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
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
Portfolio Optimization Models

- Mean-Variance Model
- Portfolio models for fixed income
- Scenario optimization
- Stochastic programming
Change in Focus

### Computation Past
- Algorithm limits application
- Problem representation low priority
- Large expensive projects
- Long development times
- Centralized expert groups
- High computational costs
  ➔ Users left out

### Model Now
- Modeling skill limits applications
- Algebraic model representation
- Smaller projects and rapid development
- Decentralized modeling teams
- Machine independence
  ➔ Users involved

### Application Future
- 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
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
<table>
<thead>
<tr>
<th>Agenda Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
</tr>
<tr>
<td>Mathematical Optimization in Finance</td>
</tr>
<tr>
<td><strong>An illustrative Example: The Mean Variance Model</strong></td>
</tr>
<tr>
<td>Advanced Portfolio Optimization Models</td>
</tr>
<tr>
<td>Grid Computing</td>
</tr>
</tbody>
</table>
### MV Model Algebra

<table>
<thead>
<tr>
<th>Section</th>
<th>Equation/Inequality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance of Portfolio</td>
<td>[ \text{Min } \sum_{i=1}^{I} \sum_{j=1}^{J} x_i Q_{i,j} x_j ]</td>
</tr>
<tr>
<td>Target return</td>
<td>[ \sum_{i=1}^{I} \mu_i x_i \geq r ]</td>
</tr>
<tr>
<td>Budget constraint</td>
<td>[ \sum_{i=1}^{I} x_i = 1 ]</td>
</tr>
<tr>
<td>No short sales</td>
<td>[ x_i \geq 0 ]</td>
</tr>
</tbody>
</table>
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 …
"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 $x_j$ may not exceed some threshold $\tau$

![Table and diagram showing trading restrictions](image)

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

Variables
- \( x(i) \)  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
- \( \text{xdef}(i) \)  final portfolio definition,
- \( \text{maxinc}(i) \)  bound of maximum lot increase of fraction of \( i \),
- \( \text{mininc}(i) \)  bound of minimum lot increase of fraction of \( i \),
- \( \text{maxdec}(i) \)  bound of maximum lot decrease of fraction of \( i \),
- \( \text{mindec}(i) \)  bound of minimum lot decrease of fraction of \( i \),
- \( \text{binsum}(i) \)  restricts use of binary variables,
- \( \text{turnover} \)  restricts maximum turnover of portfolio;

\[
\begin{align*}
\text{xdef}(i) & \quad x(i) \ =e= \ bdata(i,'old') - xd(i) + xi(i); \\
\text{maxinc}(i) & \quad xi(i) \ =l= \ bdata(i,'umax')* y(i); \\
\text{mininc}(i) & \quad xi(i) \ =g= \ bdata(i,'umin')* y(i); \\
\text{maxdec}(i) & \quad xd(i) \ =l= \ bdata(i,'lmax')* z(i); \\
\text{mindec}(i) & \quad xd(i) \ =g= \ bdata(i,'lmin')* z(i); \\
\text{binsum}(i) & \quad y(i) + z(i) \ =l= 1; \\
\text{turnover} & \quad \text{sum}(i, xi(i)) \ =l= \tau;
\end{align*}
\]

Model Type: MIQCP
Procedural Elements

```plaintext
$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 ($\tau = 0.3$)

- **Share of portfolio (%)**
  - Canada
  - Sweden
  - France
  - UK
  - Greece
  - Japan
  - US

- **Solution points**
  - oldr
  - min
  - var
  - p1
  - p2
  - p3
  - p4
  - p5
  - p6
  - p7
  - p8
  - p9
  - p10
  - max
  - ret

- **Graphs:**
  - Return of portfolio (%)
  - Variance of portfolio

- **Values:**
  - 0.02
  - 0.04
  - 0.06
  - 0.08
  - 0.1
  - 0.12
  - 0.14
  - 0.16

- **Axes:**
  - 0 5 10 15 20 25 30

- **Additional information:**
  - $\tau = 0.3$
## 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**

- **Scenarios from tree**
- **Dollar Short**, **Dollar Long**

**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

<table>
<thead>
<tr>
<th>Theory</th>
<th>Templates available online</th>
</tr>
</thead>
</table>
| • **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.) | • **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
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..

```
loop(pp,
   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);
);
```
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*:

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**
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?
<table>
<thead>
<tr>
<th>Country</th>
<th>Address</th>
<th>Phone</th>
<th>Fax</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>GAMS Development Corp.</td>
<td>+1 202 342 0180</td>
<td>+1 202 342 0181</td>
<td><a href="http://www.gams.com">http://www.gams.com</a></td>
</tr>
<tr>
<td></td>
<td>1217 Potomac Street, NW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20007</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>USA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe</td>
<td>GAMS Software GmbH</td>
<td>+49 221 949 9170</td>
<td>+49 221 949 9171</td>
<td><a href="http://www.gams.de">http://www.gams.de</a></td>
</tr>
<tr>
<td></td>
<td>Eupener Str. 135-137</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50933 Cologne</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Germany</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phone: +49 221 949 9170</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fax: +49 221 949 9171</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>