Model – GAMS Formulation

```
*               Costs to mimimize
cost..          Z =e=    sum((i,j), c(i,j)*X(i,j))
                      +  sum( j,    p      *U(j));

*               Supply limitation
supply(i)..     sum(j, X(i,j))   =l=  a(i);

*               Demand requirement
demand(j)..     sum(i, X(i,j))   =g=  b(j) - U(j);

Model transport / all /;

Solve transport using lp minimizing Z;
```

Transportation Problem – Add Uncertainty

- Uncertain demand factor *bf*



- Decisions to make:
  - How many units should he shipped “here and now” (without knowing the outcome of the uncertain demand)?
    → *First-stage decision*
  - How many units need to be bought 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

Stochastic NB Problem – GAMS Extension

```
* Add demand factor bf
demand(j).. sum(i, X(i,j)) =g= bf * b(j)-U(j);

...

* Make bf uncertain
randvar bf discrete 0.3 0.95
                    0.5 1.00
                    0.2 1.05

* Define non-default stages
stage 2 bf u demand
```

New GAMS (EMP) Keywords

Excursus: EMP, what?

With new modeling and solution concepts do not:

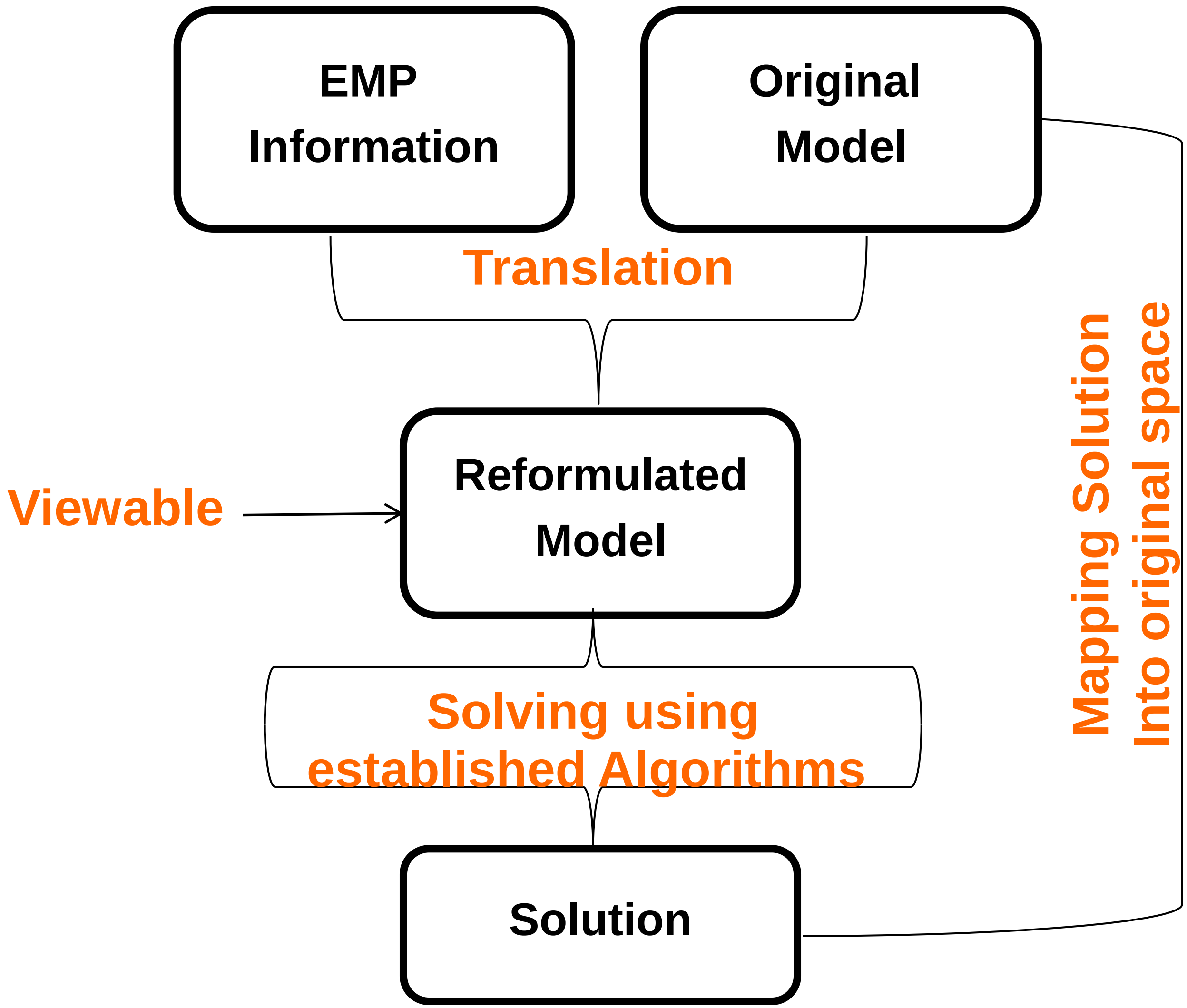
- overload existing GAMS notation right away !
- attempt to build new solvers right away !

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

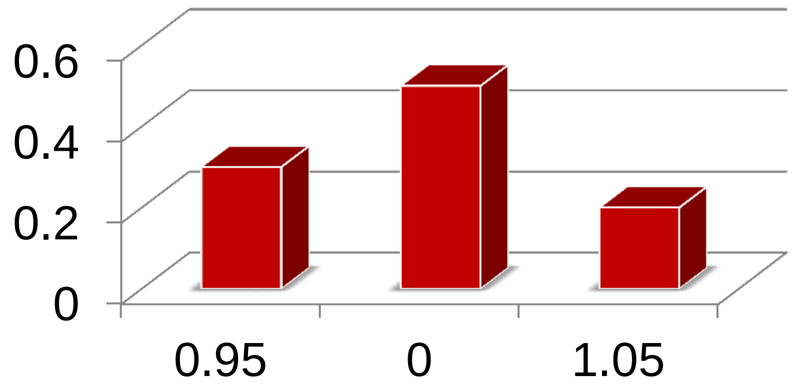
→ **Extended Mathematical Programming (EMP)**

JAMS: a GAMS EMP Solver

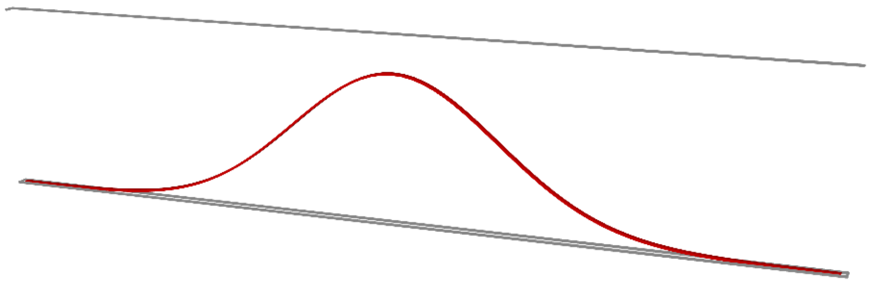


Random Variables

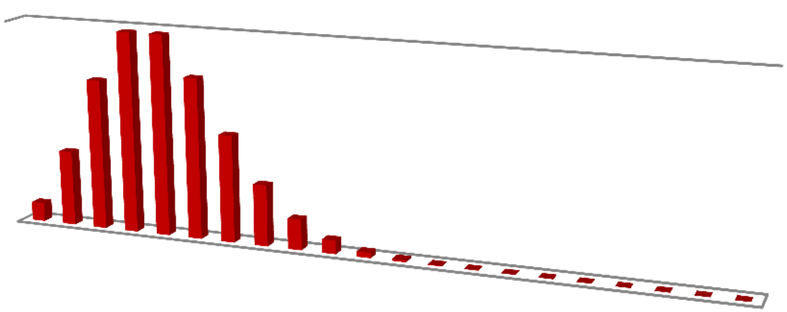
Discrete Distribution



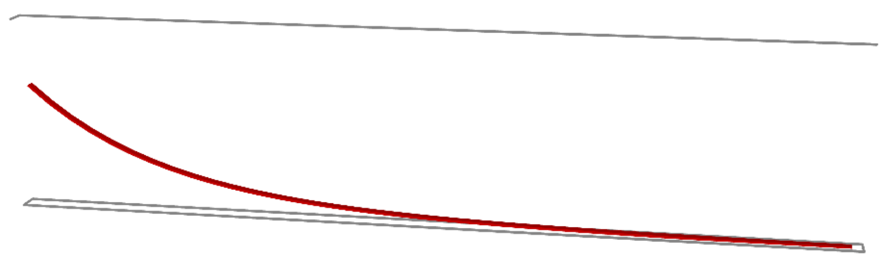
Normal Distribution



Poisson Distribution



Exponential Distribution



Random Variables (RV) [`randVar`]

- Defines both discrete and parametric random variables:

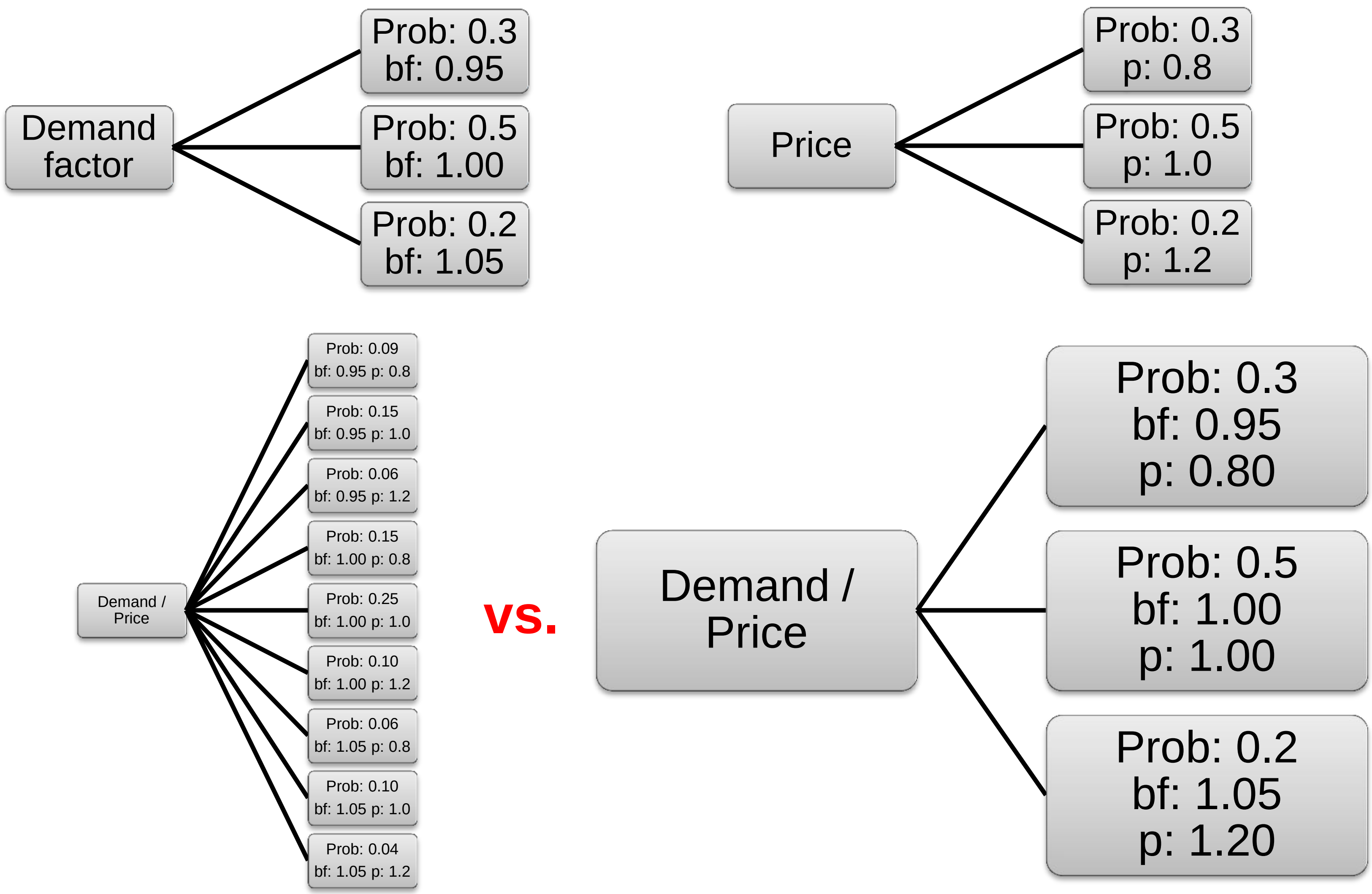
```
randVar rv discrete prob val {prob val}
```

- The distribution of discrete random variables is defined by pairs of the probability `prob` of an outcome and the corresponding realization `val`

```
randVar rv distr par {par}
```

- The name of the parametric distribution is defined by `distr`, `par` defines a parameter of the distribution

Joint Random Variables



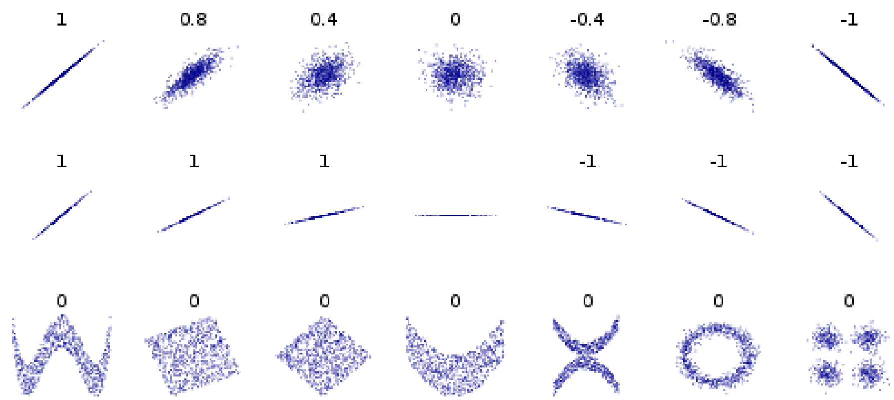
Joint RVs [jRandVar]

- Defines discrete random variables and their joint distribution:

```
jRandVar rv rv {rv} prob val val {val}
                   {prob val val {val}}
```

- At least two discrete random variables `rv` are defined and the outcome of those is coupled
- The probability of the outcomes is defined by `prob` and the corresponding realization for each random variable by `val`

Correlation between RVs [correlation]



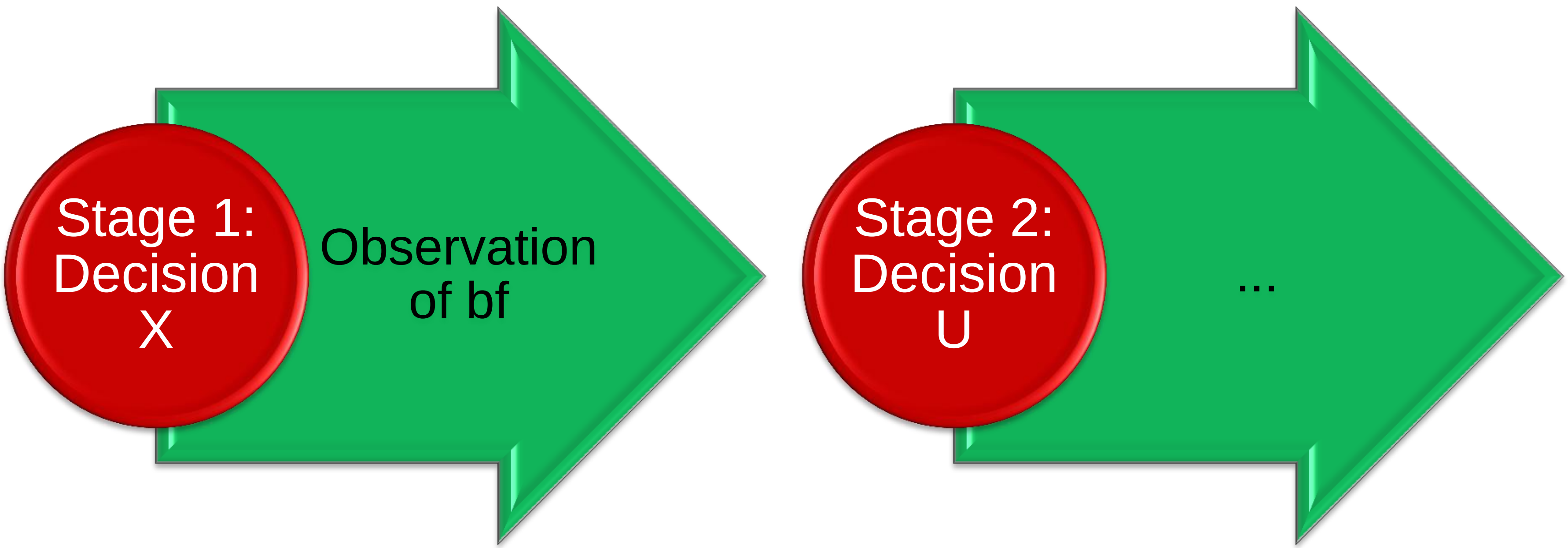
Source: Wikipedia

- Defines a correlation between a pair of random variables:

```
correlation rv rv val
```

- `rv` is a random variable which needs to be specified using the `randvar` keyword and `val` defines the desired correlation (-1 ≤ `val` ≤ 1)

Stages



Stages [stage]

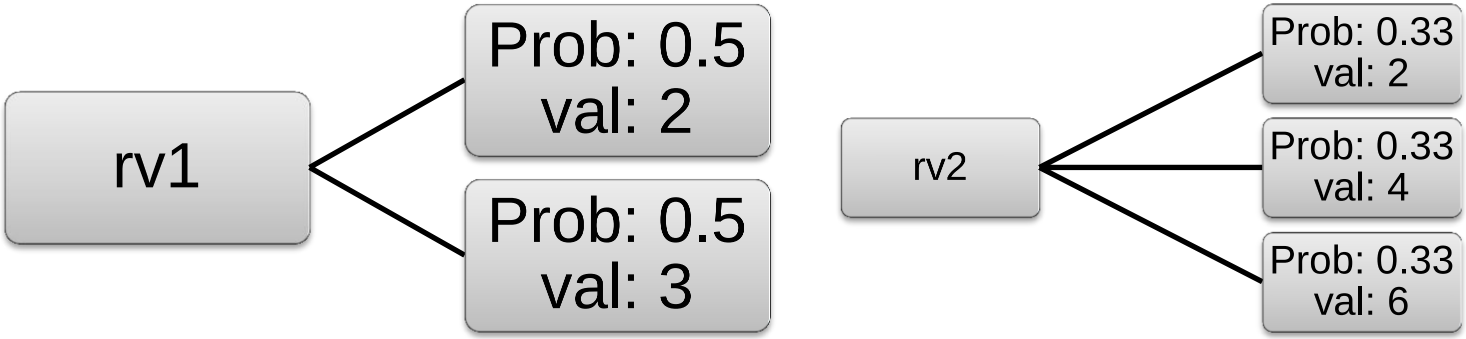
- Defines the stage of random variables (`rv`), equations (`equ`) and variables (`var`):

```
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..   rv1*X1 + 2*X2 =g=  5;
E2..   rv2*X1 + 6*X2 =g= 10;
Model sc / all /;
solve sc min z use lp;
```




```
chance E1 0.5
chance E2 0.5
```

Chance Constraints

1 out of 2 must be true

[0.5 ≥ 0.5]

```
• 2*X1 + 2*X2 =g= 5;
• 3*X1 + 2*X2 =g= 5;
```

2 out of 3 must be true

[0.66 ≥ 0.5]

```
• 2*X1 + 6*X2 =g= 10;
• 4*X1 + 6*X2 =g= 10;
• 6*X1 + 6*X2 =g= 10;
```

Just in case: X1 = 1 and X2 = 1 are optimal.

Chance Constraints [`chance`]

- Defines individual or joint chance constraints (CC):

```
chance equ {equ} [holds] minRatio [weight|varName]
```

- Individual CC: A single constraint `equ` has to hold for a certain ratio (0 ≤ `minRatio` ≤ 1) of the possible outcomes
- Joint CC: A set of constraints `equ` has to hold for a certain ratio (0 ≤ `minRatio` ≤ 1) of the possible outcomes
- If `weight` is defined, the violation of a CC gets penalized in the objective (weight violationRatio)
- If `varName` is defined the violation get multiplied by this existing variable

Expected Value [`ExpectedValue`]

- This is the default objective:

```
ExpectedValue [x EV_x]
```

- If only `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:

```
cVaR [x cVaR_x] theta
```

- If only `cVaR theta` is defined, the cVaR of the GAMS objective variable to the quantile level `theta` will be optimized

- If the variable pair `x cVaR_x` is defined, GAMS will replace its objective variable by `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:

```
...
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:

```
Set dict / scen .scenario.''
           bf   .randvar .s_bf
           ''   .opt     .srep
           x    .level   .s_x/;
```

Adding Uncertainty to Transport

```
...
demand(j)..   sum(i, x(i,j))  =g=  bf*b(j) - U(j) ;
...
file emp / '%emp.info%' /; put emp '* problem %gams.i%'/;
$onput
randvar bf discrete 0.3 0.95
                    0.5 1.00
                    0.2 1.05
stage 2 bf u demand
$offput
putclose emp;

Set scen         scenarios / s1*s3 /;
Parameter
    s_bf(scen,j)  demand factor realization by scenario
    s_x(scen,i,j) shipment per scenario
    s_s(scen) ;

Set dict / scen .scenario.''
           bf   .randvar .s_bf
           x    .level   .s_x /;

Solve transport using emp minimizing z scenario dict;
```

Summary

Available GAMS SP Solvers

	DE	DECIS	LINDO
chance	√		√
correlation			√
cVaR	√		
expectedValue	√		√
jrandVar	√	√	√
randVar (discrete)	√	√	√
randVar (parametric)			√

GAMS

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

<u>Europe</u>

GAMS Software GmbH
P.O. Box 40 59
50216 Frechen,
Germany

Phone: +49 221 949 9170
Fax: +49 221 949 9171

info@gams.de

<u>USA</u>

GAMS Development Corp.
1217 Potomac Street, NW
Washington, DC 20007
USA

Phone: +1 202 342 0180
Fax: +1 202 342 0181

sales@gams.com
support@gams.com

http://www.gams.com