Enhanced Model Deployment in GAMS

Using R/Shiny to deploy and visualize GAMS models in a Web Interface

Lutz Westermann
Frederik Proske
GAMS Software GmbH
Motivation
Motivation

• AMLs are powerful tools for developing solver-independent optimization models

• However, end-users of optimization software are very often not modeling experts
  → Models should be robust (informative error messages, no infeasibility, … )

• Need for easy-to-use tool to visualize data and compare results
How to address the Need for Visualization?

1. Deployment frameworks like AMPL Quandec, FICO Xpress Insight, IBM Decision Optimization Center, …
   
   **Drawback:** High price (costs for these frameworks easily exceed the cost of the modeling system by a factor of 10 to 15)

2. Highly flexible APIs to develop applications (e.g. GAMS comes with several APIs in C++, C#, Java, Python, …)
   
   **Drawback:** Requires a lot of programming effort
Using R/Shiny to deploy GAMS Models

Currently under Development
Background – R

- A language and environment for statistical computing and graphics
- Highly extensible
- Allows to produce well-designed publication-quality plots, including mathematical symbols and formulae
- Runs on Unix, Windows and MacOS
- Available as Free Software under the terms of the Free Software Foundation’s GNU General Public License

Source: www.r-project.org
Background – Shiny

• Open source R package to build interactive web applications
• Provides extensive visualization options
• Allows to host standalone apps on a webpage or embed them in R Markdown documents or build dashboards

Source: shiny.rstudio.com
From GAMS Model to Visual User Interface

Equation

defit(date) 'fit to Dow Jones index'
defpick(symbol) 'can only use stock if picked'
defnumstock 'few stocks allowed'
defobj 'absolute violation (L1 norm) from index';

defit(ds).. sum(s, price(ds,s)*w(s)) =e= index(ds) + slpos(ds) - slneg(ds);
defpick(s).. w(s) =l= p(s);
defnumstock.. sum(s, p(s)) =l= maxstock;
defobj.. obj =e= sum(ds, slpos(ds) + slneg(ds));
Initialization:
✓ Annotating GAMS model (defining the input and output data to be displayed in the WebUI)
Initialization – Connection of GAMS and the User Interface

✔ Fully functional interface by only specifying input and output data
✔ Tabular view of input (editable) and output data
✔ Graphical visualization via pivot charts
Initialization – Connection of GAMS and the User Interface

```gams
$onExternalInput
Set date 'date'
symbol 'stockSymbol';

Parameter
stockData(date,symbol,hrd) 'data of stock on date' ;
maxstock 'maximum number of stocks to select' ;
trainingdays 'number of days for training' ;
$offExternalInput

$onExternalOutput
Set wHd 'w header' / 'weight' /
fHd 'fund header' / 'dow jones','index fund' /
errHd 'stock symbol header' / 'absolute error train', 'absolute error test' ;
Parameter
partOfPortfolio(symbol,wHd) 'what part of the portfolio'
dowVSindex(date,fHd) 'dow jones vs. index fund'
aberror(date,errHd) 'absolute error'
Singleton Set lastDayTraining(date) 'last date of training period' ;
$offExternalOutput

$if not exist webui.gms
$if set GMSWEBUI $abort Asked to do webui but can't find webui.gms. Set idir=path/to/webui
$batinclude webui
```
### Initialization – Connection of GAMS and the User Interface

```plaintext
$onExternalInput
Set date 'date';
symbol 'stockSymbol';

Parameter
stockData(date,symbol,hrd) 'data of stock on date' {}

Scalar
maxstock 'maximum number of stocks to select' {}

$offExternalInput

$onExternalOutput
Set whdr 'fund header' /
errHdr 'stock symbol header' /

Parameter
partOfPortfolio(symbol,whdr) 'what part of the portfolio'
dowVSindex(date,fHdr) 'dow jones vs. index fund'
absererror(date,errHdr) 'absolute error'

Singleton Set lastDayTraining(date) 'last date of training period' vertical marker in chart'';

$offExternalOutput

$if not exist webui.gms
Sifs set GMSWEBUI Sabort Asked to do webui but cant find webui.gms. Set idir=path/to/webui
$batinclude webui
```
Initialization – Connection of GAMS and the User Interface

```gams
Set date   'date'
symbol 'stockSymbol';

Parameter
stockData(date,symbol,hrd)  'data of stock on date'  
maxstock         'maximum number of stocks to select'
trainingdays     'number of days for training'

Scalar

$offExternalInput

Set wHdr   'w header' /'weight' /
fHdr      'fund header' /'dow jones','index fund' /
errHdr    'stock symbol header' /'absolute error train','absolute error test' /

Parameter
partOfPortfolio(symbol,wHdr)    'what part of the portfolio'
dowVSindex(date,fHdr)           'dow jones vs. index fund'
absererror(date,errHdr)         'absolute error'

Singleton Set lastDayTraining(date) 'last date of training period' 

$offExternalInput

@if not exist webui.gms
$batininclude webui
```
Initialization – Connection of GAMS and the User Interface

```
$onExternalInput
Set date 'date'
symbol 'stockSymbol';

Parameter
stockData(date,symbol,hrd) 'data'
maxstock 'maximum number of stocks'
trainingdays 'number of days for testing'

Scalar

$offExternalInput

$onExternalOutput
Set wHdr 'w header'
fHdr 'fund header'
errHdr 'stock symbol header'

Parameter
partOfPortfolio(symbol,wHdr) 'what part of the portfolio'
dowVSindex(date,fHdr) 'dow jones vs. index fund'
absererror(date,errHdr) 'absolute error'

Singleton Set lastDayTraining(date)

$offExternalOutput
```

```javascript
### { "headers":{"date":{"readonly":true}}
### { "slider":{"min":1, "max":"card(stockdata)"}}
### { "slider":{"min":1, "max":"card(stockdata)"}}
```
From GAMS Model to Visual User Interface

1. Initialization:
   ✓ Annotating GAMS model (defining the input and output data to be displayed in the WebUI)

2. Basic setup (optional):
   ✓ Configuration of graphics and UI
Basic Setup – Configuration

✓ Configuration via JSON file
✓ Access to a number of pre-implemented tools for graphical representation
✓ Focus on configuration, not programming

```
"dowjonesindex" : {
    "outType" : "graph",
    "graph" : {
        "tool" : "dygraph",
        "title" : "Dow Jones vs. Index Fund",
        "xdata" : {
            "dow jones" : {
                "label" : "Dow Jones index",
                "fillGraph" : true
            },
            "index fund" : {
                "label" : "Index fund"
            }
        }
    }
}
```
From GAMS Model to Visual User Interface

1. Initialization:
   ✓ Annotating GAMS model (defining the input and output data to be displayed in the WebUI)

2. Basic setup (optional):
   ✓ Configuration of graphics and UI

3. Advanced setup (optional):
   ✓ Sophisticated graphics
   ✓ User- and Application management
Advanced Setup – Sophisticated graphics

- Sophisticated graphics created in R can be included as modules
- Access to the entire R ecosystem
- Easily extendable with the wide spectrum of the R programming language
Advanced Setup – User and Application Management

- Local or server-based solution
- User authentication (e.g. LDAP, Keycloak, Google, GitHub, Facebook)
- Multi-Application support with docker-based technology
Example – Pickstock
Model Definition

- Goal: Definition of an index fund that follows the Dow Jones (DJ)
- Optimization model: Select a small subset of DJ stocks, along with weights, so that this portfolio behaves similarly to the overall index

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

subject to
\[
\sum_{s} \text{price}_{ds,s} \cdot w_s = \text{index}_{ds} + s\text{pos}_{ds} - s\text{neg}_{ds} \quad (\forall ds)
\]

\[
w_s \leq p_s \quad (\forall s)
\]

\[
\sum_{s} p_s \leq \text{maxstock}
\]

\[
w_s \geq 0, \quad p_s \in \{0, 1\} \quad (\forall s)
\]

\[
s\text{pos}_d \geq 0, \quad s\text{neg}_d \geq 0 \quad (\forall d)
\]
Annotating the GAMS Model

```gams
Set date   'date'
  symbol 'stockSymbol';

Parameter stockData(date,symbol,hrd)  'data of stock on date'### { "headers":{"date":"readonly":true} };

Scalar maxstock  'maximum number of stocks to select'### { "slider":{"min":1, "max":card(stockdata$}
trainingdays  'number of days for training'### { "slider":{"min":1, "max":card(stockdata$}

Set wHdr  'w header' / 'weight' /
fHdr  'fund header' / 'dow jones','index fund' /
errHdr  'stock symbol header' / 'absolute error train', 'absolute error test' /;

Parameter partOfPortfolio(symbol,wHdr)  'what part of the portfolio'
dowVSindex(date,fHdr)  'dow jones vs. index fund'
absererror(date,errHdr)  'absolute error'

Singleton Set lastDayTraining(date) 'last date of training period'### vertical marker in chart';

$if not exist webui.gms
$if set GMSWEBUI $abort Asked to do webui but can't find webui.gms. Set idir=path/to/webui
$batinclude webui
```
Setting the Model Input Data

- Data exchange via local files or database connection
- Visualization and modification of input data with intuitive controls
- From a GAMS model to the first interface within minutes
- Comprehensive configurability
Communication with GAMS

✓ Start and Stop model execution
✓ Access to GAMS Log file and Listing file
Inspecting the Results

✓ Extensive visualization options of the model output
✓ Interactive analysis of the results
✓ On-the-fly switch between different options to inspect the results
✓ Solve multiple scenarios or load saved data for comparison
Visualization / Graphics Libraries – Examples

- **plotly**
  - Create interactive web graphics via Plotly's JavaScript graphing library.
  - **author**: richee
  - **tags**: visualization, diagram, networks
  - **js libraries**: plotly.js

- **Diagrammer**
  - Easily create graph diagrams using R.
  - **author**: richimanone
  - **tags**: visualization, diagram, networks
  - **js libraries**: d3, vis, mermaid

- **leaflet**
  - Leaflet is an open-source JavaScript library for interactive maps. This package makes it easy to create Leaflet maps from R.
  - **author**: rntudo
  - **tags**: visualization, maps
  - **js libraries**: leaflet

- **networkD3**
  - A port of Christopher Gandrud’s d3Network package to the htmlwidgets framework.
  - **author**: christopherandrud
  - **tags**: visualization, networks
  - **js libraries**: d3

- **highcharter**
  - A wrapper for the 'highcharts' library.
  - **author**: iku

- **DT**
  - This package provides a function datable() to display R data via the DataTables javascript.

- **rbokeh**
  - An interface to Bokeh that provides a flexible, powerful, declarative framework for creating interactive visualizations.
  - **author**: davidgoble

- **ggiraph**
  - Make ggplot2 graphics interactive.
  - **author**: davidgoble

- **dygraphs**
  - An R interface to the dygraphs JavaScript charting library. It provides rich facilities for interactive graphs.
Summary & Outlook
What we have done so far

- Application connects Web User Interface with a GAMS model
- User Interface allows
  - ✓ Data exchange via local files or database
  - ✓ Modification of the input data
  - ✓ Extensive visualization options
  - ✓ Comparison of different scenarios
  - ✓ Multi-user support based on Docker technology
  - ✓ User authentication
- Tool with intuitive interface for planners

- This “product” is currently under development. If you are interested in getting involved, please contact support@gams.com (or talk to me directly)
What we plan to do

• Batch configuration and execution
  → Running GAMS in a separate container to share resources between different models
• Use docker orchestration tools (Docker Swarm, Kubernetes)
  → Load balancing
  → High availability and scalability
Thank You

Meet us at the GAMS booth!

Reminder: This “product” is currently under development. If you are interested in getting involved, please contact support@gams.com
Welcome to Jupyter @ GAMS!

Enter your credentials in order to sign in or contact GAMS Support for further information.

Getting Started

- Introduction
- Mlco Example
- PickStock Example
- A GAMS Tutorial by Richard E. Rosenthal

Further Help

- Jupyter Notebook Users Manual (from Bryn Mawr College)
- GAMS World Forum
- Contact GAMS
GAMS Jupyter Example

In [17]:
```gams
Parameter fund(date) 'Index fund report parameter';
   fund(d) = sum(s, price(d, s)*w.l(s));
Parameter error(date) 'Absolute error';
   error(d) = abs(index(d)-fund(d));
```

Plotting of the results

In [18]:
```python
%gams pull -d fund error
fig, ax = plt.subplots()
index.plot(y="value", ax=ax, xticks=[0, trainingDays, len(date)], yticks=[], label="Dow Jones")
fund.plot(y="value", ax=ax, xticks=[0, trainingDays, len(date)], yticks=[], label="Index Fund")
```

![Chart showing comparison between Dow Jones and Index Fund](chart.png)
Using GAMS Jupyter Notebooks to tell “optimization stories”

• Runs in a browser/on a server  
  → No local installation needed
• Allows to use notebook technology in combination with GAMS
• Notebooks allow to combine GAMS and Python
  • GAMS works great with well structured data and optimization models
  • Python is very rich in features to retrieve, manipulate, and visualize data that comes in all sort of ways
  • → Combining GAMS and Python in a notebook it is relatively easy to tell an optimization story with text, data, graphs, math, and models

• This “product” is currently under development. Give it a try at https://jupyterhub.gams.com/hub/login