Rapid Prototyping of Decomposition Algorithms

Alexander Meeraus
Michael Bussieck
Lutz Westermann

GAMS Development Corporation
GAMS Software GmbH
www.gams.com

AMeeraus@gams.com
MBussieck@gams.com
LWestermann@gams.com

Bad Honnef, November 16, 2012
Agenda

- Introduction
- High Performance Prototypes
- Generic Algorithms
Algebraic Modeling Languages

What’s that?

http://en.wikipedia.org/wiki/Algebraic_modeling_language

• High-level computer programming languages for the formulation of complex mathematical optimization problems

• Notation similar to algebraic notation: Concise and readable definition of problems in the domain of optimization

• Do not solve problems directly, but ready-for-use links to state-of-the-art algorithms
General Algebraic Modeling System

- Roots: World Bank, 1976
- Went commercial in 1987
- GAMS Development Corporation (Washington, Houston)
- GAMS Software GmbH (Köln, Braunschweig)

- Broad academic & commercial user community and network
Monthly System Downloads

Download GAMS Distribution 23.8.1 - March 17, 2012

Note: To deliver GAMS with the best performance we are using the Amazon CloudFront web service, a global network of edge locations for content delivery.

Microsoft Internet Explorer users who have enabled SmartScreen Filter may get several warnings during the download of a GAMS system. If you do not want to proceed, please cancel the download and download the current version for Windows 32 bit or Windows 64 bit as a zip file and unzip the file before running the setup program.

Please read the release notes before downloading a system. The installation notes for Windows and UNIX and the complete system documentation are included in any

<table>
<thead>
<tr>
<th>Region</th>
<th>Monthly Downloads</th>
<th>Data Transfer Out (includes sales consumption tax).</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.120 per GB - first 10 TB / month data transfer out</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.0100 per 10,000 HTTPS Requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.0075 per 10,000 HTTP Requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.120 per GB - first 10 TB / month data transfer out</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.0120 per 10,000 HTTPS Requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.0090 per 10,000 HTTP Requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.201 per GB - first 10 TB / month data transfer out (includes sales consumption tax).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Products:
- Windows 32 bit
- Windows Vista, Windows XP, Windows Server 2003, and compatible on AMD- or Intel-based (x86_12) architectures
- Windows 64 bit
- Windows Vista, Windows Server 2008, Windows Server 2008 R2, and compatible on AMD- or Intel-based (x86_64) architectures
- Linux 32 bit
- AMD or Intel-based 32-bit Linux systems, the software was built with the GNU Compiler Collection (GCC) under, or 4.4 or higher
- Linux 64 bit
- AMD or Intel-based 64-bit Linux systems
- AIX 5.3 or higher, PowerPC chip, 64-bit or 64-}

Please consult the release notes before downloading. The installation notes for Windows and UNIX and the complete system documentation are included in any

1 Request
18,087 Requests
21.656 GB
25.41
5.41

Download GAMS Distribution 23.8.1 - March 17, 2012

Note: To deliver GAMS with the best performance we are using the Amazon CloudFront web service, a global network of edge locations for content delivery.

Microsoft Internet Explorer users who have enabled SmartScreen Filter may get several warnings during the download of a GAMS system. If you do not want to proceed, please cancel the download and download the current version for Windows 32 bit or Windows 64 bit as a zip file and unzip the file before running the setup program.

Please read the release notes before downloading a system. The installation notes for Windows and UNIX and the complete system documentation are included in any

<table>
<thead>
<tr>
<th>Region</th>
<th>Monthly Downloads</th>
<th>Data Transfer Out (includes sales consumption tax).</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.120 per GB - first 10 TB / month data transfer out</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.0100 per 10,000 HTTPS Requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.0075 per 10,000 HTTP Requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.120 per GB - first 10 TB / month data transfer out</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.0120 per 10,000 HTTPS Requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.0090 per 10,000 HTTP Requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.201 per GB - first 10 TB / month data transfer out (includes sales consumption tax).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Products:
- Windows 32 bit
- Windows Vista, Windows XP, Windows Server 2003, and compatible on AMD- or Intel-based (x86_12) architectures
- Windows 64 bit
- Windows Vista, Windows Server 2008, Windows Server 2008 R2, and compatible on AMD- or Intel-based (x86_64) architectures
- Linux 32 bit
- AMD or Intel-based 32-bit Linux systems, the software was built with the GNU Compiler Collection (GCC) under, or 4.4 or higher
- Linux 64 bit
- AMD or Intel-based 64-bit Linux systems
- AIX 5.3 or higher, PowerPC chip, 64-bit or 64-
<table>
<thead>
<tr>
<th>Year</th>
<th>Frederick W. Lanchester Prize: Winner [-hide]</th>
</tr>
</thead>
</table>

Citation: Multi-Level Planning: Case Studies in Mexico, edited by Louis M. Goreux and Alan S. Manne.
This book is a benchmark quantitative study of policy-oriented issues in a growing economy. In the modern tradition of Professor W. Leontief's input-output analysis, a team of researchers from several institutions employed advanced mathematical programming approaches to study in depth the problems of interdependency among national economic choices. This monograph on multi-level planning is impressive in its dedication to developing and testing large-scale models based on available statistical data. The team's decision a half-decade ago to give special emphasis to the agricultural and energy sectors was prophetic in anticipating many of today's critical world-wide problems. Beyond its substantial contribution to empirical analysis, the book also enhances conceptual understanding of multi-level national planning as well as demonstrates the benefits to strategic policy analysis of continuing technical innovations in operations research.
(c) Regional farmer employment accounting rows:

\[-RESr + 3 \sum_{d \in r} dFLq + \sum_{q \in r} dFLt = 0, \quad \text{each } r\]

-Regional farmer employment activity

\[\text{Sum over districts and quarters of quarterly farmer employment}\]

\[\text{Sum over districts and months of monthly farmer employment}\]

\[= 0\]

(d) Total employment accounting row in man-years:

\[-12LMAN + \sum_{t} LMAN_t = 0\]

\[-12 \left(\text{Total employment in man-years}\right) + \left(\text{Sum over months of total employment in man-months}\right) = 0\]

(e) Total monthly employment accounting rows in man-months:

\[-2.2LMAN_t + \sum_{d} dDL_t + \sum_{q} dFLq + \sum_{q} dFLt = 0, \quad \text{each } t \text{ and } q \text{ such that } t \in q\]

\[-2.2 \left(\text{Total employment in month } t\right) + \left(\text{Sum over districts of day labor employment in month } t\right)\]

\[\sum \text{Sum over districts of quarterly farmer employment in the quarter containing month } t\]

\[= 0\]
### Model Data

**Table 3**

<table>
<thead>
<tr>
<th>Cultivation month and operation</th>
<th>Mechanized Unskilled labor</th>
<th>Partially mechanized Unskilled labor</th>
<th>Partially mechanized Machinery</th>
<th>Non-mechanized Unskilled labor</th>
<th>Non-mechanized Animals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Preparatory tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fallow</td>
<td>0.12</td>
<td>0.12</td>
<td>0.5</td>
<td>1.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Cross-plowing</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harrowing</td>
<td>2.0</td>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Land levelling</td>
<td>0.25</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canal cleaning</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2nd Irrigation ditches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forming borders</td>
<td>1.0</td>
<td>1.0</td>
<td>0.2</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Linking borders</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Water application</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harrowing</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seeding and fertilization</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintenance of field works</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3rd Thinning plants</td>
<td>4.0</td>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cultivation</td>
<td>4.0</td>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weeding</td>
<td>4.0</td>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applications of insecticides (2)</td>
<td>6.0</td>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Matrix Generator
### Matrix Generator Input

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>25</th>
<th>1</th>
<th>8</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>AGGREGATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ALA</td>
<td>ALG</td>
<td>ALV</td>
<td>ARO</td>
<td>AZU</td>
<td>CHG</td>
<td>CHV</td>
<td>FRI</td>
<td>GAR</td>
<td>JIT</td>
<td>JON</td>
<td>MAI</td>
<td>MAT</td>
<td>MEL</td>
</tr>
<tr>
<td></td>
<td>PLU</td>
<td>SAL</td>
<td>SAN</td>
<td>SDR</td>
<td>SOR</td>
<td>SOT</td>
<td>SRY</td>
<td>TRI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.12</td>
<td>0.0165</td>
<td>0.0286</td>
<td>99999</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AZU</td>
<td>-0.25</td>
<td>1.0</td>
<td>0.0070</td>
<td>2827020.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JIT</td>
<td>-0.4</td>
<td>1.0</td>
<td>0.1150</td>
<td>174752.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEP</td>
<td>-0.6</td>
<td>1.0</td>
<td>0.0540</td>
<td>10.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLU</td>
<td>-1800.0</td>
<td>1.0</td>
<td>0.5770</td>
<td>85209.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHI</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHV</td>
<td>0.1500</td>
<td>1.0</td>
<td>144559</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOR</td>
<td>0.0600</td>
<td>1.0</td>
<td>2285818</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDR</td>
<td>0.0930</td>
<td>1.0</td>
<td>6665</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHG</td>
<td>0.0010</td>
<td>1.0</td>
<td>228109</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALV</td>
<td>0.0400</td>
<td>1.0</td>
<td>79019</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALA</td>
<td>0.0990</td>
<td>1.0</td>
<td>1427</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAR</td>
<td>0.0990</td>
<td>1.0</td>
<td>77997</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAI</td>
<td>0.0860</td>
<td>1.0</td>
<td>142804</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEC</td>
<td>0.0300</td>
<td>1.0</td>
<td>345979</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRI</td>
<td>0.0600</td>
<td>1.0</td>
<td>34397</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARQ</td>
<td>0.780</td>
<td>1.0</td>
<td>16850</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAP</td>
<td>0.0600</td>
<td>1.0</td>
<td>69350</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAR</td>
<td>0.0990</td>
<td>1.0</td>
<td>60158</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRA</td>
<td>2.0</td>
<td>1.0</td>
<td>12.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAI</td>
<td>0.0860</td>
<td>1.0</td>
<td>142804</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRI</td>
<td>0.0600</td>
<td>1.0</td>
<td>345979</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRU</td>
<td>0.0400</td>
<td>1.0</td>
<td>79019</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAL</td>
<td>0.0780</td>
<td>1.0</td>
<td>16850</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JON</td>
<td>0.0600</td>
<td>1.0</td>
<td>69350</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAR</td>
<td>0.0830</td>
<td>1.0</td>
<td>193910</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOY</td>
<td>0.0600</td>
<td>1.0</td>
<td>69350</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALA</td>
<td>0.02</td>
<td>1.0</td>
<td>2.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### MPS File – Column Section

<table>
<thead>
<tr>
<th>X_ASGHC2</th>
<th>B_ASGHC2</th>
<th>A_TRA</th>
<th>1.00000</th>
</tr>
</thead>
<tbody>
<tr>
<td>X_ASGHC2</td>
<td>A_TRA</td>
<td>6.96400</td>
<td></td>
</tr>
<tr>
<td>X_ASGHC3</td>
<td>D_**,GM,N</td>
<td>0.23500</td>
<td></td>
</tr>
<tr>
<td>X_ASGHC3</td>
<td>R_**,GM,C3</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASGHC3</td>
<td>B_ASGHC3</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASGHC3</td>
<td>A_TRA</td>
<td>6.96400</td>
<td></td>
</tr>
<tr>
<td>X_ASGHC3</td>
<td>A_TRA</td>
<td>6.96400</td>
<td></td>
</tr>
<tr>
<td>X_ASGHC5</td>
<td>D_**,GM,P</td>
<td>0.15500</td>
<td></td>
</tr>
<tr>
<td>X_ASGHC5</td>
<td>R_**,GHC3</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASGHC5</td>
<td>B_ASGHC5</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASGHC5</td>
<td>A_TRA</td>
<td>6.96400</td>
<td></td>
</tr>
<tr>
<td>X_ASGHCN</td>
<td>R_**,GHCN</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASGHCN</td>
<td>A_TRA</td>
<td>6.96400</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC1</td>
<td>D_**,KS,N</td>
<td>0.26500</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC1</td>
<td>R_**,KS,C1</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC1</td>
<td>B_ASKSC1</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC1</td>
<td>A_TRA</td>
<td>7.56000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC2</td>
<td>D_**,KS,N</td>
<td>0.31000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC2</td>
<td>R_**,KS,C2</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC2</td>
<td>B_ASKSC2</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC2</td>
<td>A_TRA</td>
<td>7.56000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC3</td>
<td>D_**,KS,N</td>
<td>0.33500</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC3</td>
<td>R_**,KS,C3</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC3</td>
<td>B_ASKSC3</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC3</td>
<td>A_TRA</td>
<td>7.56000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC8</td>
<td>D_**,KS,N</td>
<td>0.20600</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC8</td>
<td>R_**,KSAS</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC8</td>
<td>B_ASKSC8</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC8</td>
<td>A_TRA</td>
<td>7.56000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC9</td>
<td>D_**,KS,N</td>
<td>0.15500</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC9</td>
<td>R_**,KSAS</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC9</td>
<td>B_ASKSC9</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSC9</td>
<td>A_TRA</td>
<td>7.56000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSCN</td>
<td>R_**,KS,CN</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>X_ASKSCN</td>
<td>B_ASKSCN</td>
<td>1.00000</td>
<td></td>
</tr>
</tbody>
</table>
### MPS Revision File

<table>
<thead>
<tr>
<th>BRANCH</th>
<th>NEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>REVISE</td>
<td>REVS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HEADERS</th>
<th>CARD NO.</th>
<th>NAME</th>
<th>COLUMNS</th>
<th>MODIFICATION</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Card Read Summary**

<table>
<thead>
<tr>
<th>CARD NO.</th>
<th>NAME</th>
<th>COLUMNS</th>
<th>MODIFICATION</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The table above represents the structure and content of the MPS Revision File, detailing the branching, next steps, revisions, and summary data for each header card.
### MPS Output

**DATE 07/10/76 TIME 22:12:21**

**PRINT OPTION** = COMPLETE OUTPUT w/SPECIAL

**NAME** = CENTRAL **INC** = 0 **RHS** = 0

**DIR** = MAXIMIZE **CJOB** = 12

**APX** = III 1,000 PAGE

<table>
<thead>
<tr>
<th>NUMBER</th>
<th>NAME</th>
<th>TYPE</th>
<th>STATUS</th>
<th>COL ACTIVITY</th>
<th>OBJ COEF</th>
<th>D UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>GCR.1</td>
<td>PL</td>
<td>LOWER</td>
<td></td>
<td>-97,40000</td>
<td>INF</td>
<td>6,44851</td>
</tr>
<tr>
<td>102</td>
<td>GCR.2</td>
<td>PL</td>
<td>ACTIVE</td>
<td>00007</td>
<td>-701,0000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>GCR.3</td>
<td>PL</td>
<td>ACTIVE</td>
<td>00330</td>
<td>-2429,7000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>GCR.4</td>
<td>PL</td>
<td>LOWER</td>
<td></td>
<td>-9410,0000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>GCR.5</td>
<td>PL</td>
<td>LOWER</td>
<td></td>
<td>-5110,0000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>GCR.6</td>
<td>PL</td>
<td>ACTIVE</td>
<td>06067</td>
<td>-13,2000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>GCR.7</td>
<td>PL</td>
<td>ACTIVE</td>
<td>00226</td>
<td>-231,57000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>GCR.8</td>
<td>PL</td>
<td>ACTIVE</td>
<td>06002</td>
<td>70,7100</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>GCR.9</td>
<td>PL</td>
<td>ACTIVE</td>
<td>00045</td>
<td>70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>GCR.10</td>
<td>PL</td>
<td>ACTIVE</td>
<td>07685</td>
<td>70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>GCR.11</td>
<td>PL</td>
<td>ACTIVE</td>
<td>180,74000</td>
<td>70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>GCR.12</td>
<td>PL</td>
<td>ACTIVE</td>
<td>09963</td>
<td>70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>GCR.13</td>
<td>PL</td>
<td>ACTIVE</td>
<td>09226</td>
<td>70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>GCR.14</td>
<td>PL</td>
<td>ACTIVE</td>
<td>09660</td>
<td>70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>GCR.15</td>
<td>PL</td>
<td>ACTIVE</td>
<td>07400</td>
<td>70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>GCR.16</td>
<td>PL</td>
<td>ACTIVE</td>
<td>08000</td>
<td>70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>GCR.17</td>
<td>PL</td>
<td>LOWER</td>
<td></td>
<td>-70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>GCR.18</td>
<td>PL</td>
<td>ACTIVE</td>
<td>09963</td>
<td>-70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>GCR.19</td>
<td>PL</td>
<td>ACTIVE</td>
<td>09226</td>
<td>-70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>GCR.20</td>
<td>PL</td>
<td>ACTIVE</td>
<td>09660</td>
<td>-70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>GCR.21</td>
<td>PL</td>
<td>ACTIVE</td>
<td>07400</td>
<td>-70,71000</td>
<td>INF</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>GCR.22</td>
<td>PL</td>
<td>ACTIVE</td>
<td>08000</td>
<td>-70,71000</td>
<td>INF</td>
<td></td>
</tr>
</tbody>
</table>

---
PLANNING PROBLEM AND OBJECTIVES INITIALLY OFTEN

UNSTRUCTURED
ILL-DEFINED
CONFLICTING
UNCERTAIN
CHANGING
EMOTIONAL

MATHEMATICAL MODEL USED TO RECOGNIZE AND FORMULATE PROBLEMS, DEFINE ISSUES AND EXPLORE SOLUTION SPACE
RESULT: - Drain of resources (technical, time, money)
- Essentially no documentation

SOLUTION

ANALYST

MODEL

DATA

PRESENT TECHNOLOGY

REAL WORLD PROBLEM

Operating Systems
Model Generators
Computer Languages
Report Generators
Data Base Systems
Solution Packages
MAJOR CONSTRAINTS: COST
SKILLS
TIME
TOOLS
DOCUMENTATION
TRUST
•
•
•
RESULT:  - Limited drain of resources
         - Same representation of models for humans and machines
         - Model representation is also model documentation
ON THE DEVELOPMENT OF A GENERAL ALGEBRAIC MODELING SYSTEM IN A STRATEGIC PLANNING ENVIRONMENT*

Johannes BISSCHOP** and Alexander MEERAUS

Development Research Center, The World Bank, Washington, DC 20433, U.S.A.

Received 18 March 1980
Revised manuscript received 8 May 1981

Modeling activities at the World Bank are highlighted and typified. Requirements for successful modeling applications in such a strategic planning environment are examined. The resulting development of a General Algebraic Modeling System (GAMS) is described. The data structure of this system is analyzed in some detail, and comparisons to other modeling systems are made. Selected aspects of the language are presented. The paper concludes with a case study of the Egyptian Fertilizer Sector in which GAMS has been used as a modeling tool.

Key words: Algebraic Modeling System, Modeling Language, Strategic Planning, Applications.
GAMS’ Fundamental concepts

- Platform independence
  - Hassle-free switch of solution methods
  - Open architecture and interfaces to other systems
  - Balanced mix of declarative and procedural elements

10+ Supported Platforms

- Solaris 64bit
- Solaris
- AXU
- AIX
- Linux 64bit
- Windows
- Linux
- HP
- Mac
GAMS’ Fundamental concepts

- Platform independence
- Hassle-free switch of solution methods
- Open architecture and interfaces to other systems
- Balanced mix of declarative and procedural elements

25+ Integrated Solvers
GAMS’ Fundamental concepts

- Platform independence
- Hassle-free switch of solution methods
- **Open architecture and interfaces to other systems**
- Balanced mix of declarative and procedural elements

Binary Data Exchange

- Fast exchange of data
- Syntactical check on data before model starts
- Data Exchange at any stage (Compile and Run-time)
- Platform Independent
- Direct GDX interfaces and general API
- Scenario Management Support
- Full Support of Batch Runs
GAMS’ Fundamental concepts

- Platform independence
- Hassle-free switch of solution methods
- Open architecture and interfaces to other systems
- Balanced mix of declarative and procedural elements

Declaration of...
- Sets
- Parameters
- Variables
- Equations
- Models
- ...

Procedural Elements like…
- loops
- if-then-else
- ...

22
Solver Prototypes in GAMS

- DICOPT (Grossmann)
- BARON (Sahinidis)
- SBB (Drud)
- NLPEC
- …
- EMP
New Modeling and Solution Concepts

• Breakouts of traditional MP classes
  – Extended Nonlinear Programs
  – Chance Constraints
  – CVaR Constraints
  – Robust Programming
  – Bilevel Programs
  – Generalized Disjunctive Programs
  – Multi Agent Equilibrium

• Limited support with common model representation
• No conventional syntax
• 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, structure, and semantics
• Express extended model in symbolic (source) form and apply existing modeling/solution technology
• Package new tools with the production system

⇒ Extended Mathematical Programming (EMP)
JAMS: a GAMS EMP Solver

- EMP Information
- Original Model

Translation

Reformulated Model

Solving using established Algorithms

Solution

Viewable

Mapping Solution into original space
EMP Library

- Distributed with GAMS
Agenda

- Introduction
- High Performance Prototypes
- Generic Algorithms
Simple Transport Model

Sets
i factories /f1*f3/  
j distribution centers /d1*d5/

Parameter
capacity(i) /f1 500, f2 450, f3 650/  
demand(j) /d1 160, d2 120, d3 270, d4 325, d5 700/  
prodcost unit production cost /14/  
price sales price /24/  
wastecost cost of removal of overstocked products /4/

Table transcost(i,j) unit transportation cost

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1</td>
<td>2.49</td>
<td>5.21</td>
<td>3.76</td>
<td>4.85</td>
<td>2.07</td>
</tr>
<tr>
<td>f2</td>
<td>1.46</td>
<td>2.54</td>
<td>1.83</td>
<td>1.86</td>
<td>4.76</td>
</tr>
<tr>
<td>f3</td>
<td>3.26</td>
<td>3.08</td>
<td>2.60</td>
<td>3.76</td>
<td>4.45</td>
</tr>
</tbody>
</table>
Simple Transport Model – Cont.

Variables

- \text{ship}(i,j) \quad \text{shipments}
- \text{product}(i) \quad \text{units produced}
- \text{received}(j) \quad \text{unit received}
- \text{sales}(j) \quad \text{sales (actually sold)}
- \text{waste}(j) \quad \text{overstocked products}

Positive variables ship, product, sales, waste;

Equations

- \text{obj}
- \text{production}(i)
- \text{receive}(j)
- \text{selling}(j)
- \text{market}(j);

\begin{align*}
\text{obj.. profit} &= \sum_j \text{price} \cdot \text{sales}(j) - \sum_{(i,j)} \text{transcost}(i,j) \cdot \text{ship}(i,j) \\
&\quad - \sum_j \text{wastecost} \cdot \text{waste}(j) - \sum_i \text{prodcost} \cdot \text{product}(i);
\end{align*}

\begin{align*}
\text{production}(i) &= \sum_j \text{ship}(i,j); \\
\text{product.up}(i) &= \text{capacity}(i); \\
\text{receive}(j) &= \sum_i \text{ship}(i,j); \\
\text{selling}(j) &= \text{sales}(j) = \text{received}(j) - \text{waste}(j); \\
\text{market}(j) &= \text{sales}(j) = 1 \cdot \text{demand}(j); \\
\end{align*}

\text{model transport /all/;}
\text{solve transport maximizing profit using lp;}
Benders Decomposition for 2-Stage SP

Set
\( s \) scenarios /lo, mid, hi/

* Stochastic demand plus probabilities

Table ScenarioData(s,*)

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo</td>
<td>150</td>
<td>100</td>
<td>250</td>
<td>300</td>
<td>600</td>
<td>0.25</td>
</tr>
<tr>
<td>mid</td>
<td>160</td>
<td>120</td>
<td>270</td>
<td>325</td>
<td>700</td>
<td>0.5</td>
</tr>
<tr>
<td>hi</td>
<td>170</td>
<td>135</td>
<td>300</td>
<td>350</td>
<td>800</td>
<td>0.25</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\min c^T x + \sum_{\omega} p(\omega) d^T_{\omega} y_{\omega} \\
Ax &= b \\
T_\omega x + W_\omega y_\omega &= h_\omega \\
x &\geq 0, y_\omega &\geq 0 \\
\min d^T_{\omega} y_{\omega} \\
W_\omega y_\omega &= h_\omega - T_\omega x^\nu \\
y_\omega &\geq 0 \\
\theta &\geq \sum_{\omega \in \Omega} p_\omega \left(-\pi^\ell_\omega [T_\omega x + W_\omega y^\ell_\omega - h_\omega]\right), \ell = 1, \ldots, \nu - 1 \\
x &\geq 0
\end{align*}
\]
Benders Decomposition for 2-Stage SP

Based on paper by E. Kalvelagen

```
masterobj..
    zmaster =e= theta -sum((i,j), transcost(i,j)*ship(i,j))
       - sum(i, prodcost*product(i));

receive(j)..
    received(j) =e= sum(i, ship(i,j));

production(i)..
    product(i) =e= sum(j, ship(i,j));
    product.up(i) = capacity(i);

optcut(dyniter)..
    theta =l= cutconst(dyniter) + sum(j, cutcoeff(dyniter,j)*received(j));

subobj..
    zsub =e= sum(j, price*sales(j)) - sum(j, wastecost*waste(j));

selling(j)..
    sales(j) + waste(j) =e= received.l(j);

market(j)..
    sales(j) =l= demand(j);
```
Benders Decomposition for 2-Stage SP

Based on paper by E. Kalvelagen

```gams
loop(iter$(not done),
  * solve subproblems
  dyriter(iter) = yes;
  loop(s,
    demand(j) = ScenarioData(s,j);
    solve subproblem max zsub using lp;
    objsub(s) = zsub.l;
    cutconst(iter) = cutconst(iter) + p(s)*sum(j,market.m(j)*demand(j));
    cutcoeff(iter,j) = cutcoeff(iter,j) + p(s)*selling.m(j);
  );
  lowerbound = max(lowerbound, objmaster + sum(s, p(s)*objsub(s)));

  * convergence test
  if( (upperbound-lowerbound) < 0.001*(1+abs(upperbound)),
    done = 1;
  else
  * solve masterproblem
    solve masterproblem max zmaster using lp;
    upperbound = zmaster.l;
    objmaster = zmaster.l - theta.l;
  );
);```

→ t_gams.gms
Benders GAMS Implementation

- GAMS Implementation (solver Cplex)
  - 17 iterations: \((3+1) \times 17 = 68\) small models
- 6.1 secs (all default)
- 5.8 secs (minimize listing file size)
- 4.6 secs (GAMS stays in memory)
- 0.5 secs (communicate with solver through memory)

```plaintext
option limrow=0, limcol=0, solprint=silent,
     solvelink=%Solvelink.LoadLibrary%;
```

- Grid computing
- Smart update of sub-model (Scenario Solver/GUSS)
- Object Oriented API (e.g., .NET)
subproblem.solvelink = %Solvelink.AsyncGrid%;

loop(s,
    demand(j) = sDemand(s,j);
    solve subproblem max zsub using lp;
    h(s) = subproblem.handle;
);

Repeat
    loop(s$handlecollect(h(s)),
        objsub(s) = zsub.l;
        display$hhandledelete(h(s)) 'trouble deleting handles';
        h(s) = 0 );
    display$sleep(card(h)*0.02) 'sleep for some time';
until card(h)=0;
Parallel Power – GAMS Grid Facility

```gams
loop(iter$(not done),
  * solve subproblems
  dyniter(iter) = yes;
* Submission loop
  loop(s,
    demand(j) = ScenarioData(s,j);
    solve subproblem max zsub using lp;
    h(s) = subproblem.handle;
  );
* Collection loop
repeat
  loop(s$handlecollect(h(s)),
    objsub(s) = zsub.l;
    cutconst(iter) = cutconst(iter) + p(s)*sum(j,market.m(j)*sDemand(s,j));
    cutcoeff(iter,j) = cutcoeff(iter,j) + p(s)*selling.m(j);
    display$handledelete(h(s)) 'trouble deleting handles';
    h(s) = 0 ;
    display$sleep(card(h)*0.02) 'was sleeping for some time';
  until card(h) = 0;
lowerbound = max(lowerbound, objmaster + sum(s, p(s)*objsub(s)))
);
* convergence test
```
GUSS: Gather-Update-Solve-Scatter

market(j).. sales(j)=l= demand(j);
subobj.. zsub =e= sum(j, price*sales(j)) - sum(j, wastecost*waste(j));

loop(s,
    demand(j) = sDemand(s,j);
    solve subproblem max zsub using lp;
    objsub(s) = zsub.l;
);

set dict / s. scenario. ''
    demand. param. sDemand
    zsub. level. Objsub /
solve subproblem max zsub using lp scenario dict;
GUSS: Gather-Update-Solve-Scatter

* GUSS setup
Set dict / s. scenario. ''
demand. param. sDemand
market. marginal. sMarket
selling. marginal. sSelling
zsub. level. objsub /

loop(iter$(not done),
   * solve subproblems
      dyniter(iter) = yes;
solve subproblem max zsub using lp scenario dict;
cutconst(iter) = cutconst(iter) + sum(s,p(s)*sum(j,sMarket(s,j)*sDemand(s,j)));
cutcoeff(iter,j) = cutcoeff(iter,j) + sum(s,p(s)*sSelling(s,j));

   lowerbound = max(lowerbound, objmaster + sum(s, p(s)*objsub(s)));

   * convergence test
      if( (upperbound-lowerbound) < 0.001*(1+abs(upperbound)),
         done = 1;
      else
         * solve masterproblem
            solve masterproblem max zmaster using lp;
            upperbound = zmaster.l;
            objmaster = zmaster.l - theta.l;
      );

=> t_guss.gms
GUSS: Gather-Update-Solve-Scatter

- 100 scenarios, 38 iterations: \((100+1)*38 = 3838\) models

<table>
<thead>
<tr>
<th>Setting</th>
<th>Solve time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solvelink=0 (default)</td>
<td>294.36</td>
</tr>
<tr>
<td>Solvelink=%Solvelink.CallModule%</td>
<td>239.57</td>
</tr>
<tr>
<td>Solvelink=%Solvelink.LoadLibrary%</td>
<td>16.21</td>
</tr>
<tr>
<td>GUSS</td>
<td>9.18</td>
</tr>
</tbody>
</table>

- Updates model data instead of matrix coefficients/rhs
- Hot start (keeps the model hot inside the solver and uses solver’s best update mechanism)
- Saves model generation and solver setup time
- A priori knowledge of all scenario data
Calling GAMS from your Application

Creating Input for GAMS Model
→ Data handling using GDX API

Callout to GAMS
→ GAMS option settings using Option API
→ Starting GAMS using GAMS API

Reading Solution from GAMS Model
→ Data handling using GDX API
Low level APIs → Object Oriented API

- Low level APIs
  - GDX, OPT, GAMSX, GMO, …
  - High performance and flexibility
  - Automatically generated imperative APIs for several languages (C, Delphi, Java, Python, C#, …)

- Object Oriented GAMS API
  - Additional layer on top of the low level APIs
  - Object Oriented
  - Written by hand to meet the specific requirements of different Object Oriented languages
Features of the object oriented API

- No modeling capability, model is still written in GAMS.

- Prepare input data and retrieve results in a convenient way → \textit{GAMSDatabase}.

- Control GAMS execution → \textit{GAMSJob}.

- Scenario Solving: Feature to solve multiple very similar models in a dynamic and efficient way. → \textit{GAMSModellInstance}.

- Seamless integration of GAMS into other programming environments.
GAMSMModelInstance etc.

**GAMSJob**
- Manages the execution of a GAMS program given by GAMS model source

**GAMSCheckpoint**
- Captures the state of a GAMSJob

**GAMSMModelInstance**
- A single mathematical model generated by a GAMS solve statement

**GAMSMModifier**
- Marks elements of a GAMSMModelInstance to be modifiable
do
{
    masteri.Solve(GAMSModelInstance.SymbolUpdateType.BaseCase);
    if (1 < iter)
        upperbound = masteri.SyncDB.GetVariable("zmaster").FirstRecord().Level;
    objmaster = masteri.SyncDB.GetVariable("zmaster").FirstRecord().Level - theta.FirstRecord().Level;
    received.Clear();
    foreach (GAMSVariableRecord r in masteri.SyncDB.GetVariable("received")) {
        received.AddRecord(r.Keys.Value = r.Level;
        cutcoeff.AddRecord(iter.ToString(), r.Keys[0]);}
    cutconst.AddRecord(iter.ToString());
    double objsub = 0.0;
    foreach (GAMSSetRecord s in data.OutDB.GetSet("s"))
    {
        demand.Clear();
        foreach (GAMSSetRecord j in data.OutDB.GetSet("j")
            demand.AddRecord(j.Keys.Value = scenarioData.FindRecord(s.Keys[0], j.Keys[0]).Value;
        subi.Solve(GAMSModelInstance.SymbolUpdateType.BaseCase);
        double probability = scenarioData.FindRecord(s.Keys[0], "prob").Value;
        objsub += probability * subi.SyncDB.GetVariable("zsub").FirstRecord().Level;
        foreach (GAMSSetRecord j in data.OutDB.GetSet("j"))
        {
            cutconst.FindRecord(iter.ToString()).Value += probability *
            cutcoeff.FindRecord(iter.ToString(), j.Keys[0]).Value += probability *
            subi.SyncDB.GetEquation("selling").FindRecord(j.Keys).Marginal;}
    }
    lowerbound = Math.Max(lowerbound, objmaster + objsub);
    iter++;
    if (iter == maxiter + 1)
        throw new Exception("Benders out of iterations");
} while ((upperbound - lowerbound) >= 0.001 * (1 + Math.Abs(upperbound)));
### Object Oriented GAMS API

<table>
<thead>
<tr>
<th>Setting</th>
<th>Solve time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solvelink=0 (default)</td>
<td>294.36</td>
</tr>
<tr>
<td>Solvelink=%Solvelink.CallModule%</td>
<td>239.57</td>
</tr>
<tr>
<td>Solvelink=%Solvelink.LoadLibrary%</td>
<td>16.21</td>
</tr>
<tr>
<td>GUSS</td>
<td>9.18</td>
</tr>
<tr>
<td>C# Application</td>
<td>2.43</td>
</tr>
</tbody>
</table>
Excursus: SP with EMP

```plaintext
model transport /all/;
solve transport maximizing profit using lp;

file emp /'temp.info%'/; put emp '* problem %gams.i%'/;

jrandvar demand('d1') demand('d2') demand('d3') demand('d4') demand('d5')
  0.25  150  100  250  300  600  
  0.5   160  120  270  325  700  
  0.25  170  135  300  350  800  

stage2 demand sales waste profit obj selling market

putclose emp;

Set scen Scenarios / s1*s3 /;
Parameter
  sc_demand(scen,j) demand by scenario
  sc_sales(scen,j)  sales by scenario
  sc_waste(scen,j)  waste by scenario
  sc_profit(scen)  profit by scenario;

Set dict / scen .scenario.
  demand .randvar. sc_demand
  sales .level. sc_sales
  waste .level. sc_waste
  profit .level. sc_profit /;

option emp=de;
solve transport maximizing profit using emp scenario dict;
display ship.l, sc_demand, sc_sales, sc_waste, sc_profit;
```

 ➔ t_emp.gms
ACSOM

- Advanced Collaborative System Optimization Modeler
- Explore high-dimensional Pareto surface for configuration of (expensive) military vehicles
- Collaboration between:
  - General Dynamics Land Systems (GDLS)
  - Industrial & Systems Engineering, Wayne State University
  - GAMS Development Corp
Abrams M1A2

- Remote weapons station
- Loader's Armor Gun Shield
- Tank/infantry telephone
- Rear protecting unit slat armor
- Abrams Reactive Armor Tiles
The Whole-system Design Problem

21 SUBSYSTEMS

Options per subsystem | Theoretically Possible Subsystem Combinations
---|---
2 | 2,092,152
3 | 10,460,353,203
Considers All to Make the Whole
Full Spectrum

Numerous Subsystems with a multitude of options for each

Force Protection Approach?
Auto loader?
- Configuration/Approach?

Two, Three, Four Crew Members?

Hull Material?
- Aluminum?
- Steel?
- Titanium?

Which Core Data Network?

Hull Design?

Power?
- What Type of Engine?
- What Type Transmission?

What Type of Suspension System?
- Torsion bars?
- Passive?
- HSUs?
- Track?
- Fully Active?

Which Servo Motor Controller Architecture?

Approved for Public Release, Distribution Unlimited, GDLS approved, log 2008-09, dated 03/19/08
• Suite of Algorithms
  – Complete Enumeration
  – Acsom 1.2 algorithm
  – Integer Cut
  – WSU algorithms: ACE, UPSA

• Developed in
  – VB.Net
  – GAMS
    • GAMS Models and Algorithms
    • .NET API for recursive ACE algorithm
  – Cplex MIP Solver
  – MySQL
Aims For GAMS Models

• High Performance
• Use multiple cores
  – For current crop of laptops
  – For high-performance machines
  – Be prepared for advancing technologies (more parallelism)
• Focus on whole system
  – High level modeling language can help improve design
  – Although slower than traditional programming languages, by using better design we can achieve higher performance
  – Examples:
    • Use parallel instances of scenario solver where possible
    • Use bulk SQL in one spot instead of record level SQL scattered over application
Reimplementation of Acsom 1.2 algorithm

Old system:
- VB.Net
- MPL (single point)
- Cplex
- SQL throughout

New System:
- GAMS (complete algorithm; streamlined design)
- Multiple parallel instances of scenario solver
- Cplex
- Bulk SQL just once

Performance improvement
- On the same hardware:
  - From hours and even days for large problems
  - To minutes, up to an hour
Test Environment
Parallel GAMS jobs (2)


For a different client we needed to run a randomized algorithm that solves many small MIP models. They are so small that using multiple threads inside the MIP solver does not give much performance boost (much of the time is spent outside the pure Branch & Bound part - such as preprocessing etc.). However as the MIP problems are independent of each other we could generate all the necessary data in advance and then call the scenario solver ([http://www.gams.com/modlib/adddocs/gusspaper.pdf](http://www.gams.com/modlib/adddocs/gusspaper.pdf)). This will keep the generated problem in memory, and does in-core updates, so we don't regenerate the model all the time.

When running the algorithm with n=100,000 MIP models we see the following performance. Note that besides the MIP models there is also a substantial piece of GAMS code that implements other parts of the algorithm.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>number of MIP models</th>
<th>solve time</th>
<th>rest of algorithm</th>
<th>total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional GAMS loop (call solver as DLL)</td>
<td>100,000</td>
<td>1068 sec</td>
<td>169 sec</td>
<td>1237 sec</td>
</tr>
<tr>
<td>Scenario Solver</td>
<td>100,000</td>
<td>293 sec</td>
<td>166 sec</td>
<td>459 sec</td>
</tr>
</tbody>
</table>

To get more performance I tried to run the scenario solver in parallel. That is not completely trivial as the solver has a number glitches (e.g. scratch files with fixed, hard coded names). I also run parts of the GAMS algorithm in parallel, but some parts had to be done in the master model after merging the results.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>number of MIP models</th>
<th>Worker threads</th>
<th>parallel sub-problem time</th>
<th>rest of algorithm (serial)</th>
<th>total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel + Scenario Solver</td>
<td>100,000</td>
<td>4</td>
<td>116 sec</td>
<td>67 sec</td>
<td>183 sec</td>
</tr>
</tbody>
</table>

The implementation does not win the beauty contest, but it could be developed quickly. For these larger...
Agenda

- Introduction
- High Performance Prototypes
- Generic Algorithms
Indexed Model – Scalar Model

- **Model** = Model Instance

- Indexed Models
  - Algebraic Modeling Systems were invented to support this type of modeling

- Scalar Models:
  - Variables \( x(1), x(2), x(3), \ldots \)
  - Equations \( e(1), e(2), e(3), \ldots \)
  - Algorithm development
  - Automatic translation of model instance into other AML scalar: GAMS/Convert (AMPL, Lingo, Minopt, \ldots)
### Model Translation

#### GAMS Code

```
MODEL:
[Obj]   var x1 >= 0;
        var x2 >= 0;
        [e1]   var x3 > 0;
        var x4 > 0;
        var x5 > 0;
        var x6 > 0;
        [e2]   minimize + 0;
        [e3]   subject to e1 ..
                - 0.225*x1 - 0.153*x2 - 0.162*x3 - 0.225*x4 - 0.162*x5 - 0.126*x6 + x7 =E= 0;
        [e4]   e2 ..
                x1 + x2 + x3 =L= 350;
        [e5]   e3 ..
                x4 + x5 + x6 =L= 600;
        [e6]   e4 ..
                x1 + x4 =G= 325;
        [e7]   e5 ..
                x2 + x5 =G= 300;
        [e8]   e6 ..
                x3 + x6 =G= 275;

Solve tr using LP minimizing x7;
```

---

**Variables**

- \( x_1, x_2, x_3, x_4, x_5, x_6, x_7 \)
- \( x_1, x_2, x_3, x_4, x_5, x_6 \)

**Positive Variables**

- \( x_1, x_2, x_3, x_4, x_5, x_6 \)

**Equations**

- \( e_1 \): minimize + 0
- \( e_2 \): \( x_1 + x_2 + x_3 =L= 350 \)
- \( e_3 \): \( x_4 + x_5 + x_6 =L= 600 \)
- \( e_4 \): \( x_1 + x_4 =G= 325 \)
- \( e_5 \): \( x_2 + x_5 =G= 300 \)
- \( e_6 \): \( x_3 + x_6 =G= 275 \)

---

**Model Translation**

- No execution
- No file creation, i.e., server
GAMS/Convert

- Mapping Between Indexed and Scalar Model
  - DictMap:
  - Access to Gradient
    - Jacobian/Hessian

- For example:
  - Non-linear
  - OA Cut at 3

```
C:\Users\bus.sieclc\Desktop\b~ honnef\dictmap.gdx
~\[~I-a.I
C:\Users\bussieck\Desktop\bad honnef\jacobian.gdx
```

<table>
<thead>
<tr>
<th>Entry</th>
<th>Symbol</th>
<th>Type</th>
<th>Dim</th>
<th>Nr Elem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>i</td>
<td>Set</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>j</td>
<td>Set</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>cost_EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>supply_EM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Entry</th>
<th>Symbol</th>
<th>Type</th>
<th>Dim</th>
<th>Nr Elem</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>demand_EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>x_VM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>z_VM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol search</th>
<th>demand_EM(*, *)</th>
<th>Map for equation demand</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Entry</th>
<th>Symbol</th>
<th>Type</th>
<th>Dim</th>
<th>Nr Elem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>i</td>
<td>Set</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>j</td>
<td>Set</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>jobg</td>
<td>Set</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>objcof</td>
<td>Par</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>e</td>
<td>Equ</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>x</td>
<td>Var</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol search</th>
<th>A(i, j): Jacobian</th>
</tr>
</thead>
</table>

```
Research Applications

• Tom Rutherford:
  – EPEC:
    Equilibrium Problems with Equilibrium Constraints

• Francisco Trespalacios, Ignacio Grossmann:
  – Basic Step for Disjunctive (Non-linear) Programs

---

1. RF back.
2. LF back and side (left).
3. RF side & forward
4. LF forward.
The Problem of the Day

Mixed Integer Nonlinear Program (MINLP)

\[
\begin{align*}
\text{minimize} & \quad f(x, y) \\
\text{subject to} & \quad c(x, y) \leq 0 \\
& \quad x \in X, \; y \in Y \text{ integer}
\end{align*}
\]

- $f, c$ smooth (convex) functions
- $X, Y$ polyhedral sets, e.g. $Y = \{y \in [0, 1]^p \mid Ay \leq b\}$
- $y \in Y$ integer $\Rightarrow$ hard problem
- $f, c$ not convex $\Rightarrow$ very hard problem
Outer Approximation (Duran and Grossmann, 1986)

**Motivation:** avoid solving huge number of NLPs
- Exploit MILP/NLP solvers: decompose integer/nonlinear part

**Key idea:** reformulate MINLP as MILP (implicit)
- Solve alternating sequence of MILP & NLP

NLP subproblem $y_j$ fixed:

\[
\begin{aligned}
\text{minimize} & \quad f(x, y_j) \\
\text{subject to} & \quad c(x, y_j) \leq 0 \\
& \quad x \in X
\end{aligned}
\]

Main Assumption: $f$, $c$ are convex
Outer Approximation (Duran and Grossmann, 1986)

- let \((x_j, y_j)\) solve NLP\((y_j)\)
- linearize \(f, c\) about \((x_j, y_j) = z_j\)
- new objective variable \(\eta \geq f(x, y)\)
- MINLP \((P) \equiv MILP (M)\)

\[
\begin{align*}
(M) \quad \text{minimize} & \quad \eta \\
\text{subject to} & \quad \eta \geq f_j + \nabla f_j^T (z - z_j) \quad \forall y_j \in Y \\
& \quad 0 \geq c_j + \nabla c_j^T (z - z_j) \quad \forall y_j \in Y \\
& \quad x \in X, y \in Y \text{ integer}
\end{align*}
\]

**SNAG**: need all \(y_j \in Y\) linearizations
Outer Approximation (Duran and Grossmann, 1986)

\((M_k): \) lower bound (underestimate convex \(f, c\))

\(\text{NLP}(y_j): \) upper bound \(U\) (fixed \(y_j\))

\[\rightarrow \text{stop, if lower bound} \geq \text{upper bound} \]
Simple OA Implementation in GAMS

- **Simple**: Convex + Binary variable only
- User help to identify
  - Non-linear constraints
  - Binary variables
- OA Implementation:
  - Scalar Model for (r)MINLP
    - Solve with fixed binary variables
    - Get gradients for OA cuts (Jacobian)
  - Scalar Model for MIP (lower bound)
    - Add OA Cuts
    - Cut off discrete solutions (MIP cuts)
  - Rewrite GAMS source code inserts for scalar model
Tools for Algorithm Development

- Matrix tools:
  - Cholesky
  - Eigenvalue
  - Eigenvector
  - Invert

- Others
  - Gdxrank (sorting)
  - csdp (SDP Solver)
Summary

- Rapid Prototype Development
  - GAMS Language Elements
  - High-performance (GUSS, Grid, …)
  - Other Execution Systems (.NET API)
  - Extended Mathematical Programming (EMP)
- Scalar Models plus Toolbox to Build Algorithms Independent of a Particular Index Model
<table>
<thead>
<tr>
<th>USA</th>
<th>Europe</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAMS Development Corp.</td>
<td>GAMS Software GmbH</td>
</tr>
<tr>
<td>1217 Potomac Street, NW</td>
<td>Eupener Str. 135-137</td>
</tr>
<tr>
<td>Washington, DC 20007</td>
<td>50933 Cologne</td>
</tr>
<tr>
<td>USA</td>
<td>Germany</td>
</tr>
</tbody>
</table>

Phone: +1 202 342 0180
Fax: +1 202 342 0181

http://www.gams.com
sales@gams.com
support@gams.com

http://www.gams.com
info@gams.de
support@gams.com