GAMS’ Extended Mathematical Programming Framework

Jan-Hendrik Jagla  
Michael Ferris  
Alex Meeraus

jhjagla@gams.com  
ferris@cs.wisc.edu  
ameeraus@gams.com

GAMS Software GmbH  
www.gams.de

GAMS Development Corp.  
www.gams.com
## Agenda

<table>
<thead>
<tr>
<th>General Algebraic Modeling System</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Solution Concepts</td>
</tr>
<tr>
<td>Extended Mathematical Programming</td>
</tr>
</tbody>
</table>
Agenda

General Algebraic Modeling System

New Solution Concepts

Extended Mathematical Programming
GAMS at a Glance

General Algebraic Modeling System

- Roots: World Bank, 1976
- Went commercial in 1987
- GAMS Development Corp.
- GAMS Software GmbH
- Broad academic & commercial user community and network
GAMS at a Glance

**General Algebraic Modeling System**

- Algebraic Modeling Language
- 25+ 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
  - …
Agenda

- General Algebraic Modeling System
- New Solution Concepts
- Extended Mathematical Programming
Traditional but fundamental concept of AMLs

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

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

- Support for complementarity constraints

- Interactions between models possible
  - Series of models
  - Scenario analyses / parallelized model runs
  - Iterative sequential feedback
  - Decomposition
Current state: Solver-Side

Support of a wide collection of established MP classes through solver cluster!

- **Tremendous algorithmic and computational progress**

  **LP**
  - in fact only restricted by available memory

  **MIP**
  - Some (academic) problems still unsolvable
  - Commercial problems mostly docile

  **NLP/MINLP**
  - Predictions are problem and data specific, global vs. local solutions
Non-traditional solution concepts

• MP with Equilibrium Constraints (MPEC)
  - NLPEC
    • Solves MPECs through reformulation into NLPs

• Solving non-integer models as MCPs
  - PATHNLP
    • reformulation via KKT conditions (1\textsuperscript{st} and 2\textsuperscript{nd} order deriv.)

• Mathematical Programming System for General Equilibrium analysis
  - MPSGE

• Indicator Constraints (CPLEX)
  - Alternative to conventional BigM formulations
Non-traditional solution concepts

• Global Optimization
  ➢ BARON, LINDOGLOBAL
    • Proven global optimum
  ➢ LGO, OQNLP
    • Stochastic convergence to global optimum

• Stochastic Programming
  ➢ 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)

• …
New solution concepts

- Extended Nonlinear Programs
- Embedded Complementarity Systems
- Bilevel Programs
- Disjunctive Programs
- ...

- Breakouts of traditional MP classes
- No conventional syntax
- Limited support with common model representation
- Incomplete/experimental solution approaches
- Lack of reliable/any software
What now?

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
• Distribute information as part of the production system
• Express extended model in symbolic form and apply existing matured solution technology

⇒ Extended Mathematical Programming (EMP)
Agenda

- General Algebraic Modeling System
- New Solution Concepts
- Extended Mathematical Programming
GAMS “Solver” EMP

- Translation services
- Uses existing language features to specify additional model features
- Expresses extended model in symbolic form and passes it to existing solution methods via embedded GAMS calls
- Reads solution back into original space
- Facilitates to write out the reformulated model
Extended Nonlinear Programming

Soft penalization of constraints

- Model:

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

- Additional information:

  ```
  $onecho > %emp.info%
  Adjustequ
  e1 sqr 5
  e2 MaxZ 2
  $offecho
  $onecho > %gams.scdir%empinfo2.scr
  Strategy MCP
  Adjustequ
  e1 sqr 5
  e2 MaxZ 2
  $offecho
  ```

- EMP Tool creates the NLP model (or the MCP via KKT):

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

- Models with side constraints/variables:

\[
\min_x f(x, y) \\
\text{s.t. } g(x, y) \leq 0 \quad (\perp \lambda \geq 0) \\
H(x, y, \lambda) = 0 \quad (\perp y \text{ free})
\]

- Additional Information:

\$\text{onecho > %emp.info%} \\
\text{dualequ H y} \\
\text{dualvar \lambda g} \\
\text{offecho}\$

- EMP Tool creates the MCP model:

\[
\nabla_x \mathcal{L}(x, y, \lambda) \quad \perp x \text{ free} \\
-\nabla_\lambda \mathcal{L}(x, y, \lambda) \quad \perp \lambda \geq 0 \\
H(x, y, \lambda) = 0 \quad \perp y \text{ free}
\]
ECS Example

- Rutherford, Thomas F. (http://www.mpsge.org/nlptarget/)

```gams
parameter
   kterm Terminal capital stock

UTIL..   UTILITY =E= SUM(t, 10 * dfactor(t) * L(t) * LOG(C(t)/L(t)));
CC(t)..   C(t) =E= Y(t) - I(t);
YY(t)..   Y(t) =E= phi * L(t)**(1-kvs) * K(t)**kvs;
KK(t)..   K(t) =L= (1-delta)**10 * K(t-1) + 10 * I(t-1) + kinit$tirst(t);
TERMCAP.. kterm =E= sum(tlast, (1-delta)**10 * K(tlast) + 10 * I(tlast));

model ramsey NLP Model using parameter kterm /all/;

set iter /iter1*iter20/;

kterm = kinit * power(l+g, card(t));

parameter
   invest(t,iter) Investment in successive iterations
   kt(iter) Terminal capital stock in successive iterations;

loop(iter,
   kt(iter) = kterm;
   solve ramsey maximizing UTILITY using NLP;
   invest(t,iter) = I.L(t);
   kterm = sum(tlast(t), K.L(tlast) * Y.L(t)/Y.L(t-1));
);
```
EMP Formulation

*Substitute TERMCAP of NLP by TERMCAPV (using variable KTERMV instead of parameter kterm)
TERMCAPV.. KTERMV =E= sum(tlast, (1-delta)**10 * K(tlast) + 10 * I(tlast));

*First-order-condition for terminal capital stock variable
SSTERM.. sum(tlast(t),I(t)/I(t-1) - Y(t)/Y(t-1)) =E= 0;

model ramseynlpd /UTIL,CC,YY,KK,TERMCAPV,SSTERM/;

$onecho > %emp.info%
dualequ SSTERM KTERMV
$offecho

option nlp=emp;

solve ramseynlpd maximizing UTILITY using nlp;

Extended Mathematical Programming (EMP)
------------------------------------------
--- EMP Summary (errors=0)
    Adjusted Equations   =   0
    Dual Variable Maps   =   0
    Dual Equation Maps   =   1
    Bilevel Followers    =   0
    Disjunctions         =   0
--- The model C:\home\distrib\tvis_alpha\convtest\emp\225a\emp.scr will be solved by GAMS
---
Hierarchical Models

• Bilevel Program:

\[ \begin{align*}
\min_{x,y} & \quad f(x, y) \\
\text{s.t.} & \quad g(x, y) \leq 0, \\
& \quad y \text{ solves } \min_s v(x, s) \text{ s.t. } h(x, s) \leq 0
\end{align*} \]

• Additional Information:

\$\text{onecho > %emp.info%} \\
Bilevel x min v h \\
\$\text{offecho} \$

• EMP Tool automatically creates an MPEC by expressing the lower level optimization problem through its optimality conditions
Bilevel Model


Variables

\[ \text{variables} \quad z, x_1, x_2, x_3, x_4, h_1, h_2, u_1, u_2, u_3, u_4, v_1, v_2, v_3, v_4; \]

Equations

\[ \text{equations} \quad \text{defobj}, \text{defh1}, \text{defh2}, a_1, e_1, e_2; \]

Outer Problem

\[ \text{defobj..} \quad z = e = \text{sqr}(x_1+x_2-2) + \text{sqr}(x_3+x_4-2); \]
\[ a_1.. \quad x_1-x_2 = e = 3; \]

Inner Problem 1

\[ \text{defh1..} \quad h_1 = e = \text{sqr}(u_1-x_1) + \text{sqr}(u_2-x_2) + \text{sqr}(u_3-x_3) + \text{sqr}(u_4-x_4); \]
\[ e_1.. \quad 3*u_1 + u_2 + 2*u_3 + u_4 = e = 6; \]

Inner Problem 2

\[ \text{defh2..} \quad h_2 = e = \text{sqr}(v_1-x_1) + \text{sqr}(v_2-x_2) + \text{sqr}(v_3-x_3) + \text{sqr}(v_4-x_4); \]
\[ e_2.. \quad v_1 + v_2 + v_3 + 2*v_4 = e = 7; \]

Model

\[ \text{model} \quad \text{bilevel} / \text{all} / \]
EMP Information File + EMP Summary Log

```plaintext
option nlp=emp;

$onecho > %emp.info%
bilevel x1 x2 x3 x4
min h1 defh1 e1
min h2 defh2 e2
$offecho

solve bilevel us nlp min z;
```

Extended Mathematical Programming (EMP)
----------------------------------------
--- EMP Summary  (errors=0)  
  Adjusted Equations = 0
  Dual Variable Maps = 0
  Dual Equation Maps = 0
  Bilevel Followers = 2
  Disjunctions        = 0
--- The model C:\home\distrib\tvis_alpha\convtest\emp\225a\emp.scr will be solved by GAMS
```
Disjunction Example


• Three jobs (A,B,C) must be executed sequentially in three steps, but not all jobs require all the stages. Once a job has started it cannot be interrupted.

• The objective is to obtain the sequence of task, which minimizes the completion time.

<table>
<thead>
<tr>
<th>Stage Job</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>
Data Definition

```plaintext
table p(j,s) processing time

    1  2  3
A  5  3
B  3  2
C  2  4

alias (j,jj),(s,ss);

parameter c(j,s) stage completion time
    w(j,jj) maximum pair wise waiting time
    pt(j) total processing time;
set    less(j,jj) upper triangle;

c(j,s) = sum(ss$(ord(ss)<=ord(s)), p(j,ss));
w(j,jj) = smax(s, c(j,s) - c(jj,s-1));
pt(j)  = sum(s, p(j,s));
less(j,jj) = ord(j) < ord(jj);
```
Basic Model Definition

\begin{verbatim}
variables t completion time 
    x(j) job starting time 
positive variable x;

equations comp(j) job completion time 
    seq(j,jj) job sequencing j before jj;

comp(j).. t =g= x(j) + pt(j); 

seq(j,jj)$\!(not sameas(j,jj)).. x(j) + w(j,jj) =l= x(jj);

Above equation is incomplete!

If (j,jj) is active then (jj,j) should be relaxed
\end{verbatim}
Traditional BigM Formulation

```gams
binary variable y(j,jj) job precedence;

parameter big the famous big M;

big = sum(j, pt(j));
big=100000;

seq(j,jj)$(not sameas(j,jj)).. 

x(j) + w(j,jj) =l= x(jj) + big*( y(j,jj) $less(j,jj) 
+ (1-y(jj,j))$less(jj,j));

model m / all /; m.optcr=0;

solve m using MIP minimizing t;
```
CPLEX Indicator Formulation

```gams
seq(j,jj)$not sameas(j,jj).. x(j) + w(j,jj) =e= x(jj);

binary variable y(j,jj) job precedence;

equation dummy force names into model;

dummy.. sum(less(j,jj), y(j,jj)) =g= 0;

model m / all /;

file copt / cplex.opt /; put copt '* indicators for example 1';
loop(less(j,jj),
   put / 'indic ' seq.tn(j,jj) '$' y.tn(j,jj) yes
   / 'indic ' seq.tn(jj,j) '$' y.tn(j,jj) no);
putclose; m.optfile=1;

solve m using MIP minimizing t;
```

* indicators for example 1

<table>
<thead>
<tr>
<th>Indicators</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>seq('A','B')$y('A','B')</code></td>
<td>YES</td>
</tr>
<tr>
<td><code>seq('B','A')$y('A','B')</code></td>
<td>NO</td>
</tr>
<tr>
<td><code>seq('A','C')$y('A','C')</code></td>
<td>YES</td>
</tr>
<tr>
<td><code>seq('C','A')$y('A','C')</code></td>
<td>NO</td>
</tr>
<tr>
<td><code>seq('B','C')$y('B','C')</code></td>
<td>YES</td>
</tr>
<tr>
<td><code>seq('C','B')$y('B','C')</code></td>
<td>NO</td>
</tr>
</tbody>
</table>
EMP Disjunction Formulation

seq(j,jj)$ \text{(not sameas}(j,jj)).. \ x(j) + w(j,jj) =l= x(jj);

model m / all /;

file emp / '%emp.info%' /; put emp ' * EMP for example 1';
loop(less(j,jj),
    put / 'disjunction * ' seq.tn(j,jj) ' else ' seq.tn(jj,j) );
putclose;

option mip=emp;

solve m using MIP minimizing t;

* EMP for example 1
  disjunction * seq('A','B') else seq('B','A')
  disjunction * seq('A','C') else seq('C','A')
  disjunction * seq('B','C') else seq('C','B')
• AdjustEQU equ abs|sqr|maxz|huber|... { weight { param } }

• DualEqu {equ var}
• DualVar {var equ}

• BiLevel {var} { MAX|MIN obj {equ} }

• Disjunction [NOT] var|* {equ} { ELSEIF [NOT] var|* {equ} } [ ELSE {equ} ]
Conclusion

EMP is

- a framework for automated symbolic reformulations
- non-exhaustive and experimental
- free

EMP needs

- **Input from other researchers !!**
  - Automate further reformulation strategies
    - More of the same, boring to some, exiting to others
    - Concurrent strategies
  - Examples from existing publications
    - EMP Library
Conclusion

EMP promotes non-traditional MP classes

- Automates symbolic reformulations to avoid error-prone and time-consuming manual algebra (re)writing
- Offers solutions through established and powerful solution engines
- Makes theoretical benefits available to users from a wide variety
- Provides nonstandard model information to solver developers → new algorithms/software?

EMP bridges the gap between academia and industry!
Thank you !

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

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