Deploying GAMS Models with GAMS MIR0
(Technology Workshop)

Franz Nelissen & Frederik Proske
Agenda

GAMS at a Glance

Model Development and Model Deployment

GAMS MIRO Demo
Algebraic Modeling Languages (AML)

What’s that?

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

➢ **Notation similar to algebraic notation**: Concise and human 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 (solver)
What did this give us?

**Simplified** model development & maintenance

**Increased** productivity tremendously

Made mathematical optimization available to a **broader audience** (domain experts)
One of the success stories of OR!

- **1976:** Initial version of an AML
- **2012:** INFORMS Impact Prize awarded to “Godfathers” of Algebraic Modeling Languages
- **Nowadays:**
  - Established environment to build robust and fail safe systems
  - Commodity in a lot of different flavors
Foundation of GAMS

Powerful algebraic modeling language

Open architecture, independent layers

Evolved and Matured System
Declarative Language

- Similar to mathematical notation
- Easy to learn, only few basic language elements: sets, parameters, variables, equations, models

Indices:

- $i = \text{plants}$
- $j = \text{markets}$

Given Data:

- $a_i = \text{supply of commodity of plant } i \text{ (in cases)}$
- $b_j = \text{demand for commodity at market } j$
- $c_{ij} = \text{cost per unit shipment between plant } i \text{ and market } j$

Decision Variables:

- $x_{ij} = \text{amount of commodity to ship from plant } i \text{ to market } j$
  - $x_{ij} \geq 0$, for all $i, j$

Constraints:

- Observe supply limit at plant $i$: $\sum_j x_{ij} \leq a_i$ for all $i$ (cases)
- Satisfy demand at market $j$: $\sum_i x_{ij} \geq b_j$ for all $j$ (cases)

Objective Function: Minimize $\sum_i \sum_j c_{ij} x_{ij}$ ($\leq K$)

Sets

- $i$ canning plants / seattle, san-diego /
- $j$ markets / new-york, chicago, topeka / ;

Parameters

- $a(i) = \text{capacity of plant } i \text{ in cases}$
- $b(j) = \text{demand at market } j \text{ in cases}$
- $c(i,j) = \text{transport cost in thousands of dollars per case}$

Variables

- $x(i,j) = \text{shipment quantities in cases}$
- $z = \text{total transportation costs in thousands of dollars}$

Equations

- cost: define objective function
- supply(i): observe supply limit at plant $i$
- demand(j): satisfy demand at market $j$

Cost: $z = \sum((i,j), c(i,j) * x(i,j))$

Supply: $\sum(j, x(i,j)) = a(i)$

Demand: $\sum(i, x(i,j)) \leq b(j)$

Model transport /all/ ;
Procedural Language Elements

- **Control Flow Statements** (e.g., loops, for, if, macros, functions, ...)
- **Build complex problem algorithms within GAMS**
- **Simplified interaction with other systems through OO-APIs:**
  - Data exchange
  - GAMS Control

```plaintext
* Doane 7 6 transport_async Stream 4 syncpnr.out syncpnr syncpnr.out 7 syncpnr.out 7
64 Model transport /all/;
65
66 set s scenarios / s1*s100 /;
67 sl solvelink / aSyncGrid, aSyncThreads /;
68 parameter dd(s,i,j) distance by scenario;
69 time(*) time for 100 scenarios;
70 sl_val(sl) solvelink value / aSyncGrid % solvelink aSyncThreads % solvelink
71
72 scalar h(s), scenario handle;
73
dd(s,i,j) = uniform(0.9,1.1)*d(i,j);
75 option limrow=0, limcol=0, solprint=silent, lp=cplexd;
77 * Async SOLVE
78 loop(sl, tmp = jnow;
79 transport.solvelink = sl_val(sl);
80 loop(s, d(i,j) = dd(s,i,j);
81 solve transport using lp minimizing z;
82 h(s) = transport.handle; // save instance handle
83 );
84 repeat
85 display$readycollect(h) 'Waiting for next instance to complete';
86 loop(sshandlecollect(h(s)),
87 display$handledelete(h(s)) 'trouble deleting handles';
88 h(s) = 0; // indicate that we have loaded the solution
89 );
90 until card(h) = 0 or time$elapsed > 180; // wait until all models solve
91 time(sl) = (jnow-tmp)*24*60*60;
92 );
93 display time;
94
```
Open Architecture

- Designed to interact with other systems

Model independent of
- Platform
- Solver
- Data
- User-Interface
Platform Independence

➢ GAMS available on all major computing platforms

➢ Move your models between platforms with ease
Solver Independence

Uniform interface to all major solvers
➢ More than 30 academic and commercial solvers connected to GAMS
➢ Switch between solvers with one line of code
➢ Documentation
➢ Licensing (GAMS as a „license broker“)

Solver Manuals

A large number of solvers for mathematical programming models have been hooked up to GAMS. The tables below provide a brief description of each solver, the model types each solver solving, and the platforms supported by each solver. For general information on using GAMS solvers, see Solver Usage.

<table>
<thead>
<tr>
<th>Solver</th>
<th>Vendor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHAECP</td>
<td>Abo University</td>
<td>MINLP solver based on the extended cutting plane (ECP) method</td>
</tr>
<tr>
<td>AMPL</td>
<td>GAMS Development Corp</td>
<td>A link to solve GAMS models using solvers within the AMPL modeling system</td>
</tr>
<tr>
<td>ANTIGONE 1.1</td>
<td>Princeton University</td>
<td>Deterministic global optimization for MINLP</td>
</tr>
<tr>
<td>BARON</td>
<td>The Optimization Firm, LLC</td>
<td>Branch-And-Reduce Optimization Navigator for proven global solutions</td>
</tr>
<tr>
<td>BDMLP</td>
<td>GAMS Development Corp</td>
<td>LP and MIP solver that comes with any GAMS system</td>
</tr>
<tr>
<td>BENCH</td>
<td>GAMS Development Corp</td>
<td>A utility to facilitate benchmarking of GAMS solvers and solution verification</td>
</tr>
<tr>
<td>BONMIN 1.5</td>
<td>COIN-OR Foundation</td>
<td>COIN-OR MINLP solver implementing various branch-and-bound and outer approximation algorithms</td>
</tr>
<tr>
<td>CBC 2.9</td>
<td>COIN-OR Foundation</td>
<td>High-performance LP/MIP solver</td>
</tr>
<tr>
<td>CONOPT 3</td>
<td>ARKI Consulting and Development</td>
<td>Large scale NLP solver</td>
</tr>
<tr>
<td>CONOPT 4</td>
<td>ARKI Consulting and Development</td>
<td>Large scale NLP solver</td>
</tr>
<tr>
<td>CONVERT</td>
<td>GAMS Development Corp</td>
<td>Framework for translating models into scalar models of other languages</td>
</tr>
<tr>
<td>COUENNE 0.5</td>
<td>COIN-OR Foundation</td>
<td>Deterministic global optimization for (MI)NLP</td>
</tr>
<tr>
<td>CPLEX 12.8</td>
<td>IBM ILOG</td>
<td>High-performance LP/MIP solver</td>
</tr>
</tbody>
</table>
Platform / Solver Independence

(Parallel) Local and distributed/remote execution

- Solver execution
  - Remote Object Server / Compute Server
  - Distributed MIP (CPLEX, GUROBI)
  - Distributed LP (PIPS-IPM)

- Model execution
  - Grid Computing Facility
  - NEOS (Kestrel)
Data Independence

➢ Declarative Modeling
➢ Sparse Data Structures
➢ Scalable Models

➢ ASCII: Initial model

➢ GDX: Data layer ("contract") between GAMS and applications
  ➢ Platform independent
  ➢ Direct GDX interfaces and general API’s…
User Interface Independence

No preference for a particular User Interface

- Smart Links to popular environments, like Excel, MATLAB, R, Databases, ...
- Object Oriented APIs: .Net, Java, Python, C++
- Embedded Code Facility
- Web Interface (GAMS MIRO)
Evolved and Matured System

- Evolution through decades of R&D
- Maturity through experience and rigorous testing
- Lots of Development and Debugging Tools: Model Profiler, GAMSCHK, CONVERT, PAVER,...
- Quality Assurance
Quality Assurance

➢ What are the impacts of new features, updated modules or platforms?
➢ Is the new distribution backward compatible?

➢ GAMS Test Library: ~800 quality tests
➢ Automatically executed every night for all solver combinations (13,000+ runs/platform)
GAMS Studio

➢ Development Environment for GAMS Models
➢ Platform Independent (Win/Mac/Linux)
➢ Open source Qt project, published on GitHub (GPL)
➢ All features for efficient model development
Uniform System Documentation

➢ Tutorials
➢ GAMS Language
➢ Solver
➢ Tools
➢ APIs
➢ Online/Offline

GAMS Documentation Center

The GAMS Documentation Center provides information on installing and maintaining our GAMS (General Algebraic Modeling System) software.

- **Release Notes** - 25.1.1 Major release notes
- **User’s Guide**
  - Installation and Licensing
  - Tutorials and Examples
  - GAMS Language and Environment
- **Solver Manuals**
- **Application Programming Interface**
- **Glossary**
- **Bibliography**
Free Model Libraries

➢ More than 1,600 models
➢ Part of any distribution
➢ Organized in different libraries
➢ Application Specific e.g. Finance, Energy
➢ Data Connections
➢ System Tests
Striving for Innovation and Compatibility

Models must benefit from

- Advancing hardware / New Platforms
- Enhanced / New solver and solution technology
- New Modeling Concepts
- Improved / New interfaces

Protect Investments of Users

- Life time of a model: 15+ years
- New maintainer, platform, solver, user interface
- Backward Compatibility
- Don’t lock developers and users into a certain environment.
OR Modeler’s Perspective

- Problem class
- Algorithm / Algebra
- Data
- Solver
- Solution
Separation of Tasks

➢ Use **GAMS** for modeling and optimization

➢ Use **Object oriented GAMS APIs** for connecting GAMS to other environments
  ➢ ASCII (e.g. CSV)
  ➢ Smart Links to Databases, Spreadsheets, Matlab, R,…
  ➢ .Net, Java, Python, C++
  ➢ Embedded Code Facility (Python)
  ➢ Communication through Memory or Files

→ *(Some) Programming required*
Model Deployment

- Increasing importance for intuitive deployment and visualization
- Need for easy-to-use tools
- End-users are not modeling experts

→ Configuration instead of Programming
GAMS MIRO

- **Model Interface with Rapid Orchestration**
- A web interface for GAMS models
- Based on Shiny (R), open source (GPL)
- Desktop / Server Version

Develop GAMS model

Click to deploy

Web browser
Agenda

- GAMS at a Glance
- Model Development and Model Deployment

GAMS MIRO Demo
Model: *Pickstock*

- **Data:** Performance of all shares of the Dow Jones index over one year
- **Goal:** Find a small selection of stocks that follows the Dow Jones as closely as possible
- **Optimization model:** Select a subset (≤ maxstock) of Dow Jones stocks, along with weights, so that this portfolio behaves similarly to the overall index (in the training phase)

Training phase
Select small subset of stocks (Optimization)

Testing phase
How similar behaves the stock selection to the DJ? (Evaluation of results)

\[
\begin{align*}
\text{minimize} & \quad \text{obj} := \sum_{ds} \text{slpos}_{ds} + \text{slneg}_{ds} \\
\text{subject to} & \quad \sum_s \text{price}_{ds,s} \cdot w_s = \text{index}_{ds} + \text{slpos}_{ds} - \text{slneg}_{ds} \quad (\forall ds) \\
& \quad w_s \leq p_s \quad (\forall s) \\
& \quad \sum_s p_s \leq \text{maxstock} \\
& \quad w_s \geq 0, \quad p_s \in \{0, 1\} \quad (\forall s) \\
& \quad \text{slpos}_d \geq 0, \quad \text{slneg}_d \geq 0 \quad (\forall d)
\end{align*}
\]
Model: *Pickstock*

\[
\text{minimize } \quad \text{obj} := \sum_{ds} \text{slpos}_{ds} + \text{slneg}_{ds}
\]
Model: *Pickstock*

\[
\text{minimize } \quad \text{obj} := \sum_{ds} s\text{pos}_{ds} + s\text{neg}_{ds}
\]
Model: *Pickstock*

minimize \[ \text{obj} := \sum_{ds} \text{slpos}_{ds} + \text{slneg}_{ds} \]
Hypercube mode
scenario generation

Base mode

Hypercube mode
Hypercube mode
scenario generation

Select the maximum number of stocks

Select the number of days for training

Step size

Step size

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
<td></td>
</tr>
</tbody>
</table>

Number of scenarios = Cartesian product of scalar input combinations

+ 1

+ 5

+ 5
Scenario runs and sensitivity analysis

The GAMS MIRO Hypercube mode
Hypercube mode
Data import & monitoring of scenario runs

![Import data table](image-url)
### Hypercube mode

**Scenario management**

#### Load scenarios

<table>
<thead>
<tr>
<th>Time created</th>
<th>Job tags</th>
<th>AND</th>
<th>Time created</th>
<th>Job tags</th>
<th>AND</th>
</tr>
</thead>
<tbody>
<tr>
<td>between 2019-02-01 to 2019-02-21</td>
<td>is superman</td>
<td>-</td>
<td>between 2019-02-22 to 2019-03-04</td>
<td>is wonder woman</td>
<td>-</td>
</tr>
<tr>
<td>OR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Show 10 entries

<table>
<thead>
<tr>
<th>Owner</th>
<th>Time created</th>
<th>Job tags</th>
<th>maximum number of stocks to select</th>
<th>MIP-Solver</th>
<th>number of days for training</th>
<th>Ratio between error test and error train</th>
<th>Absolute error in entire testing phase</th>
<th>Absolute error in entire training phase</th>
<th>last date of training period</th>
</tr>
</thead>
</table>

Showing 1 to 10 of 220 entries
### Hypercube mode Analysis

#### Analyze Scenarios

<table>
<thead>
<tr>
<th>Index</th>
<th>stat_Status</th>
<th>stat_Efficiency</th>
<th>stat_SolutionQuality</th>
<th>solvedata</th>
<th>documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SolverTime - arith. means**

- **Filter:** all instances
- **SolverTime**
  - aquaman
  - joker
  - narschach
  - superman
  - wonder woman
  - virt. best
  - virt. worst

**Absolute performance profile (SolverTime)**

- Number of instances with gap <= 0.001% and no fail for all solvers
- SolverTime

**Solvedata**

- aquaman
- joker
- narschach
- superman
- wonder woman
- virt. best
- virt. worst
GAMS MIRO on a server

- Based on docker technology
- Authentication: LDAP, OAuth 2.0, Google,…
- Multi-user and application management
- Load balancing
- Rolling Updates
Key points

- Quick deployment of GAMS models
- Intuitive usage without GAMS knowledge
- Easy and convenient data management
- Powerful data visualization
- Scenario Management
- Data export
Conclusions

➢ Desktop / Server Version
➢ Commercially supported

➢ Currently a BETA version
➢ Used in commercial projects

➢ Installer for Windows, MacOS, (and Linux)
➢ Configuration Generator
➢ Comprehensive Documentation
More Information

Documentation & Software
https://www.gams.com/miro/

Video
https://youtu.be/7pUrZ-u9ZcQ

Thank you!