Agenda

- Then and now
- Recent Enhancements
- Extending Algebraic Modeling
- Well then?
Agenda

Then and now

Recent Enhancements

Extending Algebraic Modeling

Well then?
Then …

In Table 17.1 we list sizes and attributes of representative models that are “large” in the sense that they are near the limit of what is practical on a personal computer, along with the model generation time (GAMS) and solution time (solver), both in minutes. These examples were run on an 8 MHz AT with an 80287 coprocessor and 640K of RAM. The times shown are to give you a rough idea of what is possible: these are not precisely controlled benchmarks, and we have a host of performance improvements in mind for the near future.

Table 17.1: Problem Characteristics

<table>
<thead>
<tr>
<th>Name</th>
<th>Number of Rows</th>
<th>Number of Columns</th>
<th>Number of Nonzeros</th>
<th>Generation Time*</th>
<th>Solution Time*</th>
<th>Iterations</th>
<th>Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>DINAMICO</td>
<td>318</td>
<td>425</td>
<td>4156</td>
<td>3.0</td>
<td>30.1</td>
<td>628</td>
<td>MINOS</td>
</tr>
<tr>
<td>SARF</td>
<td>532</td>
<td>542</td>
<td>3949</td>
<td>37.7</td>
<td>115.8</td>
<td>2775</td>
<td>MINOS</td>
</tr>
<tr>
<td>FERTD</td>
<td>458</td>
<td>2968</td>
<td>7252</td>
<td>11.4</td>
<td>28.3</td>
<td>1368</td>
<td>ZOOM</td>
</tr>
<tr>
<td>CAMCOE</td>
<td>243</td>
<td>280</td>
<td>1356</td>
<td>0.8</td>
<td>7.0</td>
<td>189</td>
<td>MINOS</td>
</tr>
<tr>
<td>GANGES</td>
<td>274</td>
<td>357</td>
<td>1405</td>
<td>1.8</td>
<td>7.3</td>
<td>187</td>
<td>MINOS</td>
</tr>
<tr>
<td>YMICERM</td>
<td>168</td>
<td>258</td>
<td>953</td>
<td>0.9</td>
<td>7.6</td>
<td>600</td>
<td>ZOOM</td>
</tr>
<tr>
<td>EGYPT</td>
<td>281</td>
<td>618</td>
<td>3168</td>
<td>4.0</td>
<td>25.3</td>
<td>1551</td>
<td>ZOOM</td>
</tr>
</tbody>
</table>

*Measured in minutes.
*The problem is too big for MINOS. ZOOM was used instead.
**A nonlinear problem. 63% of the non-zeroes are nonlinear.
***A nonlinear problem. 58% of the non-zeroes are nonlinear.
****A binary problem, with 55 binary variables (solved with a relative termination criterion of 10%).
*****A linear problem, solved using XMP which is contained within ZOOM.
... and now

<table>
<thead>
<tr>
<th>Type</th>
<th>s in 1988</th>
<th>s in 2007</th>
<th>Improvement Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>camcge</td>
<td>NLP</td>
<td>468</td>
<td>0.031</td>
</tr>
<tr>
<td>dinamico</td>
<td>LP</td>
<td>1986</td>
<td>0.125</td>
</tr>
<tr>
<td>egypt*</td>
<td>MIP</td>
<td>1758</td>
<td>0.015</td>
</tr>
<tr>
<td>fertd*</td>
<td>MIP</td>
<td>2382</td>
<td>0.062</td>
</tr>
<tr>
<td>ganges</td>
<td>NLP</td>
<td>546</td>
<td>0.109</td>
</tr>
<tr>
<td>sarf</td>
<td>LP</td>
<td>9210</td>
<td>0.139</td>
</tr>
<tr>
<td>yemcem*</td>
<td>MIP</td>
<td>510</td>
<td>0.140</td>
</tr>
</tbody>
</table>

à Hardly predictable how much performance gain comes from hardware and how much from software

* MIP 1988 solver ZOOM, 2007 solver CPLEX
Agenda

- Then and now
- Recent Enhancements
- Extending Algebraic Modeling
- Well then?
Release Notes

New Solvers
- COIN-OR Solvers (http://www.coin-or.org/)
  • MINLP solver: CoinBonmin
- AlphaECP
  • MINLP solver
  • Extended Cutting Plane method by T. Westerlund and T. Lastusilta (Abo Akademi University, Finland)
- LINDOGlobal
  • finds proven optimal solutions to non-convex MINLP
  • Global Optimization Solver from LINDO Systems, Inc

Improvements
• BARON, CONOPT, CPLEX, MOSEK, XPRESS,...
Multiple Threads

• CPLEX
  – parallel extension for B&B and interior point solver
  – concurrent optimizer
  – academic license includes 4 threads

• MOSEK
  – parallel extension for the interior solver comes free of charge
  – concurrent optimizer

• XPRESS
  – parallel extension for B&B and interior point solver
  – academic license includes 4 threads

• XA (XAPAR)
Grid Computing

- pool of connected computers managed and available as common computing resource
- e.g. Condor, Sun Grid Engine

```plaintext
Loop(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);
```

Diagram:
- Loop
- Generation
- Solution
- Update
Minor Changes to Model
SUNgrid

www.network.com

- On-demand grid computing service operated by Sun Microsystems
- Access to enormous computing power over Internet
- Opteron-based servers with 4 GB of RAM per CPU
- Solaris 10 OS, and Sun Grid Engine 6 software.
- $1 per CPU-hour
- GAMS Distribution 22.5 available
- 250 free CPU hours to new users

www.gams.com/sungrid

F. Nelissen: “Grid Computing in Finance using an Algebraic Modeling System”, Thursday 2:30-3pm, Room A2 4 1.32
Agenda

- Then and now
- Recent Enhancements
- Extending Algebraic Modeling
- Well then?
Algebraic Modeling Languages

- Problem format is old/traditional

\[ \min_x f(x) \text{ s.t. } g(x) \leq 0, h(x) = 0 \]

- Limited support for symbolic reformulations
  - logical constructs
  - constraints tightening, softening
  - stochastic programming

Automated symbolic reformulations will gain importance
Symbolic Reformulations and GAMS

- **GAMS/DECIS**
  - solves two-stage stochastic linear programs with recourse
  - two-stage decomposition (Benders)
  - stores only one instance of the problem and generates scenario sub-problems as needed
  - solution Strategies (Universe problem/Importance sampling)

- **GAMS/NLPEC**
  - solves MPECs as NLPs
  - 23 different reformulation strategies

- **GAMS/PATHNLP**
  - solves NLPs as MCPs
  - internal reformulation via KKT conditions
  - requires 1\textsuperscript{st} and 2\textsuperscript{nd} order derivatives
Symbolic Reformulations and GAMS

- GAMS/Convert
  - model translation tools
  - GAMS à other formats/languages
  - makes use of infix notation available
  - GAMS
  - NLP2MCP
  - CHull

Option GAMS
- converts indexed input model into scalar model
- “standardized” model format
- easier to deal with
- confidentiality issues
- nonlinearity information is still available
NLP2MCP

- Michael C. Ferris and Jeffrey D. Horn (1998)
  
  “NLP2MCP: Automatic conversion of nonlinear programs into mixed complementarity problems”

  - Reformulate NLP via KKT into MCP model
  - Requires 1\textsuperscript{st} derivatives

- Original approach
  - individual tool that translates an indexed model
  - tough approach due to potential beastiness of NLPs

- New approach
  - integrated in GAMS/Convert
  - result is the source of a scalar MCP model
NLP2MCP

• Why convert to MCP
  – Second order information implicitly available
  – Remove “superbasic” dependence
  – Exploit multiplier information
  – Incorporate into MPEC

• Likely that MCP solver will find a solution
  – Solution is only guaranteed to be feasible for the original problem
  – In the convex case, every KKT point corresponds to a global solution of the NLP
Reference is paper by R.T. Rockafellar (1999)

- Classical problem

\[
\begin{align*}
\min_{x_1, x_2, x_3} & \quad \exp(x_1) \\
\text{s.t.} & \quad \log(x_1) = 1 \\
& \quad x_2^2 \leq 2 \\
& \quad x_1 / x_2 = \log(x_3), \ 3x_1 + x_2 \leq 5, \ x_1 \geq 0, \ x_2 \geq 0
\end{align*}
\]

- Soft penalization of constraints

\[
\begin{align*}
\min_{x_1, x_2, x_3} & \quad \exp(x_1) + 5 \| \log(x_1) - 1 \|^2 + 2 \max(x_2^2 - 2, 0) \\
\text{s.t.} & \quad x_1 / x_2 = \log(x_3), \ 3x_1 + x_2 \leq 5, \ x_1 \geq 0, \ x_2 \geq 0
\end{align*}
\]
Extending NLPs automatically

• Motivation
  – automates tedious algebra writing
  – avoids mistakes
  – infeasibility analysis
  – real world applications always need a solution

• Implement a “solver” that
  – extends an NLP based on user provided information
  – passes the modified model to an NLP solver
  – reads the solution back
  – is aware of a class of different extension functions
GAMS “Solver” ENLP

- Converts input model to a scalar NLP or MCP
- Reads ENLP information file to extend the model

\begin{verbatim}
$onecho > %gams.srcdir%enlpinfo.scr
 e1 sqr 5
 e2 MaxZ 2
$offecho
\end{verbatim}

- Solves the created ENLP / EMCP
- Reads solution back
- Solver options
  - `SolveasMCP` ENLP will be converted to and solved as MCP
  - `SubSolver` subsolver to run
  - `SubSolverOpt` optfile value to pass to the subsolver
  - `Terminate` terminate after generating scalar GAMS model
  - …
Symbolic Reformulations and GAMS

- GAMS/Convert
  - model translation tools
  - GAMS à other formats/languages
  - makes use of infix notation available
  - GAMS
  - NLP2MCP è ENLP
  - Chull è LOGMIP
Convex Hull

- Convex Hull reformulation of linear and nonlinear models with disjunctions
- User provides disjunction information

```plaintext
file dj2 / %gams.sordir%loginfo.scr /; dj.nd=0; dj.nw=0; dj.lw=0;
put dj2 ' * convex hull for example 1';
loop(llt(j,jj),
    put / 'disj ' y.tn(j,jj) ' ' seq.tn(j,jj) ' else ' seq.tn(jj,j));
putclose;
```

- Result is a scalar GAMS model representing the Convex Hull
Logical Mixed Integer Programming (LogMIP)

- Developed by Aldo Vecchietti and Ignacio E. Grossmann

  http://www.logmip.ceride.gov.ar/

- Add-on modeling framework to formulate models with disjunction and logic propositions

- Language Compiler

- Solution algorithms and techniques for solving linear and nonlinear disjunctive programming problems

- Add-on tool as of GAMS Distribution 21.7 (April 2005)
LogMIP Example

\[\begin{align*}
\text{min } Z &= T \\
\text{s.t. } & T \geq x_1 + 8 \\
& T \geq x_2 + 5 \\
& T \geq x_3 + 6 \\
& \begin{bmatrix}
Y_1 \\
x_1 - x_3 + 5 \leq 0
\end{bmatrix} \lor \begin{bmatrix}
\neg Y_1 \\
x_3 - x_1 + 2 \leq 0
\end{bmatrix} \\
& \begin{bmatrix}
Y_2 \\
x_2 - x_3 + 1 \leq 0
\end{bmatrix} \lor \begin{bmatrix}
\neg Y_2 \\
x_3 - x_2 + 6 \leq 0
\end{bmatrix} \\
& \begin{bmatrix}
Y_3 \\
x_1 - x_2 + 5 \leq 0
\end{bmatrix} \lor \begin{bmatrix}
\neg Y_3 \\
x_2 - x_1 \leq 0
\end{bmatrix} \\
& T, x_1, x_2, x_3 \geq 0 \\
& Y_k \in \{true, false\}, k = 1, 2, 3.
\end{align*}\]

<table>
<thead>
<tr>
<th>Stage</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>4</td>
<td>-</td>
</tr>
</tbody>
</table>

Raman & Grossmann (1994)
LogMIP “Solver”

- Reformulation
  - Convex Hull (only linear)
  - Big M (only linear)
  - Need MIP solver

- Logic based Method
  - Logic-based Outer Approximation
  - Solves a series of NLP and MIP sub-problems
  - Needs NLP an MIP solver
CPLEX Indicator Constraints

- New way of expressing relationships among variables

- Specify binary variable to control whether or not a constraint takes effect

- Can be numerically more robust and accurate than conventional Big M formulations

```plaintext
file copt / cplex.opt /; put copt '* indicators for example 1'
loop(1t(j,jj),
   put / 'indic ' seq.tn(j,jj) ' $' y.tn(j,jj) ' yes
      / 'indic ' seq.tn(jj,j) ' $' y.tn(j,jj) ' NO
   );

m.optfile=1; option mip=cplex;
solve m using MIP minimizing t;
```
Agenda

- Then and now
- Recent Enhancements
- Extending Algebraic Modeling
- Well then?
Well then?

- GAMS already supports a wide collection of established mathematical programming classes
  
  (LP, MIP, QCP, MIQCP, NLP, DNLP, MINLP, MCP, MPEC, CNS, Global)

- New research breaks out of traditional MP classes
  - broaden algebraic modeling
  - proven research add-ons to GAMS
  - engaged in systematic modification of constraints
    (activate, deactivate, soften, tighten)
  - But! Each approach has its individual intermediate format
Automatic Reformulation Framework

Need of an integrated framework for automated mathematical programming reformulations

- provide new facilities for seamless integration of new model types
- benefit models with constructs like
  - disjunctions
  - indicator constraints
  - extended nonlinear programs
  - conditional value at risk
  - stochasticity (chance constraints)
Evolution in the GAMS way

0. – try as research code
   – committed to backward compatibility
   √

1. – analyze big picture
   – generalize
   √

2. – implement sub-language
   – does it proof itself?
   → 22.6

3. – define standard
   – integrate in GAMS language
   – work with solver developers
   ?
## Contacting GAMS

<table>
<thead>
<tr>
<th>Region</th>
<th>Address</th>
<th>Phone</th>
<th>Fax</th>
<th>Email</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe</td>
<td>GAMS Software GmbH</td>
<td>+49 221 949 9170</td>
<td>+49 221 949 9171</td>
<td><a href="mailto:info@gams.de">info@gams.de</a></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><a href="mailto:support@gams-software.com">support@gams-software.com</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50933 Cologne, Germany</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>GAMS Development Corp.</td>
<td>+1 202 342 0180</td>
<td>+1 202 342 0181</td>
<td><a href="mailto:sales@gams.com">sales@gams.com</a></td>
<td><a href="http://www.gams.com">http://www.gams.com</a></td>
</tr>
<tr>
<td></td>
<td>1217 Potomac Street, NW Washington, DC 20007</td>
<td></td>
<td></td>
<td><a href="mailto:support@gams.com">support@gams.com</a></td>
<td></td>
</tr>
</tbody>
</table>