Deploying Optimization Applications
- Concepts and Challenges -

Franz Nelissen
F Nelissen@gams.com
GAMS Development Corp.
GAMS Software GmbH
www.gams.com

San Antonio, TX, April 2013
Company Background

- Roots: World Bank, 1976
- Went commercial in 1987
- GAMS Development Corporation (Washington, Houston)
- GAMS Software GmbH (Frechen, Braunschweig)
- Tool Provider: General Algebraic Modeling System
Agenda

- What is GAMS?
- Model Deployment
- Recent Enhancements
What is GAMS?

GAMS at a Glance

Design Principles
Algebraic Modeling Languages (AML)

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
Impact of Algebraic Modeling Languages

• Simplified model development, changes, and transfer

• Rapid prototyping

• Added value to existing applications

• Increased productivity, quality, reliability and maintainability

→ Important vehicle to make mathematical optimization available to a broader audience, e.g. domain specific experts
GAMS: Core Elements

- Language compiler and execution system
- Solver for various problem classes
- User interfaces
- Connectivity and Productivity Tools

**Solver**
- ALPHAECP, BARON, COIN, CONOPT, CPLEX, DECIS, DICOPT, GLOMIQO, GUROBI, IPOPT, KNITRO, LGO, LINDO, MINOS, MOSEK, OQNLP, PATH, SNOPT, SULUM, XA, XPRESS, ...

**User Interfaces**

**GAMS Language Compiler and Execution System**

**Connectivity Tools**
- Uniform Data Exchange (ASCII, binary)
- Low-level and object-oriented API
- Ext. programs: EXCEL, MATLAB, R, ...

**Productivity Tools**
- Integrated Development Environment (IDE)
- Integrated Data Browser and Charting Engine
- Model Debugger and Profiler
- Model Libraries
- Benchmarking and Deployment
- Quality Assurance and Testing System
- Data and Model Encryption
- Grid Computing
- Scenario Reduction
GAMS: Broad Range of Application Areas*

- Agricultural Economics
- Chemical Engineering
- Econometrics
- Environmental Economics
- Finance
- International Trade
- Macro Economics
- Management Science/OR
- Micro Economics
- Applied General Equilibrium
- Economic Development
- Energy
- Engineering
- Forestry
- Logistics
- Military
- Mathematics
- Physics

* Illustrative examples in the GAMS Model Library
GAMS: Broad User Network

18 Mar 2013 to 24 Mar 2013: 5,071 visits shown above
Statistics updated 24 Mar 2013@08:38GMT: 5,094 visits
Total since 9 Sep 2010: 569,609. Previous 24hrs: 773.

Notes | Country totals =>

Current Country Totals
From 18 Mar 2013 to 24 Mar 2013

- United States (US): 1,572
- India (IN): 998
- Finland (FI): 278
- Turkey (TR): 152
- Pakistan (PK): 123
- Bangladesh (BD): 109
- Spain (ES): 93
Download GAMS Distribution 24.0.2

Please consult the release notes before downloading a system. The installation notes will guide you through the process.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 64 bit</td>
<td>Windows 8 x64, Windows 7 x64, Windows Vista x64, Windows 2000 x64</td>
</tr>
<tr>
<td>Linux 32 bit</td>
<td>AIX 5.3 or higher, PowerPC chip, 64 bit (aix_64)</td>
</tr>
<tr>
<td>Linux 64 bit</td>
<td>AMD- or Intel-based 64-bit Linux systems. The software was tested on Ubuntu 14.04 and Fedora 22.</td>
</tr>
<tr>
<td>Mac OS X Intel 64 bit</td>
<td>Macintosh Intel-based systems (x64_64) built on Darwin 12.1.0</td>
</tr>
<tr>
<td>Solaris SPARC 32 bit</td>
<td>Solaris 2.8 or higher on SUN Sparc (sparc_32)</td>
</tr>
<tr>
<td>Solaris SPARC 64 bit</td>
<td>Solaris 2.8 or higher on SUN Sparc (sparc_64)</td>
</tr>
<tr>
<td>Solaris x64_64 bit</td>
<td>Solaris 10 or higher on AMD- or Intel-based 64-bit (x64_64)</td>
</tr>
<tr>
<td>Linux/Wine (beta)</td>
<td>AMD- or Intel-based Linux systems. The software uses the Wine API.</td>
</tr>
<tr>
<td>Mac/Wine (beta)</td>
<td>For more information please visit this page.</td>
</tr>
</tbody>
</table>
The General Algebraic Modeling System (GAMS) is a high-level modeling system for mathematical optimization. GAMS is designed for modeling and solving linear, nonlinear, and mixed-integer optimization problems. The system is tailored for complex, large-scale modeling applications and allows the user to build large maintainable models that can be adapted to new situations. The system is available for use on various computer platforms. Models are portable from one platform to another.

GAMS was the first algebraic modeling language (AML) and is formally similar to commonly used fourth-generation programming languages. GAMS contains an integrated development environment (IDE) and is connected to a group of third-party optimization solvers. Among these solvers are BARON, COIN solvers, CONOPT, CPLEX, DICOPT, GUROBI, MOSEK, SNOPT, and XPRESS.

GAMS facilitates the users to implement a sort of hybrid algorithms combining different solvers in a seamless way. Models are described in concise algebraic statements which are easy to read, both for humans and machines. GAMS is among the most popular input formats for the NEOS Server for Optimization. Although initially designed for applications related to economics and management science, it has a large community of users from various backgrounds of engineering and science.
What is GAMS?

GAMS at a Glance

Design Principles
GAMS Design Principles

Universal language: Algebra, similar to mathematical notation

- Algebra (Expressions): model equations
- Relational Algebra (SQL): data manipulation
- Model is communication device and executable documentation of the optimization problem

\[
\begin{align*}
\text{Objective} & : & \sum_i \sum_j c_{i,j} \times x_{i,j} & \rightarrow \min \\
\text{Observe supply limit at plant } i: & & \sum_j x_{i,j} & \leq a_i \quad \forall i \\
\text{Satisfy demand at market } j: & & \sum_i x_{i,j} & \geq b_j \quad \forall j \\
& & x_{i,j} & \geq 0 \quad \forall i, j
\end{align*}
\]
Balanced mix of declarative and procedural elements:

- **Declarative:** sets, parameters, variables, equations, models
- **but also procedural elements like:**
  - **For**
    ```gams
    scalar scen;
    for (scen=1 to 10 by 0.5,
        f = 10*scen; c(i,j) = f * d(i,j) /1000;
        solve transport using lp minimizing z;
        display z.l;
    );
    ```
  - **Loop/If**
    ```gams
    loop (h,
        if (work(h),
            pay(i,h) = 0.6 * pay(i,h);
        else
            pay(i,h) = 1.5 * pay(i,h);
        );
    );
    ```
  - **Macros**
    ```gams
    $macro eqcon(y)  sqr{1-sqr[x('1')]*sqr[y]} - x('1')*sqr[y] - sqr[x('2')]) + x('2')
    eqconlow.. eqcon(y('1')) =e= vio;
    eqconlbd(dlbd).. eqcon(ylbd('1',dlbd)) =l= 0;
    eqconubd(ubd).. eqcon(yubd('1',ubd)) =u= -epsilon;
    ```
  - **User defined functions**
    ```gams
    $funclibrary fitlib
    function fitp /fitlib.fitparam/
    fit /fitlib.fitfunc/;
    ```

GAMS Design Principles
GAMS Design Principles

Independent Layers:

- Model and operating system
- Model and solution methods (solver)
- Model and modeling system
- Model and data
- Model and user interface
# Independence of Model and Operating System

## Supported Platforms

<table>
<thead>
<tr>
<th>Solver/Platform availability - 24.0</th>
<th>December 24, 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>x86</strong></td>
<td><strong>x86_64</strong></td>
</tr>
<tr>
<td>ALPHAECP</td>
<td>MS Windows</td>
</tr>
<tr>
<td>BARON 11.7</td>
<td>✓</td>
</tr>
<tr>
<td>BDMLP</td>
<td>✓</td>
</tr>
<tr>
<td>CILNDR</td>
<td>✓</td>
</tr>
<tr>
<td>CONOPT 3</td>
<td>✓</td>
</tr>
<tr>
<td>CPLEX 12.5</td>
<td>✓</td>
</tr>
<tr>
<td>DECIS</td>
<td>✓</td>
</tr>
<tr>
<td>DICOPT</td>
<td>✓</td>
</tr>
<tr>
<td>GLPK 4.0</td>
<td>✓</td>
</tr>
<tr>
<td>GUMPS</td>
<td>✓</td>
</tr>
<tr>
<td>KNITRO 8.0</td>
<td>✓</td>
</tr>
<tr>
<td>LINDO 7.0</td>
<td>✓</td>
</tr>
<tr>
<td>LINDOGLBAL 7.0</td>
<td>✓</td>
</tr>
<tr>
<td>LGO</td>
<td>✓</td>
</tr>
<tr>
<td>MILES</td>
<td>✓</td>
</tr>
<tr>
<td>MINOS</td>
<td>✓</td>
</tr>
<tr>
<td>MOSEK 6</td>
<td>✓</td>
</tr>
<tr>
<td>MPSGE</td>
<td>✓</td>
</tr>
<tr>
<td>MSNLPL</td>
<td>✓</td>
</tr>
<tr>
<td>NLPEC</td>
<td>✓</td>
</tr>
<tr>
<td>OQNLP</td>
<td>✓</td>
</tr>
<tr>
<td>PATH</td>
<td>✓</td>
</tr>
<tr>
<td>SBB</td>
<td>✓</td>
</tr>
<tr>
<td>SCIP 3.0</td>
<td>✓</td>
</tr>
<tr>
<td>SNOPT</td>
<td>✓</td>
</tr>
<tr>
<td>SOLPLEX 1.7</td>
<td>✓</td>
</tr>
<tr>
<td>SULUM 1.0</td>
<td>✓</td>
</tr>
</tbody>
</table>

For detailed information on supported versions of operating systems and solvers, please call.
## Independence of Model and Solution Methods

### Supported Model Types and Solvers

<table>
<thead>
<tr>
<th>Solver/Model type availability</th>
<th>December 24, 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Solver</strong></td>
<td><strong>LP</strong></td>
</tr>
<tr>
<td>ALPHAEC</td>
<td>✓</td>
</tr>
<tr>
<td>BARON 11.7</td>
<td>✓</td>
</tr>
<tr>
<td>BDMILP</td>
<td>✓</td>
</tr>
<tr>
<td>COIN-OR</td>
<td>✓</td>
</tr>
<tr>
<td>CONOPT 3</td>
<td>✓</td>
</tr>
<tr>
<td>CPLEX 12.5</td>
<td>✓</td>
</tr>
<tr>
<td>DECIS</td>
<td>✓</td>
</tr>
<tr>
<td>DICOPT</td>
<td>✓</td>
</tr>
<tr>
<td>GLOMIQO 2.1</td>
<td>✓</td>
</tr>
<tr>
<td>GUROBI 5.0</td>
<td>✓</td>
</tr>
<tr>
<td>KNITRO 8.0</td>
<td>✓</td>
</tr>
<tr>
<td>LINDO 7.0</td>
<td>✓</td>
</tr>
<tr>
<td>LINDOGLOBAL 7.0</td>
<td>✓</td>
</tr>
<tr>
<td>LGO</td>
<td>✓</td>
</tr>
<tr>
<td>MILES</td>
<td>✓</td>
</tr>
<tr>
<td>MINOS</td>
<td>✓</td>
</tr>
<tr>
<td>MOSEK 6</td>
<td>✓</td>
</tr>
<tr>
<td>MPSGE</td>
<td>✓</td>
</tr>
<tr>
<td>MSNLP</td>
<td>✓</td>
</tr>
<tr>
<td>NLPEC</td>
<td>✓</td>
</tr>
<tr>
<td>QNLP</td>
<td>✓</td>
</tr>
<tr>
<td>PATH</td>
<td>✓</td>
</tr>
<tr>
<td>SBB</td>
<td>✓</td>
</tr>
<tr>
<td>SCIP 3.0</td>
<td>✓</td>
</tr>
<tr>
<td>SNOPT</td>
<td>✓</td>
</tr>
<tr>
<td>SOPEX 1.7</td>
<td>✓</td>
</tr>
<tr>
<td>SULUM 1.0</td>
<td>✓</td>
</tr>
<tr>
<td>XA</td>
<td>✓</td>
</tr>
<tr>
<td>XA</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Contributed Plug&Play solvers**

**Operating System**

**Solver**
Independence of Model and Modeling System

- Don’t lock users into GAMS System
- Support translation of GAMS code into other environments
- GAMS/CONVERT converts GAMS to AMPL, BARON, CplexLP, MPS, FixedMPS, …
- Component Libraries: Wrap AML components in library/plug-in and make them accessible to .NET/Java/…
Independence of Model and Data

- Declarative Modeling
- Scalability / Sparsity
- Various interfaces (ASCII, GDX)
- GDX serves as data layer between GAMS and other applications

**Binary Data Exchange**
- Fast exchange of data at any stage
- Binary (no loss of precision)
- Sets, Parameters, Variable- and Equation values (not algebra)
- Can be compressed
- Platform Independent
- Open API for all major languages
- Scenario Management Support
- No license required
GAMS Fundamentals: Summary

- Balanced mix of declarative and procedural elements
- Platform independence
- Hassle-free switch of solution methods and solvers
- Independent layers:

→ Models benefit from
  - Advancing hardware
  - New OS / Paradigms (e.g. Cloud)
  - Enhanced / new solver technology
  - Improved / upcoming interfaces to other systems
Model Deployment

Change in Focus

Some Examples

Deploying GAMS Models
Change in Focus: 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>FERTU²</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>CAMCGE</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>GANGES⁴</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>YENCEM⁴</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>EGYPT⁷</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.
⁴The problem is too big for MINOS, ZOOM was used instead.
⁵A nonlinear problem. 63% of the non-zeroes are nonlinear.
⁶A nonlinear problem. 58% of the non-zeroes are nonlinear.
⁷A mixed binary problem, with 55 binary variables (solved with a relative termination criterion of 10%).
Change in Focus: ... and now

<table>
<thead>
<tr>
<th>Model</th>
<th>Type</th>
<th>Solution Time (s)</th>
<th>Improvement Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>camcge</td>
<td>NLP</td>
<td>468</td>
<td>15097</td>
</tr>
<tr>
<td>dinamico</td>
<td>LP</td>
<td>1986</td>
<td>15888</td>
</tr>
<tr>
<td>egypt*</td>
<td>MIP</td>
<td>1758</td>
<td>117200</td>
</tr>
<tr>
<td>fertd*</td>
<td>MIP</td>
<td>2382</td>
<td>38419</td>
</tr>
<tr>
<td>ganges</td>
<td>NLP</td>
<td>546</td>
<td>5009</td>
</tr>
<tr>
<td>sarf</td>
<td>LP</td>
<td>9210</td>
<td>66259</td>
</tr>
<tr>
<td>yemcem*</td>
<td>MIP</td>
<td>510</td>
<td>3643</td>
</tr>
</tbody>
</table>

* MIP 1988 solver ZOOM, 2008 solver CPLEX
Change in Focus: Past

Computation → Hardware / Algo.
limits application → Users left out
Change in Focus: Now

Computation → Users left out → Modeling skill limits applications → Users involved
Change in Focus: Now / Future

Computation

- Users left out

Model

- Users involved

Application

- Embedded Models
- Users hardly aware of model
## Change in Focus: Fields

<table>
<thead>
<tr>
<th>Mathematical Optimization</th>
<th>Business Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Well defined (narrow) application area</strong></td>
<td><strong>Broad application area</strong></td>
</tr>
<tr>
<td><strong>Small set of methods (LP, MIP, NLP, …)</strong></td>
<td><strong>Wide range of methods</strong></td>
</tr>
<tr>
<td><strong>Small amounts of (structured) data</strong></td>
<td><strong>Huge amounts of data</strong></td>
</tr>
<tr>
<td><strong>Visualization?</strong></td>
<td><strong>Visualization crucial</strong></td>
</tr>
<tr>
<td><strong>Niche market</strong></td>
<td><strong>Big market</strong></td>
</tr>
</tbody>
</table>
Change in Focus: Looking at Problems

- Cutting Stock Problem: Paper rolls of fixed width ("raw") must be cut into smaller portions ("finals").

- Input:
  - Width of the raw
  - Demand: Widths and number of finals

- Objective: Minimize the required number of raws

- Output:
  - Combination of cuts ("patterns")
  - Produced number of each pattern
  - Number of required raws
Change in Focus: Modeler...

GAMS

Master Model

Pricing Model

New Patterns

Gilmore & Gomory (1961)

Demand Duals
Change in Focus: ... Application Developer

- **Software Architecture**
- **Object Oriented Design**
- **Components, Encapsulation, Classes, Data Access Layer...**
Analytic Professionals & IT Professionals

- **Analytic Professionals**
  - INFORMS 10-12k members
  - INFORMS: First Certified Analytics Professional (CAP) certification, April 7-9, 2013 San Antonio

- **IT Professionals**
  - 3.3 Million IT professionals in the US (2006, Ziff Davis)
  - 15 million IT professionals worldwide (2009, Capers Jones, SPRI)
  - Rapidly changing IT environments (Cloud, RCP, Web, Mobile, …)
  - Certifications for IT professionals are standard
What is special about Optimization?

Optimization

- May take longer than one is willing to wait
- Will eventually fail

→ Algebraic Modeling Systems provide environment to build robust and fail safe systems
What is special about Optimization?

Optimization models
- are expensive to develop
- may have long lifespan

→ Modeling Systems & Applications have to be adjusted:
  • New computer paradigms (e.g. cloud computing, tablets)
  • New solver technology and solution methods
  • New graphical user interfaces and deployment environments

→ Minimize risks for (new) clients / management
→ Don’t lock developers and users into a certain environment.
→ Protect user investments
→ Provide cutting edge technology
### Model Deployment

- **Change in Focus**
- **Some Examples**
- Deploying GAMS Models
Example – All in One – Top Down

- Add MO-Capability or AML to existing analytical software systems.
- “large” user base, e.g. MATLAB, or SAS
Example – All in One – Bottom Up

- Integrate GUI-builder into AML System
- “small” user base, e.g. AIMMS (Pro) or FICO Xpress-Insight

```
“Application”

GUI Builder

AML

Solver
```

Example – All in One – Bottom Up

- Integrate GUI-builder into AML System
- “small” user base, e.g. AIMMS (Pro) or FICO Xpress-Insight

```
“Application”

GUI Builder

AML

Solver
```
Example – Embedding – Top Down

Microsoft Deployment Tools
e.g. Sharepoint, IIS, ASP.NET

“Application”

.NET

MS Office (Excel)

Solver

AML

- .NET based optimization platform
- Includes a declarative AML
- Excel plug-in
- Future??
Example – Embedding – Bottom Up

- Optimization capability included in wide-spread tool
- Excel (and other MS Tools) as deployment environment
- Also other interfaces (e.g. .NET)
Example – Model Translation/Embedding

- **Automatic** translation: early versions of OPL translated OPL source & scripts to C++

- **Manual** translation of model into programming language using tools that mirror the algebraic representation of a single model instance, e.g. IBM ILOG Concert
  - Restricted to linear models
  - Complex models may be difficult to translate
Example – Rich Client Platform

- “Construction Kit” with different connected elements
- Use (open source) existing framework to build applications, e.g. IBM ODM Enterprise

Rich Client Platform

- Eclipse (IBM)
- NetBeans (Oracle)

AML

Database

Analytical Software

Solver
Deploying GAMS Models

Integrate GAMS Models

Provide Links

Collaborate
Bridging The Gap

Modeler

Application Developer

GAMS OO API
Concept: Separation of Tasks

- Use GAMS for modeling and optimization tasks
- Programming languages like C# (.NET), Java and Python are well-suited for developing applications (GUI, Web, …)
- Frameworks available for a wide range of specific tasks, e.g. GUI and Web development
- Communication between GAMS and application through GAMS APIs
GAMS API: Basic Functionality

1. **GDX API**: Create Input for GAMS Model
2. **Callout to GAMS**
   - Option API: GAMS option settings
   - GAMSX API: Start GAMS
3. **GDX API**: Get Solution from GAMS Model
Low level GAMS APIs

- High performance and flexibility
- Automatically generated APIs for several languages (C, Delphi, Java, Python, C#, …)
- Data Exchange (GDX), job control (GAMSX), options (OPT)
- Part of any GAMS distribution, no license required
Object-oriented GAMS API

- Additional layer on top of the low level APIs
- Object-oriented: .NET, Java, and Python
- Part of any GAMS distribution, no license required
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 → GAMSDatabase Class

• Set GAMS and Solver Options → GAMSOptions Class

• Control GAMS execution → GAMSJob Class

• Scenario Solving: Feature to solve multiple very similar models in a dynamic and efficient way → GAMSModelInstance Class
namespace TransportSeq
{
    class Transport1
    {
        static void Main(string[] args)
        {
            GAMSWorkspace ws = new GAMSWorkspace();
            GAMSJob t1 = ws.AddJobFromString(GetModelText());

            t1.Run();
            foreach (GAMSVariableRecord rec in t1.OutDB.GetVariable("x"))
            {
                Console.WriteLine("  level=" + rec.Level);
                Console.WriteLine("  marginal=" + rec.Marginal);
            }
        }
    }
}
static String GetModelText()
{
    String model = @" Sets
    i  canning plants   / seattle, san-diego /
    j  markets          / new-york, chicago, topeka / ;

    Parameters
    a(i)  capacity of plant i in cases
        / seattle   350
        san-diego  600 /
    b(j)  demand at market j in cases
        / new-york  325
        chicago   300
        topeka    275 / ;

    < . . . >

    Solve transport using lp minimizing z ;"
    return model;
}
package TransportSeq;
import com.gams.api.*
class Transport1 {
    static void main(String[] args) {
        GAMSWorkspace ws = new GAMSWorkspace();

        GAMSJob t1 = ws.addJobFromString(getModelText());
        t1.run();

        for (GAMSVariableRecord rec : t1.OutDB().getVariable("x")) {
            System.out.println("x(" + rec.getKeys()[0] + ", " + rec.getKeys()[1] + "):");
            System.out.println("     level =" + rec.getLevel());
            System.out.println("     marginal =" + rec.getMarginal());
        }
    }
}
transport.py

```python
from gams import *

if __name__ == "__main__":
    ws = GamsWorkspace()

    t1 = ws.add_job_from_string(get_model_text())
    t1.run()

    for rec in t1.out_db["x"]:
        print rec
```
Scenario Solving - Loop

Loop(s,
    f = ff(s);
    solve mymodel min z using lp;
    objrep(s) = z.l;
);

• Data exchange between solves possible
• Model rim can change
• Each solve needs to regenerate the model
• User updates GAMS Symbols instead of matrix coefficients
foreach (string s in scen)
{
    f.FirstRecord().Value = v[s];
    modelInstance.Solve();
    objrep[s] = z.FirstRecord().Level;
}

• Saves model generation and solver setup time
• Hot start (keeps the model hot inside the solver and uses solver’s best update and restart mechanism)
• Data exchange between solves possible
• Model rim unchanged from scenario to scenario
GAMSModelInstance etc.

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

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

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

**GAMSModifier**
- Marks elements of a GAMSModelInstance to be modifiable
GAMSModelInstances in Parallel

Multiple GAMSModelInstances running in parallel with one common data source:
.NET - Application

- Example: Small transport desktop application written in C#
- Convenient data preparation
- Representation of the results in a predefined way
- Modeling details are hidden from the user
Wiki as Deployment Environment

- Technologies: Dokuwiki, JavaScript, PHP, …
- Use GAMS for modeling and optimization
- Run GAMS from Wiki environment
- Share results of a GAMS model
- Start a GAMS model asynchronously and come back later to see the results (batch job)

Wiki – Line Optimization Example

GAMS Application Wiki

Line Optimization

Data

Job name: 
Upload data: [Browse...]
download template
Choose a solver: Cplex

Model

mincars: 3
ccap: 467
cfx: 353100
crm: 5800
trm: 44900
cmp: 90

Submit Job  Help
JavaScript & Google Maps

Route 1  Route 2  Route 3  Route 4  Route 5  Route 6  Route 7  Route 8
Route 9  Route 10  Route 11  Route 12  Route 13  Route 14  Route 15
Route 16  Route 17  Route 18  Route 19  Route 20  Route 21

Map of the Netherlands with markers for different routes.

<table>
<thead>
<tr>
<th>from</th>
<th>to</th>
<th>running time</th>
<th>accumulated running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amsterdam CS</td>
<td>Apeldoorn</td>
<td>89.00</td>
<td>89.00</td>
</tr>
<tr>
<td>Apeldoorn</td>
<td>Hengelo</td>
<td>69.00</td>
<td>158.00</td>
</tr>
<tr>
<td>Hengelo</td>
<td>Oldenzaal Grens</td>
<td>18.00</td>
<td>176.00</td>
</tr>
</tbody>
</table>

Showing 1 to 3 of 3 entries
Deploying GAMS Models

Integrate GAMS Models

Provide Links

Collaborate
Links to other Software

Example - Excel and GAMS

- VBA GAMS API to call GAMS from Excel
- Exchange of input data and results using either GDXXRW or GDX API
Excel and GAMS

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Coal</td>
<td>-17.62</td>
<td>-35.24</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>5</td>
<td>WasteHeat</td>
<td>36.43</td>
<td>53.2</td>
<td>-42.56</td>
<td>-10.64</td>
<td>-53.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Steam</td>
<td>5.5</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.72</td>
<td></td>
<td>-16.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Exhaust</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16.7</td>
<td>-16.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.6184</td>
<td></td>
<td>39.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Solution - Generated Heat & Electricity**

Solver: CPLEX
Equations: 14786  Variables: 19489
Model Status: 8 Integer Solution
Solver Status: 1 Normal Completion
Iterations: 8488 Solve Time: 0.58
---
Objective Value: 834036
No. of Hours GT is ON: 530
No. of GT Starts: 8
Excel and Analytics

“...the truth is, for many organizations, Excel has always been a part of their business intelligence portfolio for several reasons:

• Familiarity with Excel...

• Built-in flexibility...

• Rapid development...

• Powerful data connectivity and automation capabilities...

• Little to no incremental costs...

Links to other Software

Example – MATLAB / R and GAMS

• Give MATLAB or R users access to all the optimization capabilities of GAMS
• Allow visualization and analysis of GAMS data directly within MATLAB or R
Deploying GAMS Models

Integrate GAMS Models

Provide Links

Collaborate
Collaborative System Design
GAMS available in IBM ODM Enterprise (Eclipse Plug-in)
## Recent Enhancements

<table>
<thead>
<tr>
<th>GAMS System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platforms</td>
</tr>
<tr>
<td>Solvers</td>
</tr>
<tr>
<td>Interfaces</td>
</tr>
<tr>
<td>(Scenario Solver)</td>
</tr>
</tbody>
</table>
GAMS System

• Support for user-defined:
  – Macros
  – Function libraries

• Asynchronous job execution

• Scenario Solver (GUSS)

• Extended Mathematical Programming (EMP) Framework
Excursus: EMP, what?

With new modeling and solution concepts 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

Original Model -> EMP Information

Translation

Reformulated Model

Viewable

Solving using established Algorithms

Solution

Mapping Solution into original space
SP – Deterministic Model

Scalar  
- freight in dollars per case per thousand miles  /$0/  
- p  penalty for unsatisfied demand  /$1/  
- bf  demand factor  /$1/  

Parameter  
c(i,j)  transport cost in thousands of dollars per case  
\[ c(i,j) = f \cdot d(i,j) / 1000; \]

display c;  

display u;  

Variables  
x(i,j)  shipment quantities in cases  
u(j)  unsatisfied demand ( recourse variable)  
z  total transportation costs in thousands of dollars  

Positive Variable  
x, u  

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

\[
\begin{align*}
\text{cost} & : \quad z = \sum_{i} \sum_{j} c(i,j) \cdot x(i,j) + p \cdot \sum_{j} u(j); \\
\text{supply(i)} & : \quad \sum_{j} x(i,j) = a(i); \\
\text{demand(j)} & : \quad \sum_{i} x(i,j) = g = bf \cdot b(j) - u[j]; 
\end{align*}
\]

Model  transport /all/ ;
Example – EMP File & Results

```plaintext
file emp / 'emp.info' /; put emp '* problem %gams.i%'/;
$onput
randvar bf discrete 0.3 0.95
      0.5 1.00
      0.2 1.05
stage 2 bf u demand
$offput
putclose emp;
Set scen scenarios / l,m,h /;
Parameter
   s_bf(scen) demand factor realization by scenario
   s_u(scen,j) shipment per scenario
   s_x(scen,i,j) shipment per scenario
   s_s(scen) ;
Set dict / scen .scenario.''
   bf .randvar .s_bf
   u .level .s_u
   x .level .s_x /
Solve transport using emp minimizing z scenario dict:
Display s_bf, s_x, s_u;
```

```plaintext
------ 87 PARAMETER s_bf demand factor realization by scenario
     l 0.950,     m 1.000,     h 1.050
------ 87 PARAMETER s_x shipment per scenario
          new-york   chicago   topeka
     l.seattle    35.000     315.000
     l.san-diego   290.000     288.750
     m.seattle    35.000     315.000
     m.san-diego   290.000     288.750
     h.seattle    35.000     315.000
     h.san-diego   290.000     288.750
------ 87 PARAMETER s_u
          new-york
     h     16.250

**** REPORT FILE SUMMARY
```
Stochastic Programming in GAMS

- Based on Extended Mathematical Programming (EMP) framework
- Discrete and parametric distribution for random parameters
- Supports multi-stage recourse problems and chance constraint models

- Make it easy to add uncertainty to existing deterministic models:
  - use specialized algorithms (DECIS or LINDO)
  - or create deterministic equivalent (free solver DE)
- SP examples in the GAMS EMP library
Platforms

• Support for MAC OS X

• Cross-platform licenses

• Wine (Linux, Mac)
**Solvers**

- **GLoMIQO**: Branch-and-bound global optimization for mixed-integer quadratic models

- **Lindo**: Global and stochastic optimization

- **SULUM**: Linear (and soon) mixed integer optimization
Current GAMS Distribution 24.0.2

Released February 24, 2013  www.gams.com/download

• Solver updates
  – BARON 11.9.1
  – CONOPT 3.15I
  – CPLEX 12.5
  – GLOMIQO 2.0
  – GUROBI 5.1
  – KNITRO 8.0
  – LINDO 7.0.1.487
  – MOSEK 6 rev 148
  – XPRESS 23.01.06
  – …
Interfaces

- OO API for .NET, Java and Python
- GDXRRW
  - Connects R-System with GAMS
  - Presents data in a natural form for R users

Source: http://blog.modelworks.ch
### Recent Enhancements

| GAMS System |
| Platforms   |
| Solvers     |
| Interfaces  |
| **Scenario Solver** |
The common way:

Set s / s1*s10 /
Parameter
  A_s(s,i,j) Scenario data
  xlo_s(s,j) Scenario lower bound for variable x
  xl_s(s,j) Scenario solution for x.l
  em_s(s,i) Scenario solution for e.m;
Loop(s,
  A(i,j) = A_s(s,i,j);
  x.lo(j) = xlo_s(s,j);
  solve mymodel min z using lp;
  xl_s(s,j) = x.l(j);
  em_s(s,i) = e.m(i);
);

• GAMS generates model and writes it to disk
• GAMS writes database to scratch files on disk
• GAMS calls solver and vacates memory
• After solver is done: GAMS restarts and swaps database
Gather-Solve-Update-Update-Scatter (GUSS)

Idea:
• Solve one particular model instance (i.e. same individual variables and constraints) with different data and bounds
• Works with any model type in GAMS
• Change in model data for a subsequent solve statement does not depend on previous model solutions

Advantages:
• GAMS stays live
• Model generation only once
• Fast communication with solver
• Hotstarts and advanced basis information
GUSS: Example

Set s / s1*s10 /
Parameter
  A_s(s,i,j) Scenario data
  xlo_s(s,j) Scenario lower bound for variable x
  xl_s(s,j) Scenario solution for x.l
  em_s(s,i) Scenario solution for e.m;

Set dict / s... scenario. ”’
  A. param. A_s
  x. lower. xlo_s
  x. level. xl_s
  e. marginal. em_s /

solve mymodel min z using lp scenario dict;
GUSS: Performance

- Model with 3838 scenarios

<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>
# GUSS: Example

From Erwin Kalvelagen’s Blog

<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</td>
<td>100,000</td>
<td>1068 sec</td>
<td>169 sec</td>
<td>1237 sec</td>
</tr>
<tr>
<td>(call solver as DLL)</td>
<td></td>
<td></td>
<td></td>
<td></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>

<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>

http://yetanothermathprogrammingconsultant.blogspot.de/2012/04/parallel-gams-jobs-2.html
Summary

• Change in focus from algorithm to application

• Different approaches to deploy optimization models

• The challenge remains to provide a deployment environment, which
  – leads to productivity increase
  – establishes confidence in optimization methods
  – and is easy to maintain
Summary

We believe in

– the important role of an AML to build a failsafe and reliable system
– the integration into other systems through
  • OO APIs to existing application frameworks
  • Smart links to other analytical environments
Thank You!

USA

GAMS Development Corp.
1217 Potomac Street, NW
Washington, DC 20007
USA

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

sales@gams.com

Europe

GAMS Software GmbH
P.O. Box 40 59
50216 Frechen
Germany

Phone: +49 221 949 9170
Fax: +49 221 949 9171

info@gams.de

http://www.gams.com