Interactions between Modeling Systems and Advanced Solvers

Jan-Hendrik Jagla
Michael Bussieck
Steven Dirkse
Alex Meeraus

GAMS Development Corp.
www.gams.com
GAMS Software GmbH
www.gams.de

jhjagla@gams.com
mbussieck@gams.com
sdirkse@gams.com
ameeraus@gams.com
Agenda

- General Algebraic Modeling System
- Current State of AMLs
- Extending Algebraic Modeling
- Implementation
Agenda

General Algebraic Modeling System

Current State of AMLs

Extending Algebraic Modeling

Implementation
**GAMS at a Glance**

### General Algebraic Modeling System

- **Roots:** World Bank, 1976
- **Went commercial in 1987**
- **GAMS Development Corp. (DC)**
- **GAMS Software GmbH (Cologne)**

- Broad academic & commercial user community and network
GAMS at a Glance

General Algebraic Modeling System

- Algebraic Modeling Language
- 25+ Integrated Solvers
- 10+ Supported Model Types
- 10+ Supported Platforms
- Connectivity- & Productivity Tools
  - IDE
  - Model Libraries
  - GDX, Interfaces & Tools
  - Grid Computing
  - Benchmarking
  - Compression & Encryption
  - Deployment System
  - …
Recent Enhancements

• New Solvers
  – Coin-OR Solver (Glpk, Cbc, Ipopt, Bonmin)
  – AlphaECP
  – LINDOglobal

• New Platforms: Solaris on Sparc64 and MacOS on Intel

• GAMS supports CPLEX 11 features
  – Improved Mixed Integer Programming Performance
  – Enhanced Parallel MIP
  – Multiple MIP Solutions
  – Performance Tuning Tool

• Extended Mathematical Programming (EMP)
Agenda

- General Algebraic Modeling System
- Current State of AMLs
- Extending Algebraic Modeling
- Implementation
Algebraic Modeling Languages (AML)

- Traditional but fundamental view of AMLs

  - Interface
  - Data
  - Model
  - Solver
  - Interface

- **Key concept:** Different layers with separation of
  - model and data
  - model and solution methods
  - model and operating system
  - model and interface
Current state: Model-Side

- Traditional problem format

\[
\min_x c(x) \quad \text{s.t.} \quad A_1(x) \leq b_1, \quad A_2(x) = b_2
\]

- Interactions between models possible
  - Series of models
  - Scenario analyses
  - Iterative sequential feedback
  - Decomposition
AMLs support a wide collection of established mathematical programming classes through solver clusters
New trends in research broaden algebraic modeling

• Global Optimization

• Solvers that are based on automated symbolic reformulation of model types

• Hybrid tools that make use of traditional model representation plus additional information as
  – logical constructs (indicators, disjunctions)
  – constraint modifications
    • activation and deactivation
    • softening and tightening
    • probability
Agenda

- General Algebraic Modeling System
- Current State of AMLs
- Extending Algebraic Modeling
- Implementation
Global Optimization

- Practical optimization problems are often nonlinear and non-convex, with discrete variables
- They may contain disconnected feasible regions with multiple local optima

• Find the best of all

AMLs perfect platform to promote GO

- Experience with (local) nonlinear optimization
- Separation of model and solution technology
- Established Quality Assurance
- Mathematical algebra is required (not black box)
  - Baron, LINDOglobal
Reformulation-based Solvers

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

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

• Logical Mixed Integer Programming (LogMIP)
  – Reformulation and logic-based methods on Generalized Disjunctive Programs (GDP)

• Indicator constraints (CPLEX)
  – Alternative to conventional BigM formulations

• Extended Nonlinear Programming (ENLP)
  – Softening and tightening constraints

• ...
Need of a framework for automated mathematical programming reformulations that

- integrates the different hybrid approaches
- makes GAMS ready for new cutting-edge approaches
- provides new facilities for seamless integration of new model types (Conic Programming, SDP, ...)
- automates symbolic reformulations to avoid error-prone and time-consuming manual algebra (re)writing
- makes additional information consistently available
Agenda

- General Algebraic Modeling System
- Current State of AMLs
- Extending Algebraic Modeling
- Implementation
“Evolution in the GAMS way”

- committed to backward compatibility
- try as research code

- analyze the big picture
- find a generalization

- implement as sub-language
- does it proof itself?

- generally accepted notation?
- integrate in GAMS language
- work with solver developers
GAMS/Convert

• Model translation tools
  – GAMS à other formats/languages
  – Algebraic information still available

• GAMS
  – Creates scalar “standardized” model

• NLP2MCP
  – Converts model into a scalar MCP model

• CHull
  – Creates the convex hull of a (nonlinear) disjunctive program
• Why convert to MCP
  – Second order information implicitly available
  – New model types cannot be formulated as (N)LP
    • Bi-level, embedded problems
    • Exploit multiplier information

• 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
CHull

- Convex Hull reformulation of linear and nonlinear models with disjunctions

- User provides disjunction information

```gams
file dj2 / '%gams.scrdir%loginfo.scr' /; dj.nd=0; dj.nw=0; dj.lw=0;
put dj2 /* convex hull for example 1' ';
loop (lt(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
Extended Nonlinear Programming

Soft penalization of constraints

• Model

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

• Additional information

\[
\begin{align*}
\text{\$onecho > %gams.srcdir%empinfo.scr} \\
e1 \text{ sqr 5} \\
e2 \text{ MaxZ 2} \\
\text{\$offecho}
\end{align*}
\]

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

• Reformulates model based on user-provided information
  – CHull
  – ENLP
  – EMCP (ENLP plus NLP2MCP)
  – ...

• Facilitates to only write out the reformulated model

• Passes the generated model to an appropriate solver

• Reads solution back into original space
Conclusion

- Continuously bridge the gap between academia and industry
- Incorporate cutting edge approaches
- Be able to solve new model classes
  - using existing methods
  - make it easy for solver developers to provide new algorithms
Thanks for your time!

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
sales@gams.com
support@gams.com

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
info@gams.de
support@gams-software.com
Convex Hull (old format)

* this writes the loginfo file for the CHull
file dj / %gams.scrdir%loginfo.scr /; dj.nd=0; dj.nw=0; dj.lw=0;
put dj '0'/ card(lt);
loop(lt(j,jj),
   put / '2' / '* 2' j.tl ' ' jj.tl ' 1' / '1' / 'seq 2' j.tl ' ' jj.tl / 0
   / '* 2' j.tl ' ' jj.tl ' 1' / '1' / 'seq 2' jj.tl ' ' j.tl / 0 );
putclose;

option minlp=convert; m.optfile=1;
$echo chull uich.gms > convert.opt
solve m us minlp min t;
GDP Example

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

\[T, x_1, x_2, x_3 \geq 0\]

\[Y_k \in \{\text{true, false}\}, k = 1, 2, 3.\]

<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)