Portfolio Optimization: A Technical Perspective

Franz Nelißen
FNelissen@gams.com

GAMS Software GmbH
www.gams.de

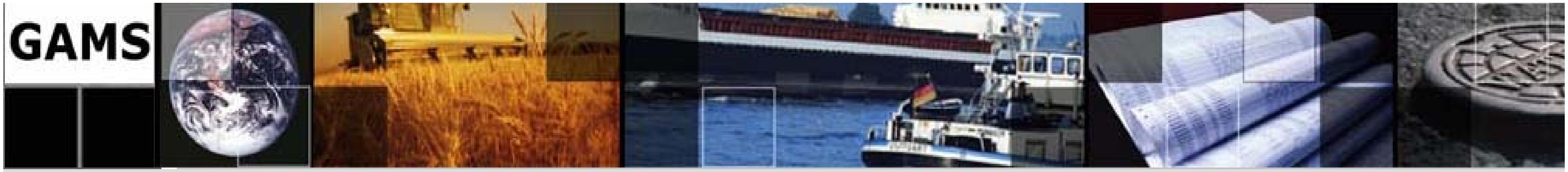
OR 2006
Karlsruhe, Germany, September 6-8, 2006

Agenda

- Introduction
- Mathematical Optimization in Finance
- An illustrative Example: The Mean Variance Model
- Advanced Portfolio Optimization Models
- Grid Computing

Agenda

- **Introduction**
- Mathematical Optimization in Finance
- An illustrative Example: The Mean Variance Model
- Advanced Portfolio Optimization Models
- Grid Computing



GAMS Development / GAMS Software

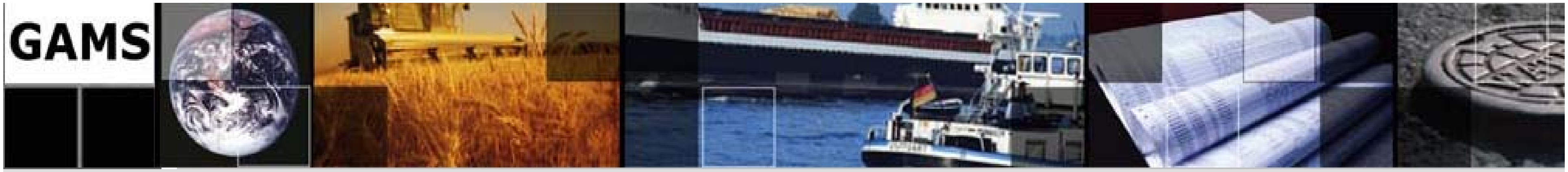
- Roots: **Research project** World Bank 1976
- Pioneer in **Algebraic Modeling Systems** used for economic modeling
- Went **commercial** in 1987
- **Offices** in Washington, D.C and Cologne

- Professional **software tool provider, not a consulting company**
- Operating in a **segmented niche market**
- Broad **academic** & **commercial** user base and network

General Algebraic Modeling System

Agenda

- Introduction
- **Mathematical Optimization in Finance**
- An illustrative Example: The Mean Variance Model
- Advanced Portfolio Optimization Models
- Grid Computing



Mathematical Optimization in Finance

Very active research field with significant contributions and important practical applications

Some of the reasons:

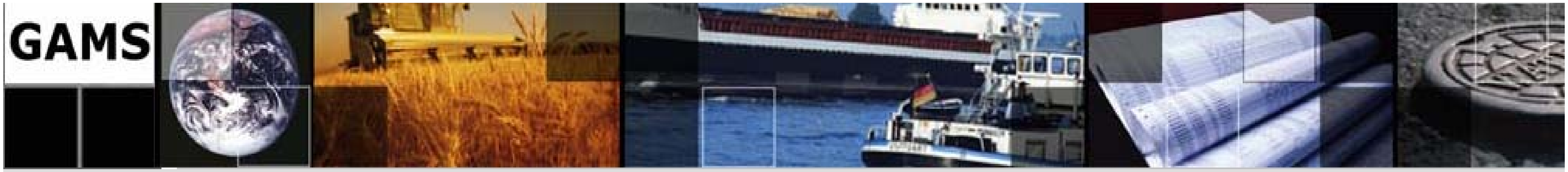
- Continual stream of challenging problems with obvious impact of uncertainty
- High availability of data
- Validation potential – benchmarking
- Very competitive and liquid markets

Many instruments, tools and strategies

GAMS

Portfolio Optimization Models

- Mean-Variance Model
- *Portfolio models for fixed income*
- Scenario optimization
- Stochastic programming



Change in Focus

Computation **Past**	*Model* **Now**	*Application* **Future**
▪ **Algorithm limits application** ▪ Problem representation low priority ▪ Large expensive projects ▪ Long development times ▪ Centralized expert groups ▪ High computational costs → **Users left out**	▪ **Modeling skill limits applications** ▪ Algebraic model representation ▪ Smaller projects and rapid development ▪ Decentralized modeling teams ▪ Machine independence → **Users involved**	▪ **Domain expertise limits application** ▪ Off-the-shelf GUI ▪ Models embedded in business applications ▪ Links to other types of models ▪ Internet/Web →**Users hardly aware of model**

GAMS

Modeling Approaches

- Programming languages: C++, Delphi, Java, VBA, …
- Spreadsheets
- Specialized tools

- **Algebraic Modeling Languages**
  - Balanced mix of declarative and procedural elements
  - Open architecture and interfaces to other systems
  - Different layers with separation of:
    - model and data
    - model and solution methods
    - model and operating system
    - model and interface

Agenda

- Introduction
- Mathematical Optimization in Finance
- **An illustrative Example: The Mean Variance Model**
- Advanced Portfolio Optimization Models
- Grid Computing

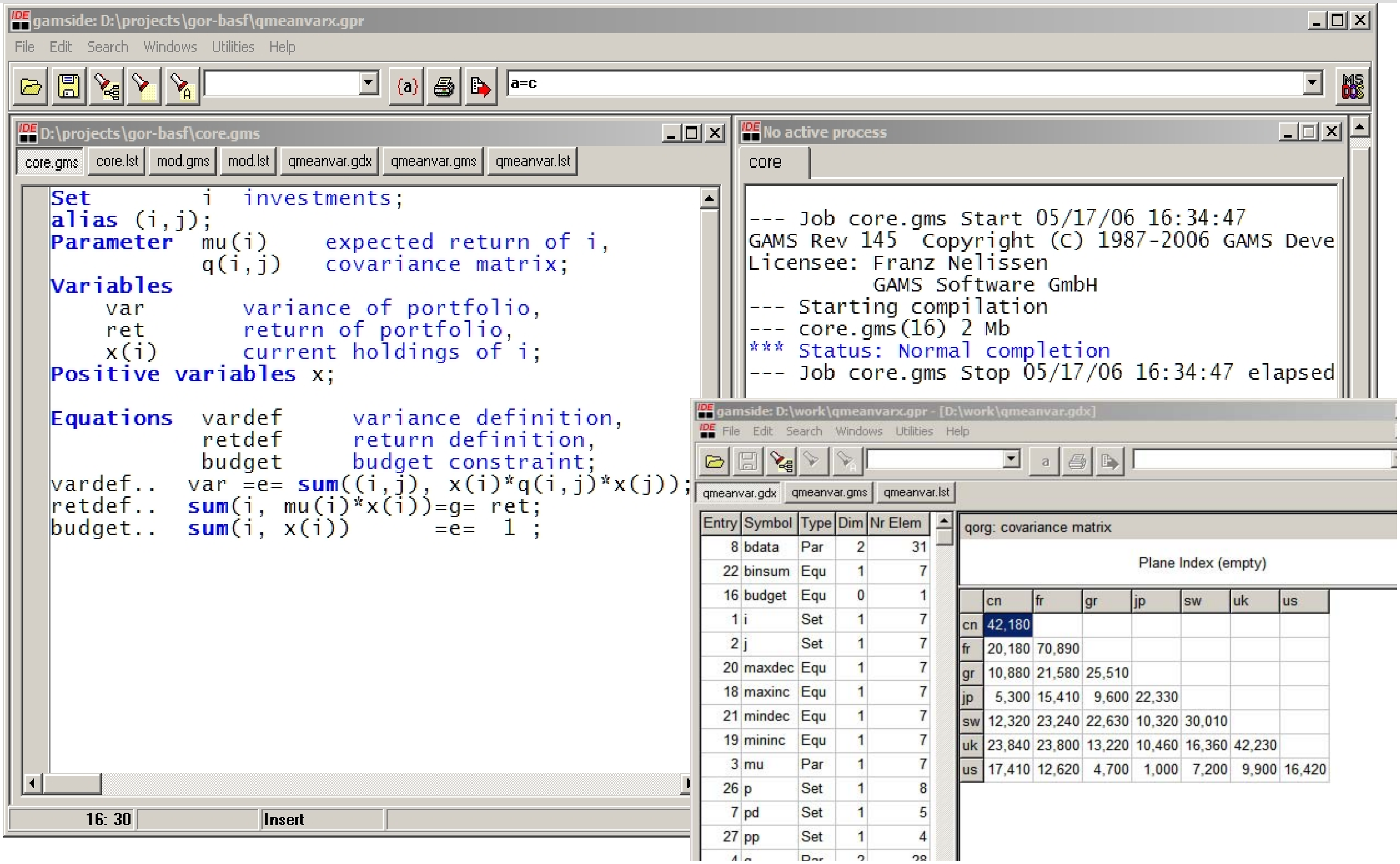
GAMS

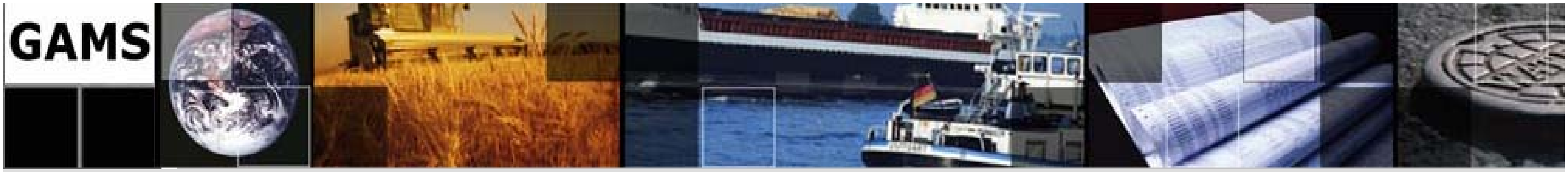
MV Model Algebra

Variance of Portfolio	$Min \sum_{i=1}^{I} \sum_{j=1}^{J} x_i \, Q_{i,j} \, x_j$
Target return	$s.t. \quad \sum_{i=1}^{I} \mu_i x_i \geq r$
Budget constraint	$\sum_{i=1}^{I} x_i = 1$
No short sales	$x_i \geq 0$

GAMS

Declarative Model and some Data





Modeling Issues

Basic MV-Model: Quadratic model

Solver
- NLP Codes (CONOPT, MINOS,...) *or*
- QCP Codes (Cplex, Mosek, Xpress)
  - take advantage of special structure

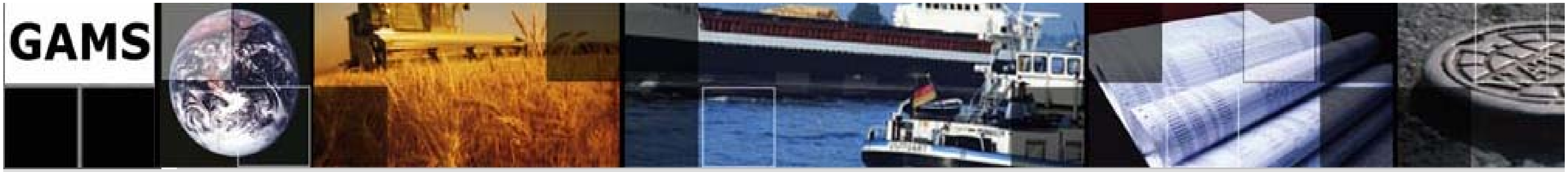
Large problem instances can be solved routinely

Business Rules

- Institutional or legal requirements
- Additional constraints, which have to be satisfied
- Trading restrictions
- Not defined by modeling experts
- Independent of risk model

Simple business rules: Do not change the model type:

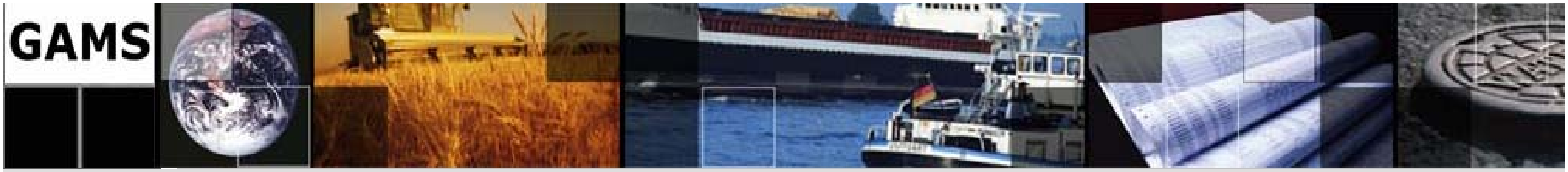
- Short selling
- Risk free borrowing
- Upper or lower bounds on certain instruments



More Complex Business Rules

Require introduction of integer (binary) variables:

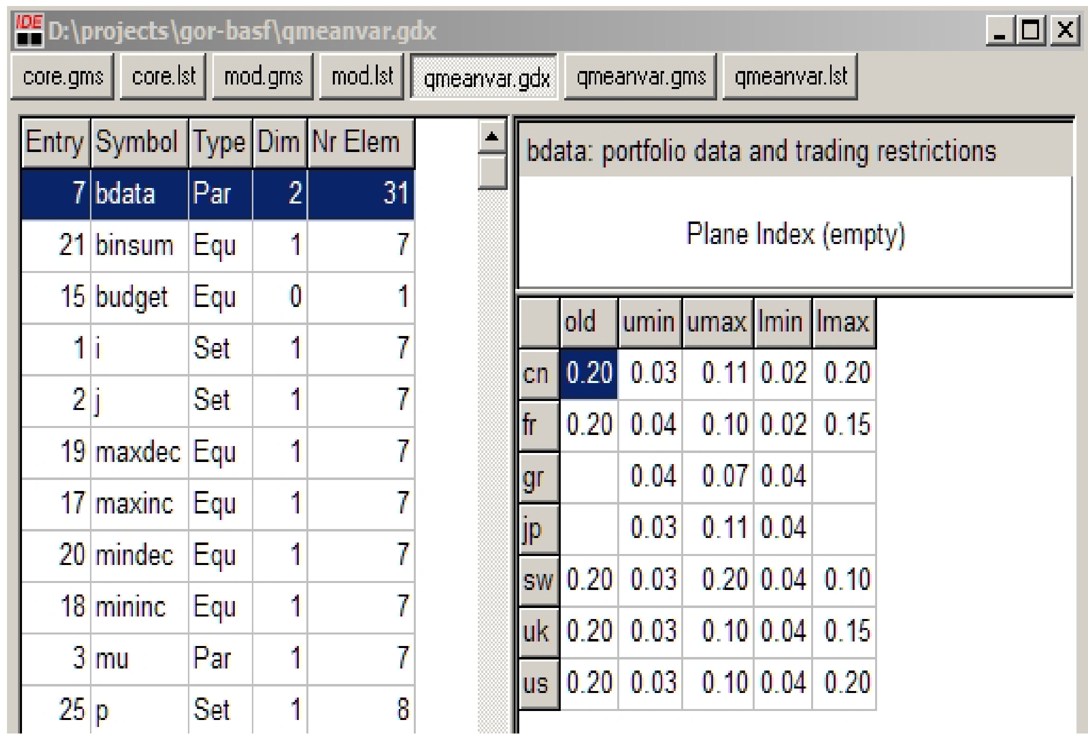
- **Cardinality Constraint**: Restrict number of investments y_i in portfolio
- **Threshold Constraint**: Investments x_i can only be purchased at certain minimum $l_{l,i}$ or maximum $l_{u,I}$
- more trading restrictions …



Trading Restrictions

"Zero or Range"-Constraint:

- Revision of existing (not optimized) portfolio
- "Zero or Range“ - Constraint: **Either** no trade **or** the trade must stay between pre-defined ranges both for purchase and selling
- Portfolio turnover: The total purchase of investments x_i may not exceed some threshold τ



Entry	Symbol	Type	Dim	Nr Elem
7	bdata	Par	2	31
21	binsum	Equ	1	7
15	budget	Equ	0	1
1	i	Set	1	7
2	j	Set	1	7
19	maxdec	Equ	1	7
17	maxinc	Equ	1	7
20	mindec	Equ	1	7
18	mininc	Equ	1	7
3	mu	Par	1	7
25	p	Set	1	8

	old	umin	umax	lmin	lmax
cn	0.20	0.03	0.11	0.02	0.20
fr	0.20	0.04	0.10	0.02	0.15
gr		0.04	0.07	0.04	
jp		0.03	0.11	0.04	
sw	0.20	0.03	0.20	0.04	0.10
uk	0.20	0.03	0.10	0.04	0.15
us	0.20	0.03	0.10	0.04	0.20

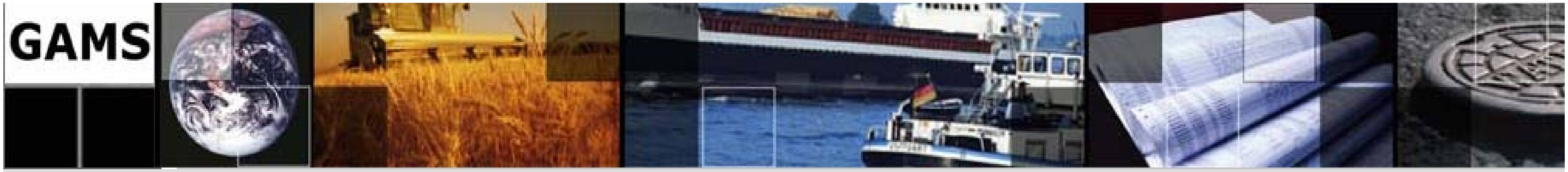
e.g. cn: either no trade (20%) or new share between 23-31% (u) or between 0-18% (l)

GAMS

GAMS Formulation

```
Variables
 xi(i)   fraction of portfolio increase,
 xd(i)   fraction of portfolio decrease,
 y(i)    binary switch for increasing current holdings of i,
 z(i)    binary switch for decreasing current holdings of i;
Binary Variables  y, z; Positive Variables xi, xd;
Equations
 xdef(i)    final portfolio definition,
 maxinc(i)  bound of maximum lot increase of fraction of i,
 mininc(i)  bound of minimum lot increase of fraction of i,
 maxdec(i)  bound of maximum lot decrease of fraction of i,
 mindec(i)  bound of minimum lot decrease of fraction of i,
 binsum(i)  restricts use of binary variables,
 turnover   restricts maximum turnover of portfolio;

xdef(i)..   x(i)   =e=  bdata(i,'old') - xd(i) + xi(i);
maxinc(i).. xi(i)  =l=  bdata(i,'umax')* y(i);
mininc(i).. xi(i)  =g=  bdata(i,'umin')* y(i);
maxdec(i).. xd(i)  =l=  bdata(i,'lmax')* z(i);
mindec(i).. xd(i)  =g=  bdata(i,'lmin')* z(i);
binsum(i).. y(i) + z(i)        =l= 1;
turnover.. sum(i, xi(i))       =l= tau;
```

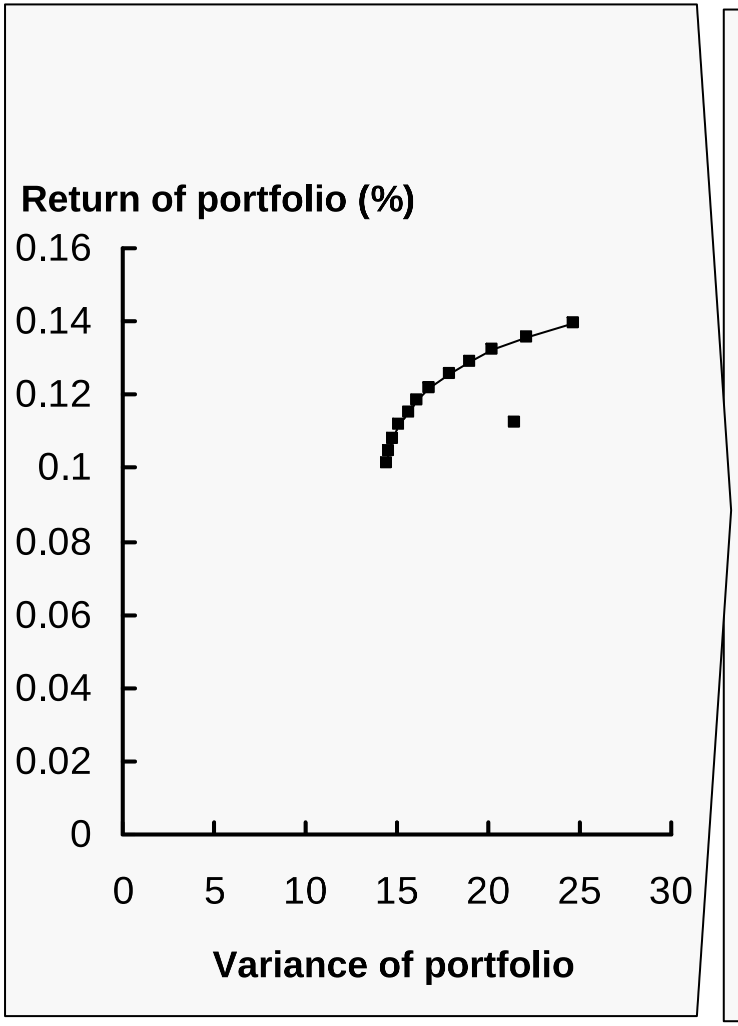
Model Type: MIQCP

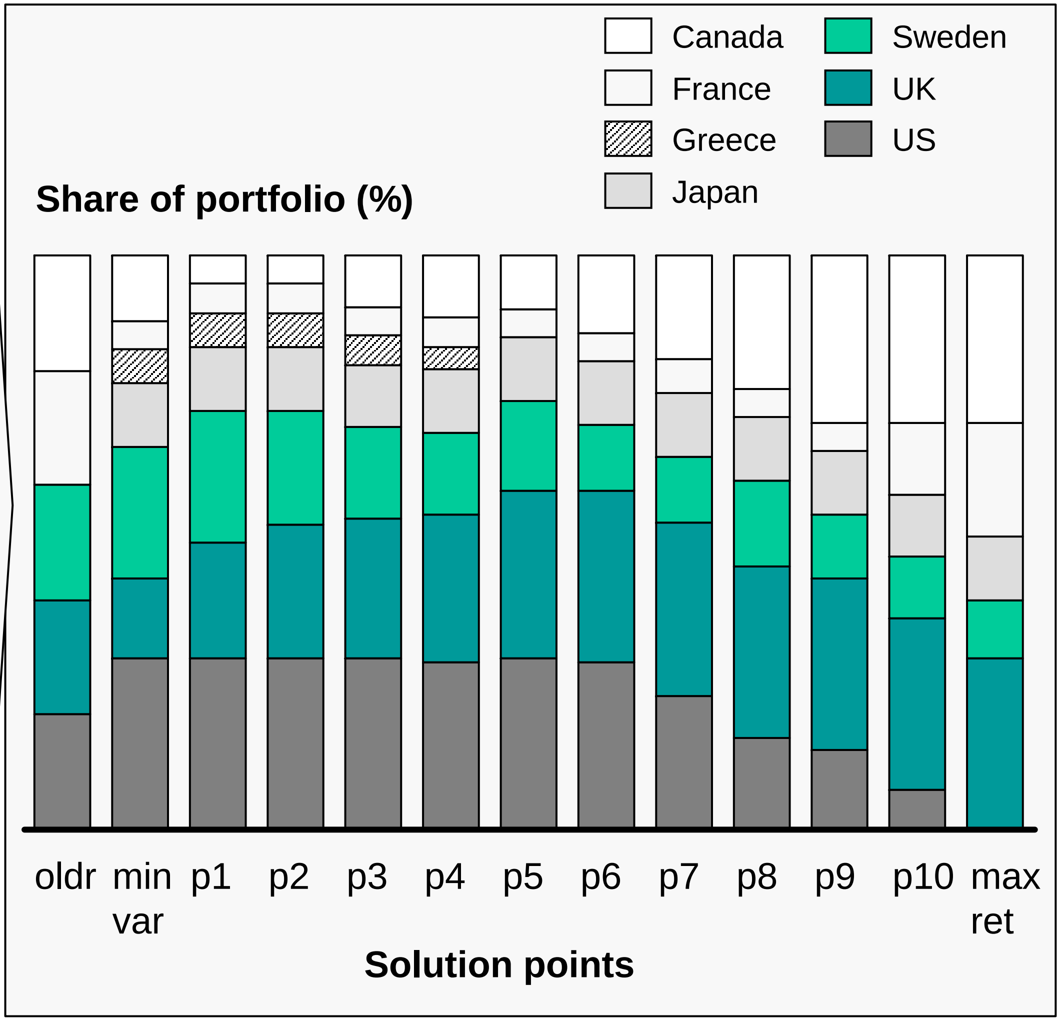


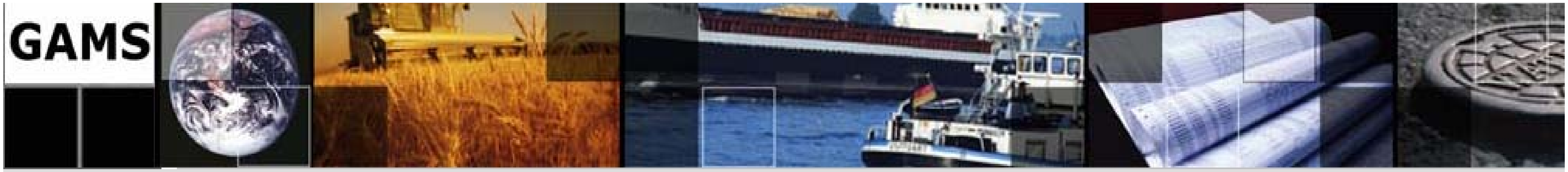
Procedural Elements

```
$gdxin data                                       # get data & setup model
$load i mu q
q(i,j) = 2*q(j,i) ;  q(i,i) = q(i,i)/2;
Model var / all / ;
set p     points for efficient frontier /minv, p1*p8, maxr/,
    pp(p) points used for loop          /      p1*p8      /;
parameter minr, maxr,rep(p,*),  repx(p,i);

# get bounds for efficient frontier
solve var minimizing v using miqcp;               #find portfolio with minimal variance
minr = r.l; rep('minv','ret') = r.l;
rep('minv','var') = v.l; repx('minv',i) = x.l(i);

solve var maximizing r using miqcp;               #find portfolio with maximal return
maxr = r.l; rep('maxr','ret')= r.l;
rep('maxr','var')=v.l;repx('maxr',i)= x.l(i);

loop(pp,                                          #calculate efficient frontier
         r.fx = minr + (maxr-minr)/(card(pp)+1)*ord(pp);
         solve var minimizing v using miqcp;
         rep(pp,'ret') =r.l;rep(pp,'var') = v.l;repx(pp,i)= x.l(i);
     );

Execute_Unload 'results.gdx',rep, repx;   # export results to GDX & Excel
Execute 'GDXXRW.EXE results.gdx par=repx rng=Portfolio!a1 Rdim=1';
Execute 'GDXXRW.EXE results.gdx par=rep  rng=Frontier!a1  Rdim=1';
```

Efficient Frontier and Portfolios (τ = 0.3)







Scenario Optimization Models

Scenarios capture complex interactions between multiple risk factors

- Different methods for risk measurement:
  - Mean Absolute Deviation Models
  - Index Tracking Models
  - Expected Utility Models
  - VAR Models (linear Version: CVAR)
- Models are solved over all scenarios

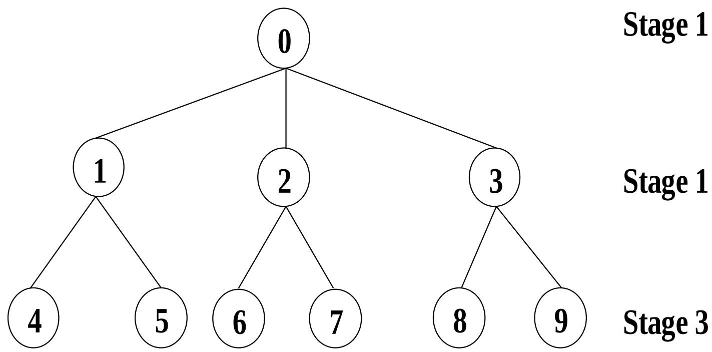
Modeling Issues:

- Linear Models, but business rules may introduce binary variables
- Lots of independent scenarios, which can be handled in parallel

Stochastic Programming (SP)

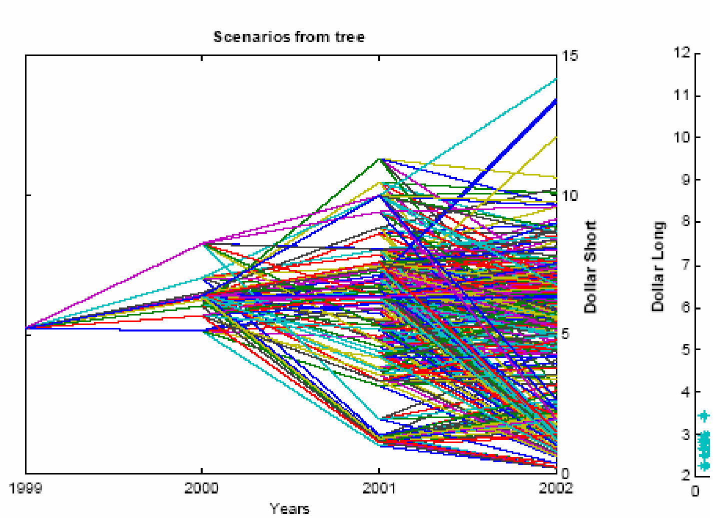
Stochastic Programming models allow sequence of decisions.

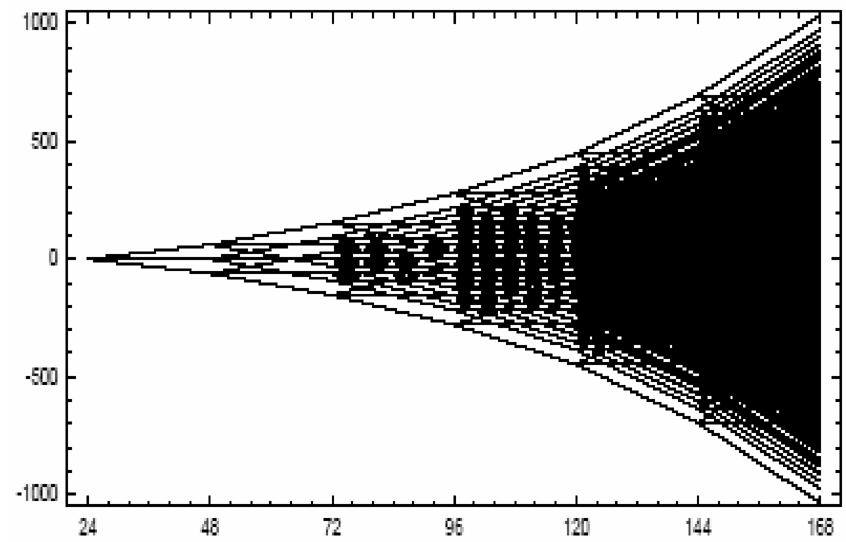
- **Scenarios**: Complete set of possible discrete realizations of the uncertain parameters with probabilities
- **Stages**: Decisions points. First stage decisions now, second stage decision (depending of the outcome of the first stage decision) after a certain period and so on
- **Recourse**: Decision variables can adept to the different out comes of the random parameters at each stage



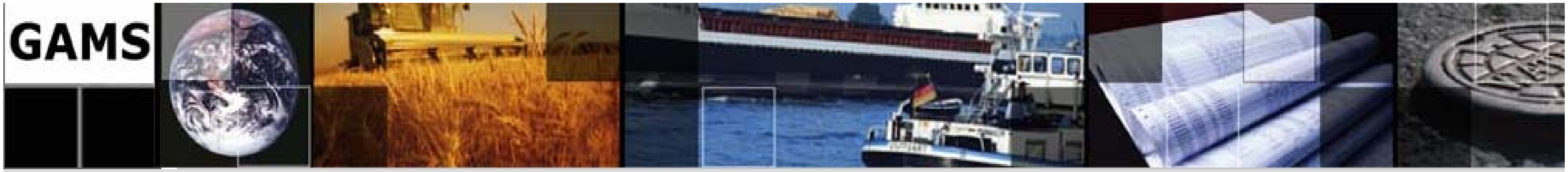
More Complex Scenario Trees

Figure 1: US dollar short rate scenarios





Original load scenario tree

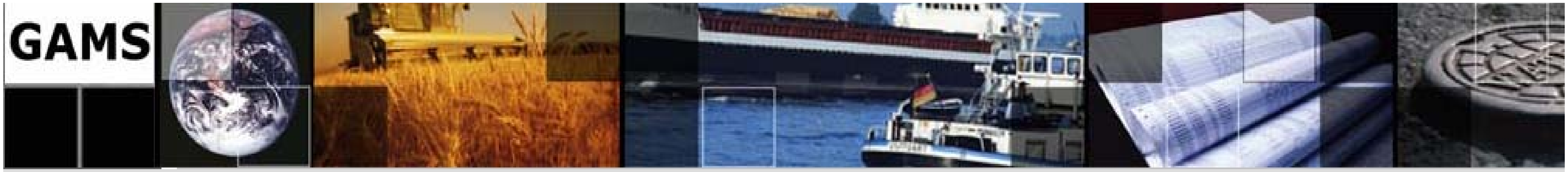


Challenges

Deterministic equivalent: Includes all scenarios and stages
→ Size of model explodes
- Generation difficult
- Solution may not be possible
- Interpretation and validation of results

→ Less applications than one may expect

But: Number of uncertain parameters is small:
- Efficient representation of the uncertain data within the Algebraic Modeling System?
- Scenarios may only differ slightly
- Problems are structured



Current Developments

New language elements :

- Special expressions and conventions for stages and scenario trees
- Random distributions for some problem data
- Support of scenario reduction techniques dramatically reduces the size of deterministic equivalent
- Automatic translation of problem description into format for various SP-solvers (DECIS, SPLINE…)
- Support for parallel optimization

But:

- Different approaches
- Not yet clear which standards will be adopted

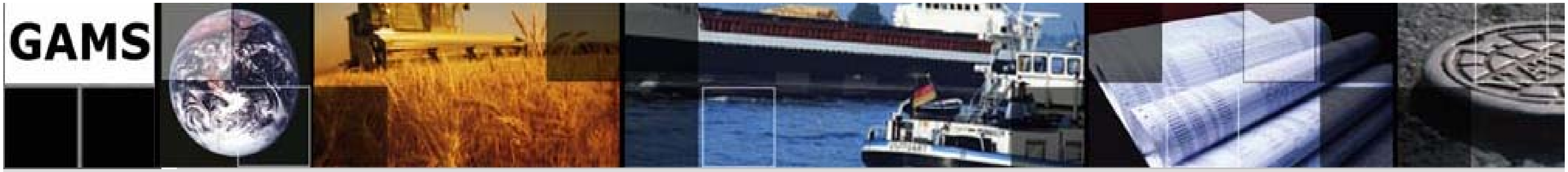
More Theory and Templates

Theory

- **Practical Financial Optimization** (forthcoming) by S. Zenios
- **A Library of Financial Optimization Models** (forthcoming) by A. Consiglio, S. Nielsen, H. Vladimirou and S. Zenios
- **Financial Optimization** by S. Zenios (ed.)

Templates available online

- **GAMS Model Library:** http://www.gams.com/modlib/libhtml/subindx.htm
- **Course Notes „Financial Optimization“:** http://www.gams.com/docs/contributed/financial/



Agenda

- Introduction
- Mathematical Optimization in Finance
- An illustrative Example: The Mean Variance Model
- Advanced Portfolio Optimization Models
- **Grid Computing**

Grid Computing

```
loop(pp,                                  #calculate efficient frontier
     r.fx = minr + (maxr-minr)/(card(pp)+1)*ord(pp);
     solve var minimizing v using miqcp;
     rep(pp,'ret') =r.l;rep(pp,'var') = v.l;repx(pp,i)= x.l(i);
     );
```

Imagine…

.. you have to solve 1.000's of independent scenarios..

.. and you can do this very rapidly for little additional money…

.. without having to do lots of cumbersome programming work..

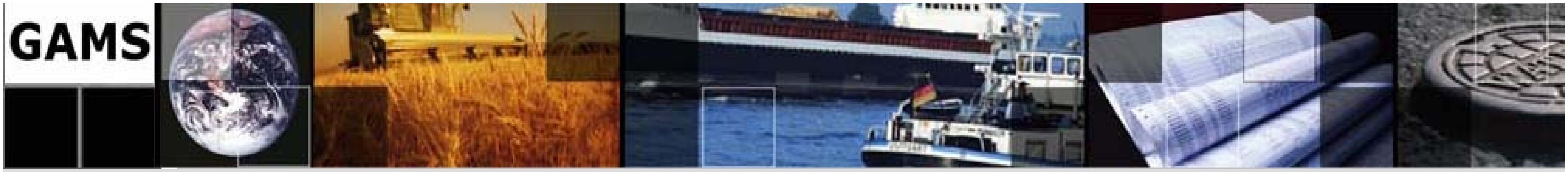
Grid Computing

GAMS

What is Grid Computing?

A pool of connected computers managed and available as a common computing resource

- Effective sharing of CPU power
- Massive parallel task execution
- Scheduler handles management tasks
- E.g. Condor, Sun Grid Engine, Globus
- Can be rented or owned in common
- Licensing & security issues



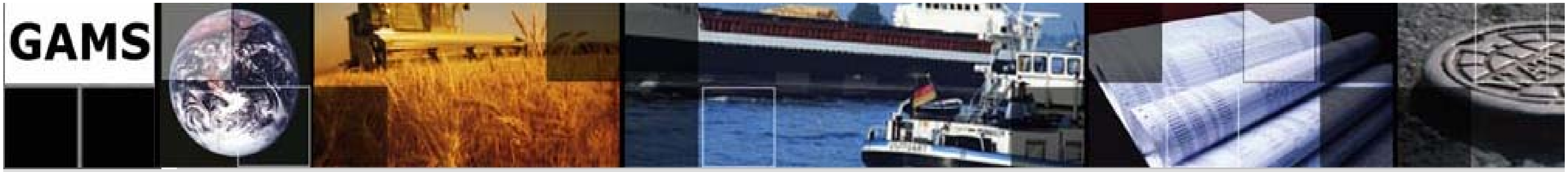
Advantages of Grid Computing

- Solve a certain number of scenarios faster, e.g:
  - sequential: 50 hours
  - parallel (200 CPUs): ~15 minutes

    → Cost is $100 (2$ CPU/h)
- Get better results by running more scenarios*:

#SIM	VaR error	CVaR error
1000	5.42%	6.74%
20,000	1.21%	1.49%

* http://www.tc.cornell.edu/NR/shared/Presentations/24Feb04.Garp.pdf



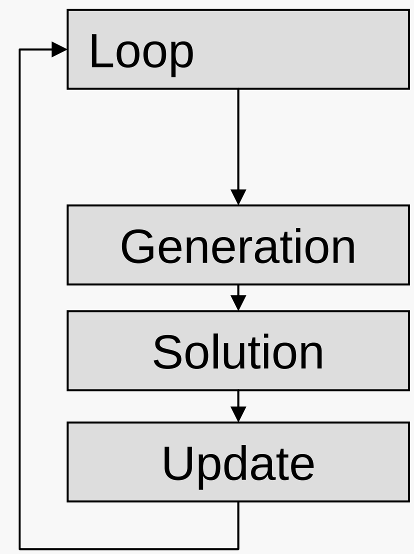
GAMS & Grid Computing

- **Scalable**:
  - support of massive grids, **but also**
  - multi-cpu / multiple cores desktop machines
  - “1 CPU - Grid”
- Platform **independent**
- Only **minor changes** to model required
- **Separation** of model and solution method
  → Model stays **maintainable**

GAMS

Simple Serial Solve Loop

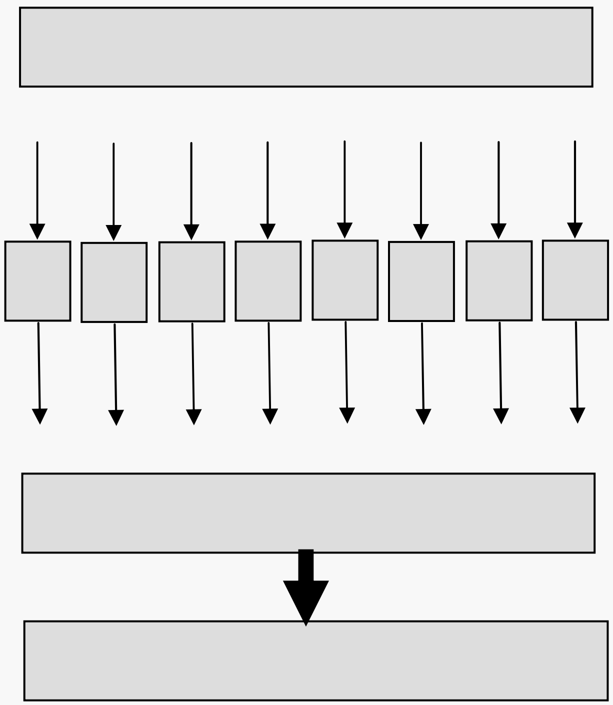
```
Loop(p(pp),
   ret.fx = rmin +(rmax-rmin)
     /(card(pp)+1)*ord(pp) ;
   Solve minvar min var using miqcp;
   xres(i,p)          = x.l(i);
   report(p,i,'inc') = xi.l(i);
   report(p,i,'dec') = xd.l(i)
  );
```



How do we get to parallel and distributed computing?

GRID Specific Enhancements

1. Submission of jobs
2. "Grid Middleware"
   - Distribution of jobs
   - Job execution
3. Collection of solutions
4. Processing of results

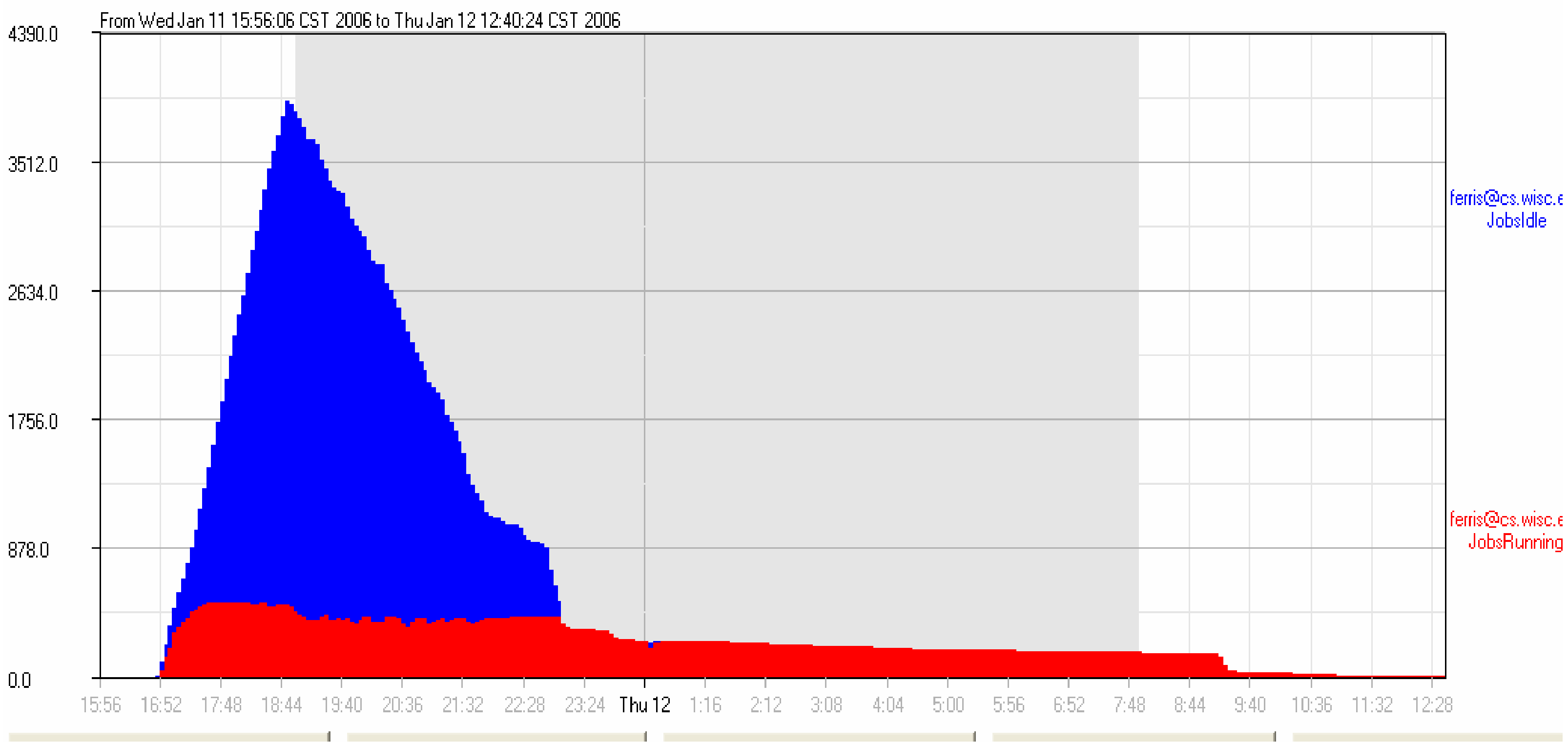


Results for 4096 MIPS on Condor Grid

- Submission started Jan 11,16:00
- All jobs submitted by Jan 11, 23:00
- All jobs returned by Jan 12, 12:40
  - 20 hours wall time, 5000 CPU hours
  - Peak number of CPU's: 500

Talk: Thursday, 08:30

"Chemie-Hörsaal 1"

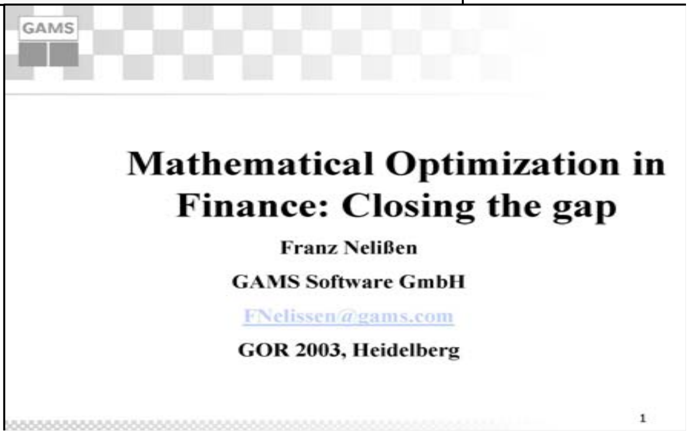


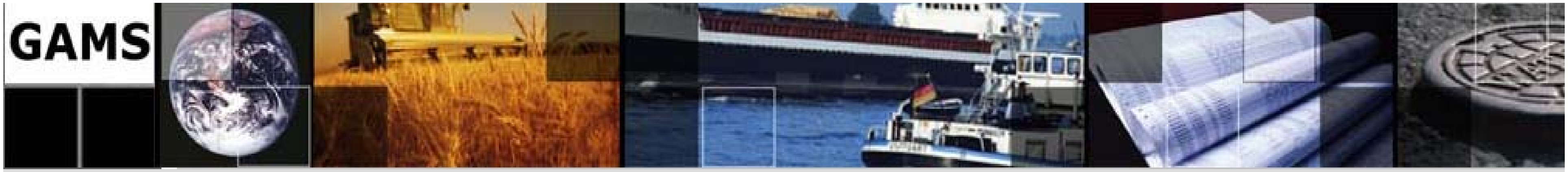
Conclusions and Summary

- Finance is a success story for OR applications
- Rich set of different risk models available

- Incorporating business rules may increase complexity of problems but is essential
- Large classes of problems can be solved without major problems

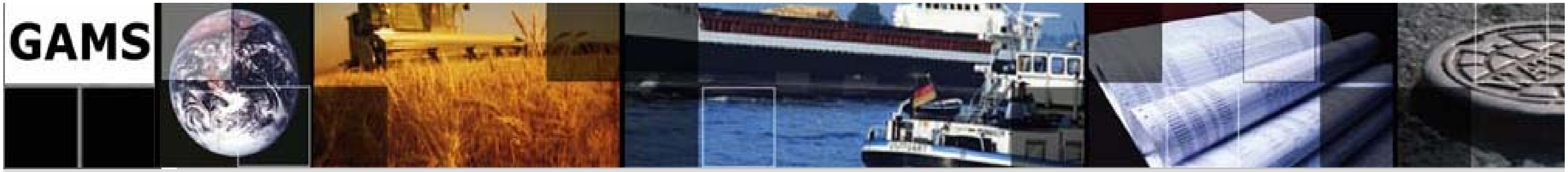
- Stochastic programming still challenging
- Grid Computing now offers lots of promising developments





The End

Thank you!
… Questions?



Contacting GAMS

Europe:

GAMS Software GmbH
Eupener Str. 135-137
50933 Cologne
Germany
Phone: +49 221 949 9170
Fax: +49 221 949 9171
http://www.gams.de

USA:

GAMS Development Corp.
1217 Potomac Street, NW
Washington, DC 20007
USA
Phone: +1 202 342 0180
Fax: +1 202 342 0181
http://www.gams.com