Recent Enhancements in GAMS

Lutz Westermann, GAMS Software GmbH

Agenda

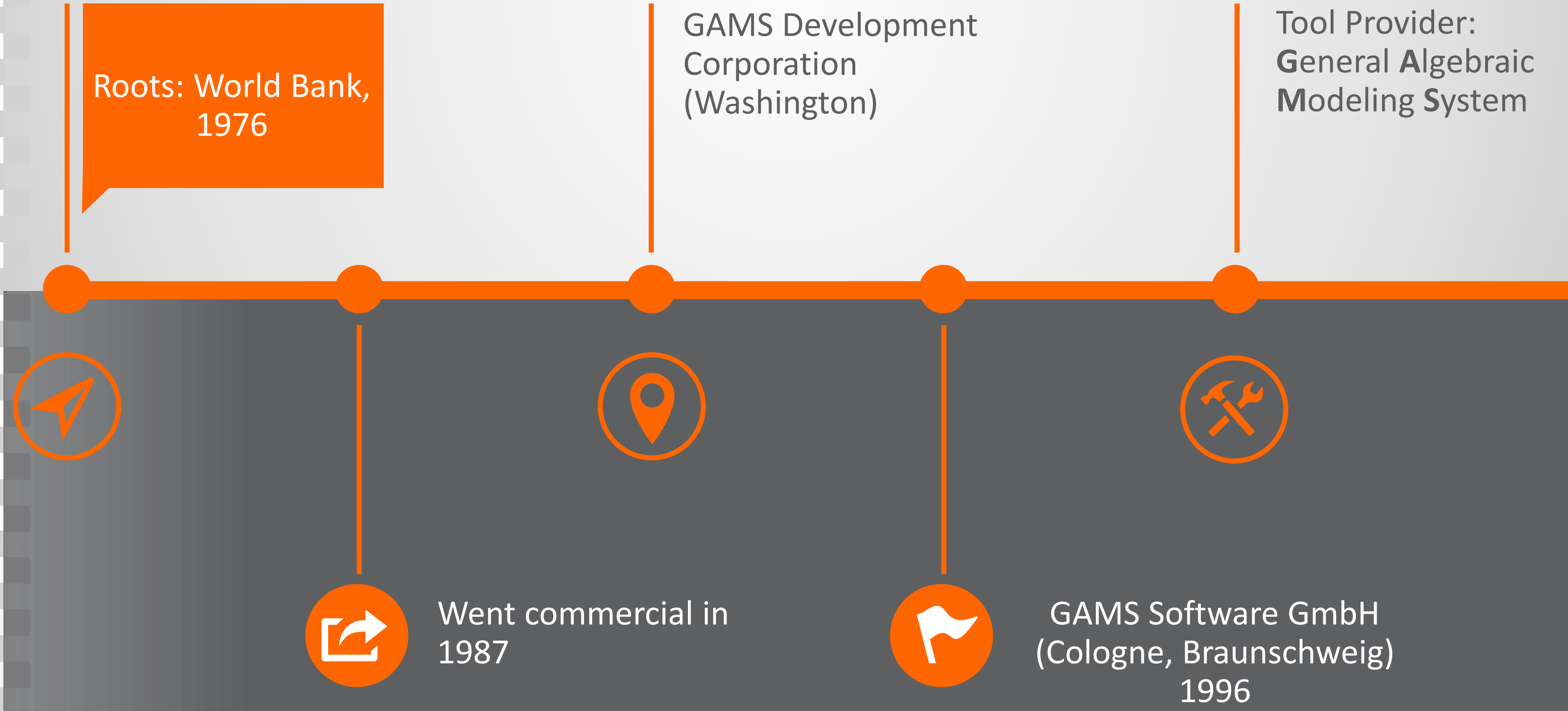
What is GAMS?

Enhancements in GAMS

Enhancements around GAMS

Summary

Company Background



Roots: World Bank, 1976

Went commercial in 1987

GAMS Development Corporation (Washington)

GAMS Software GmbH (Cologne, Braunschweig) 1996

Tool Provider: **G**eneral **A**lgebraic **M**odeling **S**ystem

Algebraic Modeling Languages (AML)

1

- High-level **computer programming languages**
  - Formulation of **mathematical optimization problems**
  - **Notation similar to algebraic notation**

2

- **Do not solve problems directly**, but offer links to state-of-the-art algorithms ("solver-links")

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

What does a modeler have to think about?

1. Problem
2. Mathematics
3. Programming
4. Performance
5. Scalability
6. Connectivity
7. Deployment
8. Maintenance (Life Cycle)
9. ...

Why is GAMS a tool for him?

Broad User Community and Network

GAMS used in more than 120 countries

ClustrMaps®
© 2011

25+ Years
GAMS Development

Broad User Community and Network

More than 10,000 licenses

50% academic users, 50% commercial users

25+ Years
GAMS Development

6,000+ monthly downloads of the free system

Broad Range of Application Areas

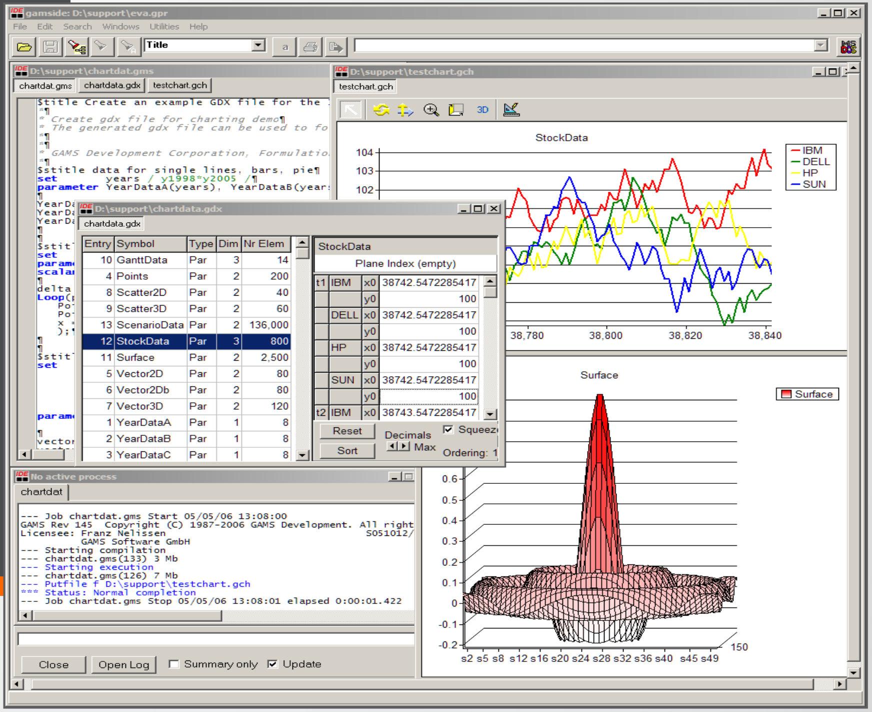
Agricultural Economics	Applied General Equilibrium
Chemical Engineering	Economic Development
Econometrics	Energy
Environmental Economics	Engineering
Finance	Forestry
International Trade	Logistics
Macro Economics	Military
Management Science/OR	Mathematics
Micro Economics	Physics

25+ Years
GAMS Development

Strong Development Environment

GAMS IDE

- Project management
- Editor / Syntax coloring / Spell checks
- Listing file / Tree view / Syntax-error navigation
- Model Debugging / Profiling
- Solver selection / Option selection
- Data viewer (GDX)
    - Export
    - Charting
- GAMS Processes Control
- Model Libraries



```
$title Create an example GDX file for the
*¶
* Create gdx file for charting demo¶
* The generated gdx file can be used to fo
*¶
*¶
* GAMS Development Corporation, Formulatio
*¶
$stitle data for single lines, bars, pie¶
set        years / y1998*y2005 /¶
parameter YearDataA(years), YearDataB(year
¶
YearDa
YearDa
YearDa
¶
¶
$stit
set
parame
scalar
¶
delta
Loop(p
   Po
   Po
   x =
   );¶
¶
¶
$stit
set



parame

¶
vector
```

Entry	Symbol	Type	Dim	Nr Elem
10	GanttData	Par	3	14
4	Points	Par	2	200
8	Scatter2D	Par	2	40
9	Scatter3D	Par	2	60
13	ScenarioData	Par	2	136,000
12	StockData	Par	3	800
11	Surface	Par	2	2,500
5	Vector2D	Par	2	80
6	Vector2Db	Par	2	80
7	Vector3D	Par	2	120
1	YearDataA	Par	1	8
2	YearDataB	Par	1	8
3	YearDataC	Par	1	8

StockData

Plane Index (empty)

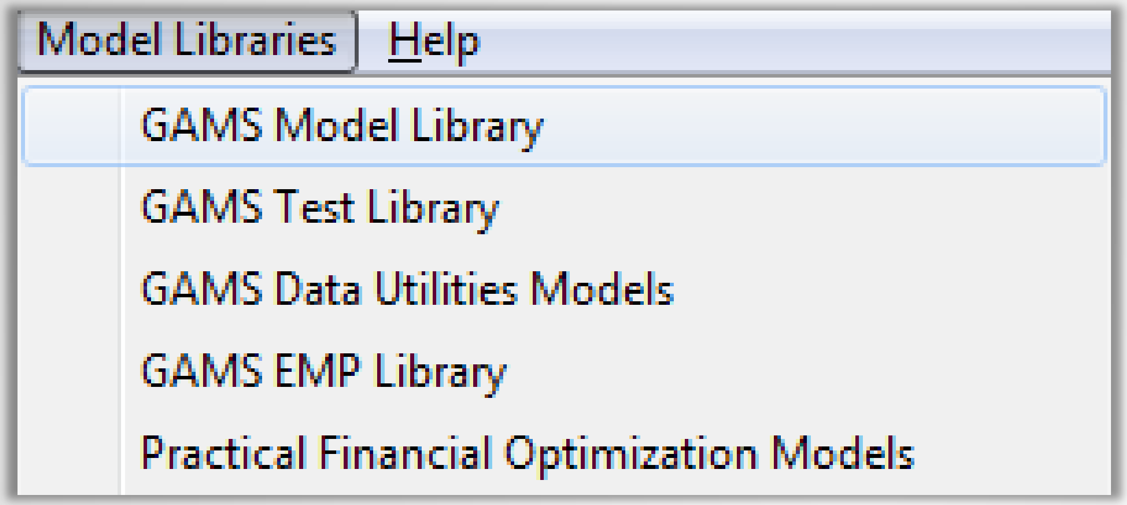
t1	IBM	x0	38742.5472285417
		y0	100
	DELL	x0	38742.5472285417
		y0	100
	HP	x0	38742.5472285417
		y0	100
	SUN	x0	38742.5472285417
		y0	100
t2	IBM	x0	38743.5472285417

Reset · Sort · Decimals ◀ ▶ Max · ☑ Squeeze · Ordering: 1

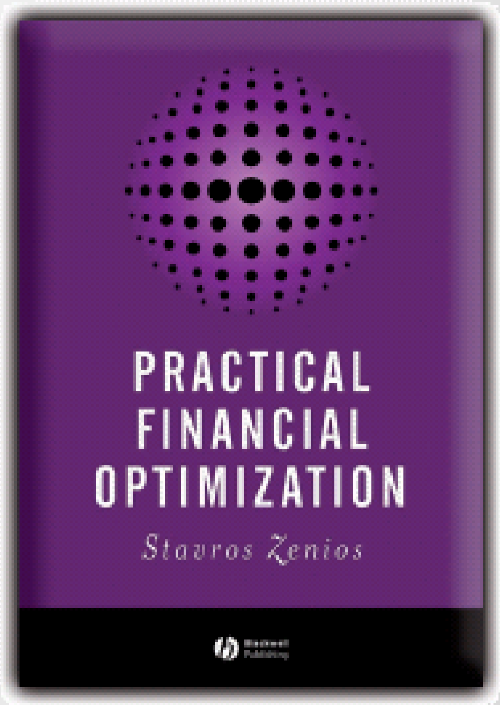
```
--- Job chartdat.gms Start 05/05/06 13:08:00
GAMS Rev 145  Copyright (C) 1987-2006 GAMS Development. All right
Licensee: Franz Nelissen                                  S051012/
          GAMS Software GmbH
--- Starting compilation
--- chartdat.gms(133) 3 Mb
--- Starting execution
--- chartdat.gms(126) 7 Mb
--- Putfile f D:\support\testchart.gch
*** Status: Normal completion
--- Job chartdat.gms Stop 05/05/06 13:08:01 elapsed 0:00:01.422
```

Close · Open Log · ☐ Summary only · ☑ Update

Free Model Libraries



➡ More than 1250 models!



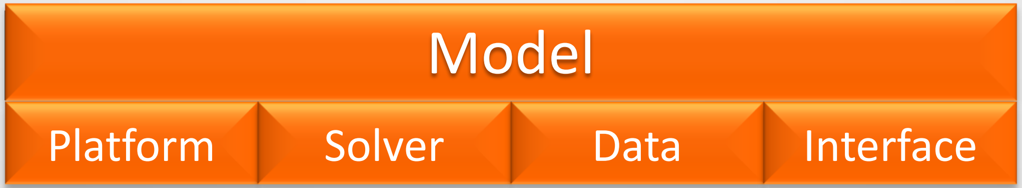
Design Principles

1

- Simple modeling language with a balanced mix of declarative and procedural elements

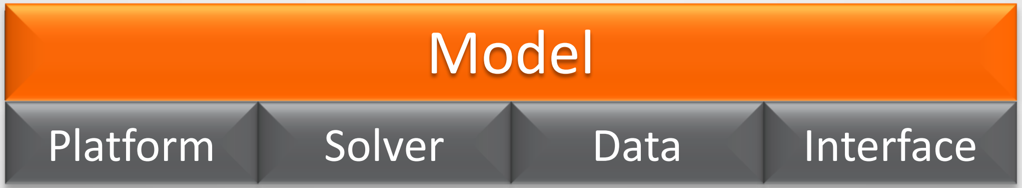
2

- Open architecture and interfaces to other systems, independent layers



Simple Declarative Language

1. Few basic language elements: sets, parameters, variables, equations, models
2. Language similar to mathematical notation
3. Easy to learn
4. Model is executable description of the problem
5. Lot‘s of code optimization under the hood



Mix of Declarative and Procedural Elements

Procedural elements like loops, for, if, macros and functions

Allow to build complex problem algorithms within GAMS

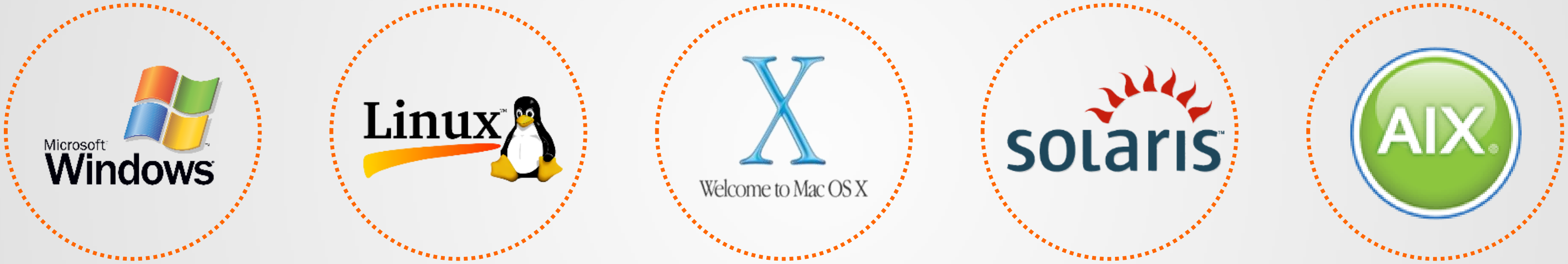
Interaction with other systems:

- Job control
- Data exchange

Model

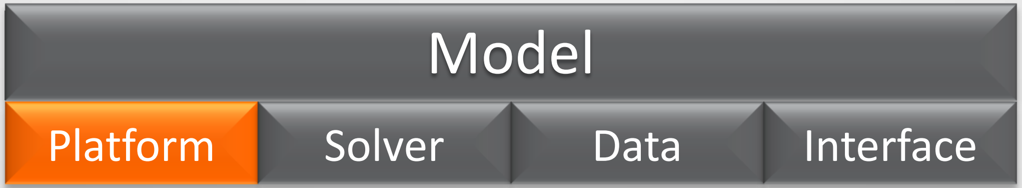
Platform | Solver | Data | Interface

Independence of Model and Operating System



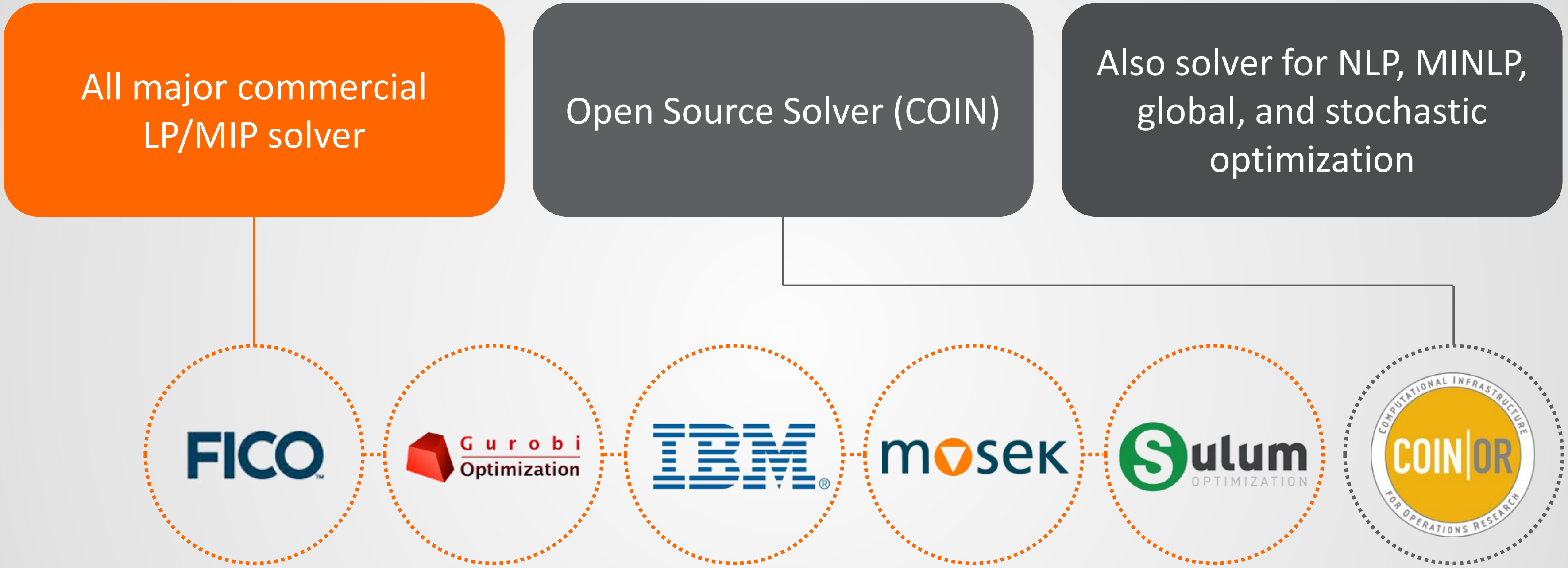
Platforms supported by GAMS:

➡Models can be moved between platforms with ease!



Independence of Model and Solver

One environment for a wide range of model types and solvers


All major commercial LP/MIP solver

Open Source Solver (COIN)

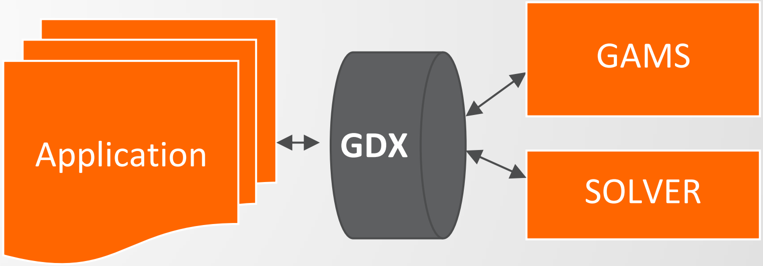
Also solver for NLP, MINLP, global, and stochastic optimization

➡ Switching between solvers with one line of code!

Independence of Model and Data

- Declarative Modeling
- ASCII: Initial model development

- GDX: Data layer (“contract”) between GAMS and applications
  - Platform independent
  - No license required
  - Direct GDX interfaces and general API
  - …



Model

Platform | Solver | Data | Interface

Independence of Model and User Interface

API's

- *Low Level*
- **Object Oriented**: .Net, Java, Python
- No modeling capability: Model is written in GAMS
- Wrapper class that encapsulates a GAMS model

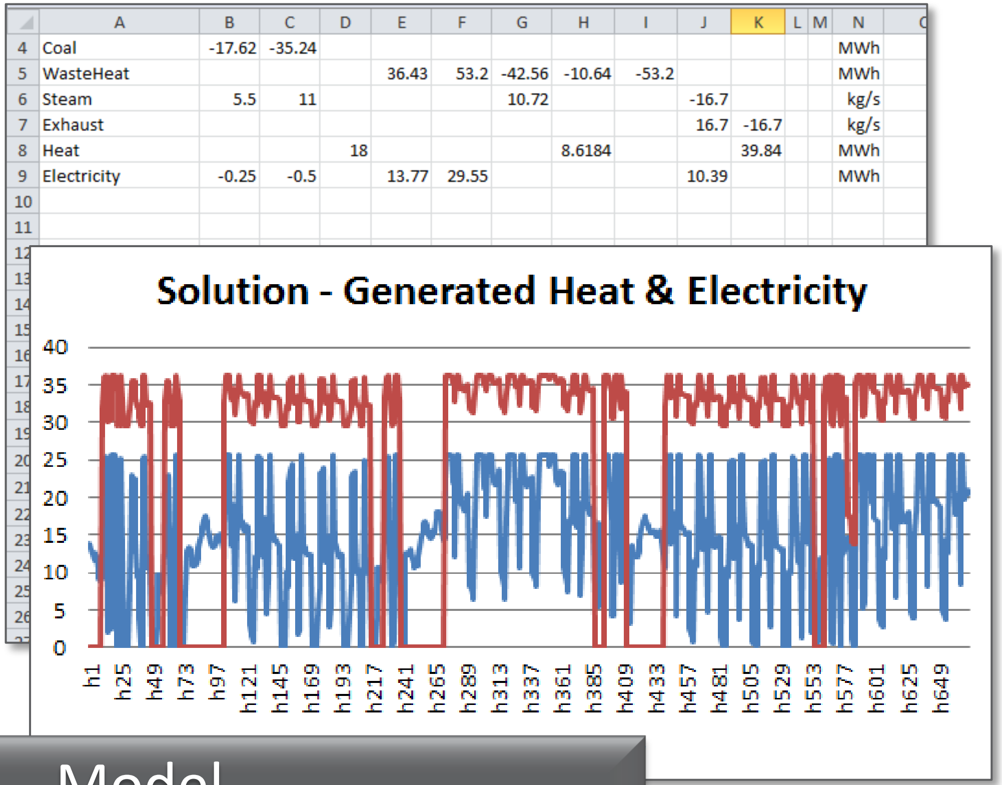
Model

Platform | Solver | Data | Interface

Smart Links to other Applications

- User keeps working in his productive tool environment
- Application accesses all optimization capabilities of GAMS through API
- Visualization and analysis of model data and results in the application

MS Excel



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
4	Coal	-17.62	-35.24											MWh
5	WasteHeat				36.43	53.2	-42.56	-10.64	-53.2					MWh
6	Steam	5.5	11				10.72			-16.7				kg/s
7	Exhaust									16.7	-16.7			kg/s
8	Heat			18				8.6184			39.84			MWh
9	Electricity	-0.25	-0.5		13.77	29.55				10.39				MWh
10														
11														

Model

Platform | Solver | Data | Interface

Smart Links to other Applications

- User keeps working in his productive tool environment
- Application accesses all optimization capabilities of GAMS through API
- Visualization and analysis of model data and results in the application

MatLab

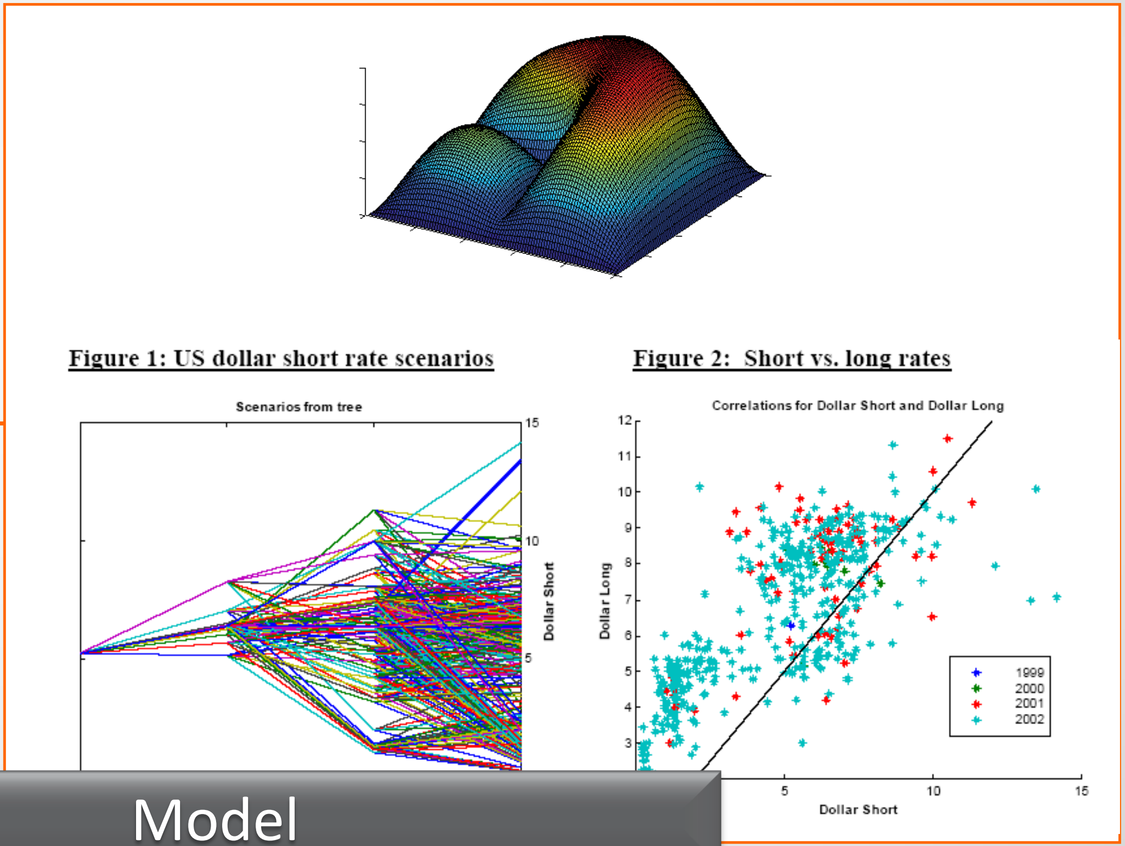


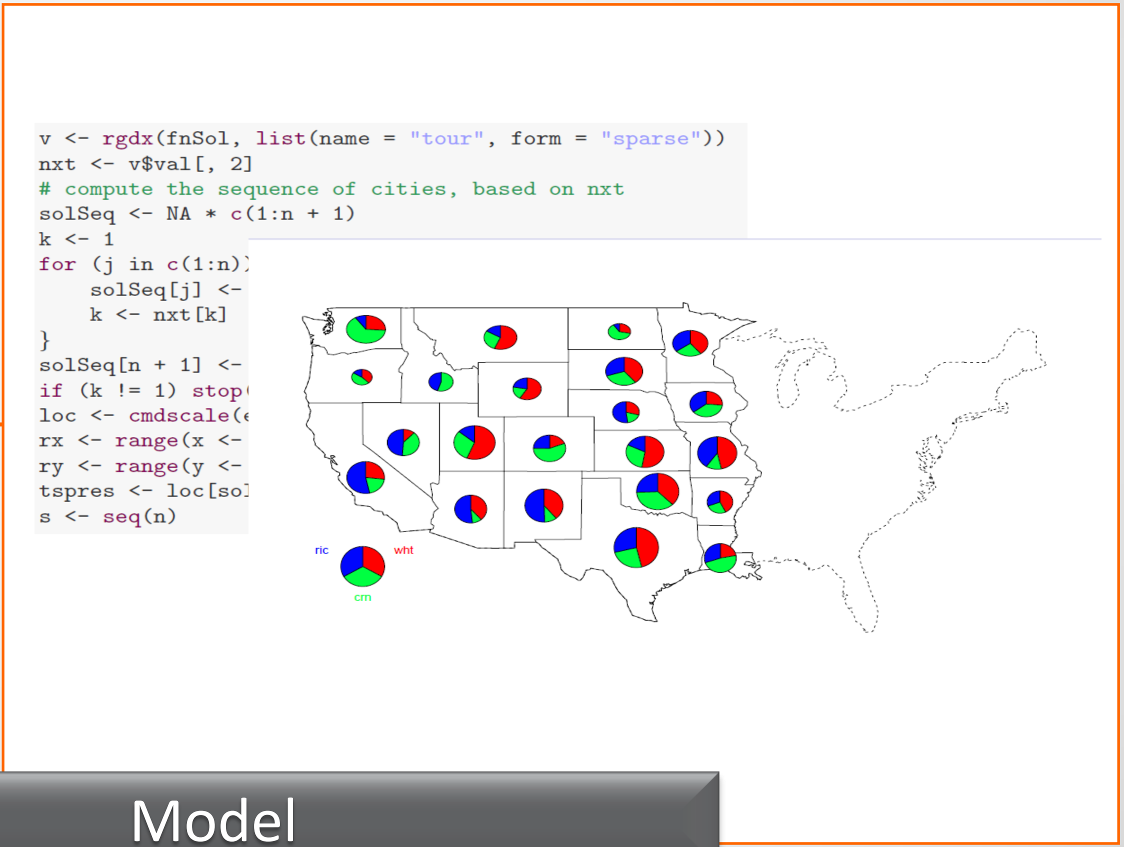
Figure 1: US dollar short rate scenarios

Figure 2: Short vs. long rates

Smart Links to other Applications

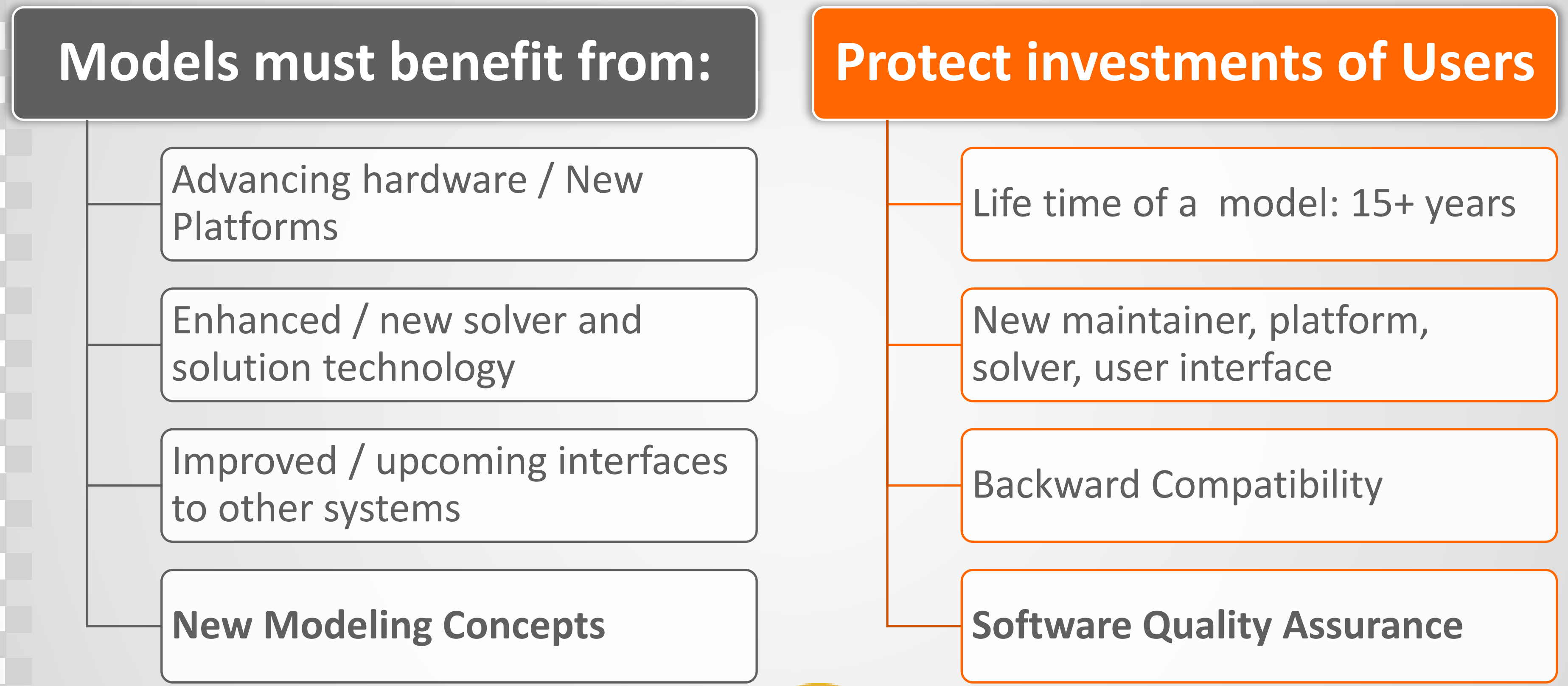
- User keeps working in his productive tool environment
- Application accesses all optimization capabilities of GAMS through API
- Visualization and analysis of model data and results in the application

R



```
v <- rgdx(fnSol, list(name = "tour", form = "sparse"))
nxt <- v$val[, 2]
# compute the sequence of cities, based on nxt
solSeq <- NA * c(1:n + 1)
k <- 1
for (j in c(1:n))
    solSeq[j] <-
    k <- nxt[k]
}
solSeq[n + 1] <-
if (k != 1) stop(
loc <- cmdscale(
rx <- range(x <-
ry <- range(y <-
tspres <- loc[sol
s <- seq(n)
```

Striving for Innovation and Compatibility



Models must benefit from:

- Advancing hardware / New Platforms
- Enhanced / new solver and solution technology
- Improved / upcoming interfaces to other systems
- **New Modeling Concepts**

Protect investments of Users

- Life time of a model: 15+ years
- New maintainer, platform, solver, user interface
- Backward Compatibility
- **Software Quality Assurance**

Agenda

What is GAMS?

Enhancements in GAMS

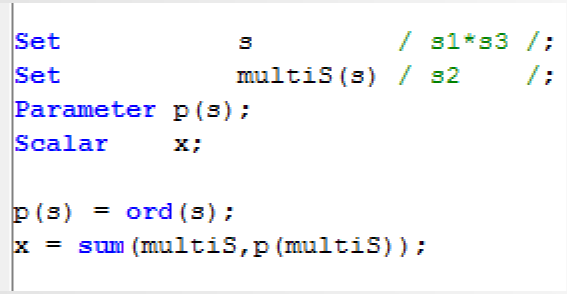
- Singleton Sets
- Value at Risk
- GUSS & Grid
- Obfuscated Save/Restart Files

Enhancements around GAMS

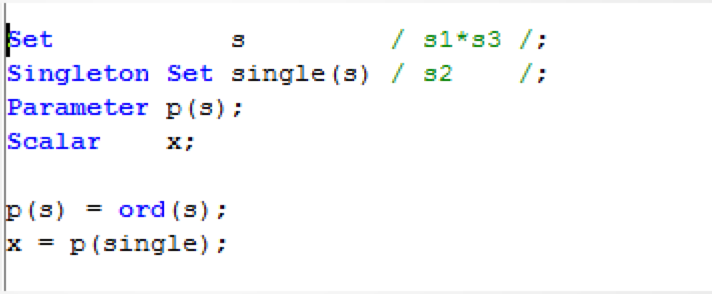
Summary

Singleton Sets

- Special GAMS **Set**
- Has at most one element (empty **Singleton Sets** are also valid)
- No need to be controlled by controlling index nor indexed operator



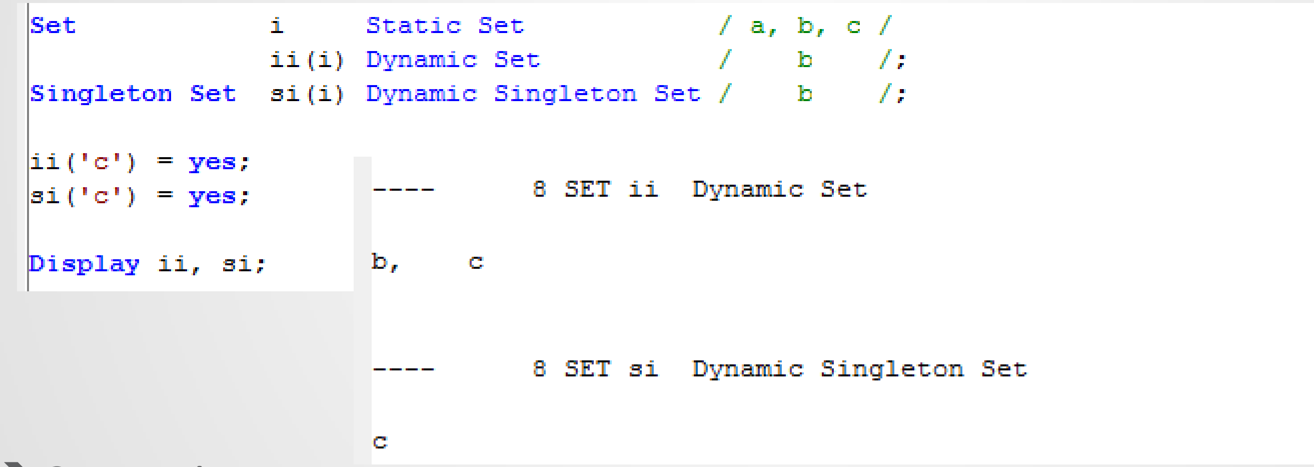
```
Set           s         / s1*s3 /;
Set           multiS(s) / s2    /;
Parameter p(s);
Scalar    x;

p(s) = ord(s);
x = sum(multiS,p(multiS));
```



```
Set           s         / s1*s3 /;
Singleton Set single(s) / s2    /;
Parameter p(s);
Scalar    x;

p(s) = ord(s);
x = p(single);
```

Singleton Sets

- Special GAMS **Set**
- Has at most one element (empty **Singleton Sets** are also valid)
- No need to be controlled by controlling index nor indexed operator
- Any assignment to a **Singleton Set** first clears or empties the **Set**, no explicit clear is necessary



```
Set              i     Static Set            / a, b, c /
                 ii(i) Dynamic Set           /    b    /;
Singleton Set    si(i) Dynamic Singleton Set /    b    /;

ii('c') = yes;
si('c') = yes;

Display ii, si;
```

```
----      8 SET ii  Dynamic Set

b,    c


----      8 SET si  Dynamic Singleton Set

c
```

➔Convenience

➔“Security”

Stochastic Programming

- The Extended Mathematical Programming (EMP) framework is used to replace parameters in the model by random variables
- Support for Multi-stage recourse problems and chance constraint models
- Easy to add uncertainty to existing deterministic models, to either use specialized algorithms or create Deterministic Equivalent (new free solver DE)
- More information: http://www.gams.com/dd/docs/solvers/empsp.pdf

Simple Newsboy Problem

```
*             LostSales = demand - UnitsSold
lSales..      L =e= d - S;
*             Inventory = UnitsBought - UnitsSold
Inv..         I =e= X - S;
*             Profit, to be maximized
Profit..      Z =e= r*S - c*X - h*I - p*L;
```

```
Model nb / all /;
Solve nb max z use lp;
```

Simple Newsboy Problem

```
*              LostSales = demand - UnitsSold
lSales..       L =e= d - S;
*              Inventory = UnitsBought - UnitsSold
Inv..          I =e= X - S;
*              Profit, to be maximized
Profit..       Z =e= r*S - c*X - h*I - p*L;
```

```
$onEcho > %emp.info%
* Make d uncertain
randvar d normal 45 10
* Define nondefault stages
stage 2 d I L S
stage 2 lSales Inv
$offEcho
```

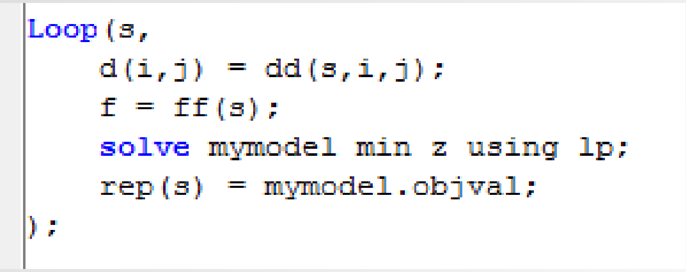
```
Set scen         / s1*s6 /;
Parameter
    s_d(scen) Demand
    s_x(scen) Units bought;
Set
 dict / scen.scenario.''
        d    .randvar .s_d
        x    .level   .s_x /;
```

```
Model nb / all /;
Solve nb max z use emp scenario dict;
```

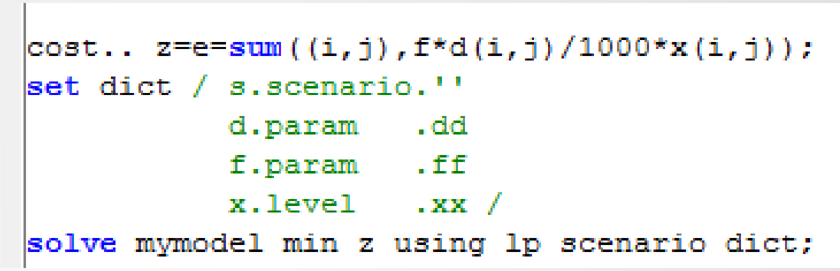
Keywords

- Risk Measures
  - `cVaR`
    - Conditional Value at Risk
  - `ExpectedValue`
    - Expected Value
  - `VaR`
    - Value at Risk
- `Chance`
  - Chance Constraints
- `JRandVar`
  - Random Variables with joint distribution
- `RandVar`
  - Discrete and parametric random variables
- `Sample`
  - Customize samples taken from random variables with continuous distribution
- `Stage`
  - Stage of (random) variables and equations

GUSS & Grid



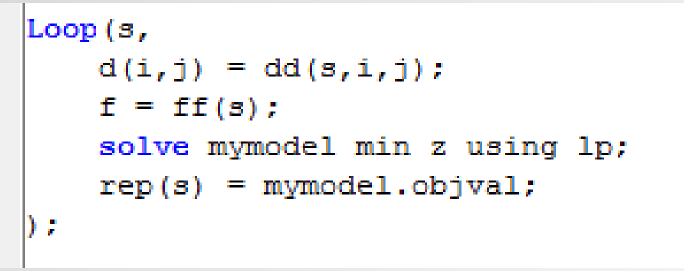
```
Loop(s,
    d(i,j) = dd(s,i,j);
    f = ff(s);
    solve mymodel min z using lp;
    rep(s) = mymodel.objval;
);
```



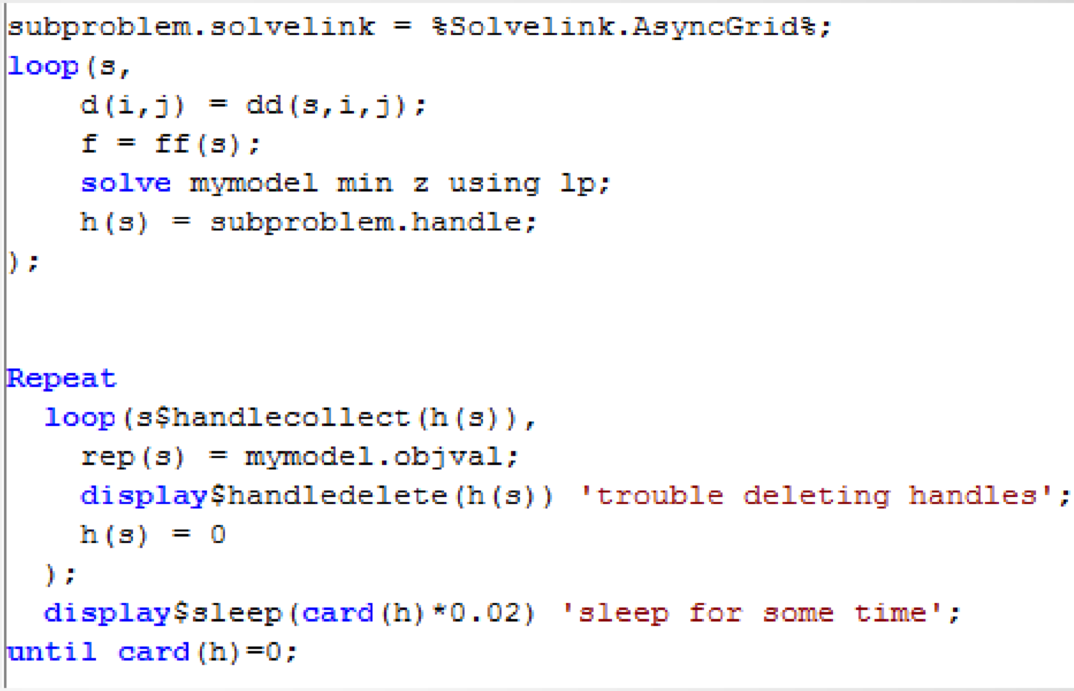
```
cost.. z=e=sum((i,j),f*d(i,j)/1000*x(i,j));
set dict / s.scenario.''
           d.param   .dd
           f.param   .ff
           x.level   .xx /
solve mymodel min z using lp scenario dict;
```

- Update model data instead of matrix coefficients/rhs
- Hot start (keep the model hot inside the solver and use solver’s best update mechanism)
- Save model generation and solver setup time
- Model rim unchanged from scenario to scenario
- Apriori knowledge of all scenario data

GUSS & Grid



```
Loop(s,
    d(i,j) = dd(s,i,j);
    f = ff(s);
    solve mymodel min z using lp;
    rep(s) = mymodel.objval;
);
```

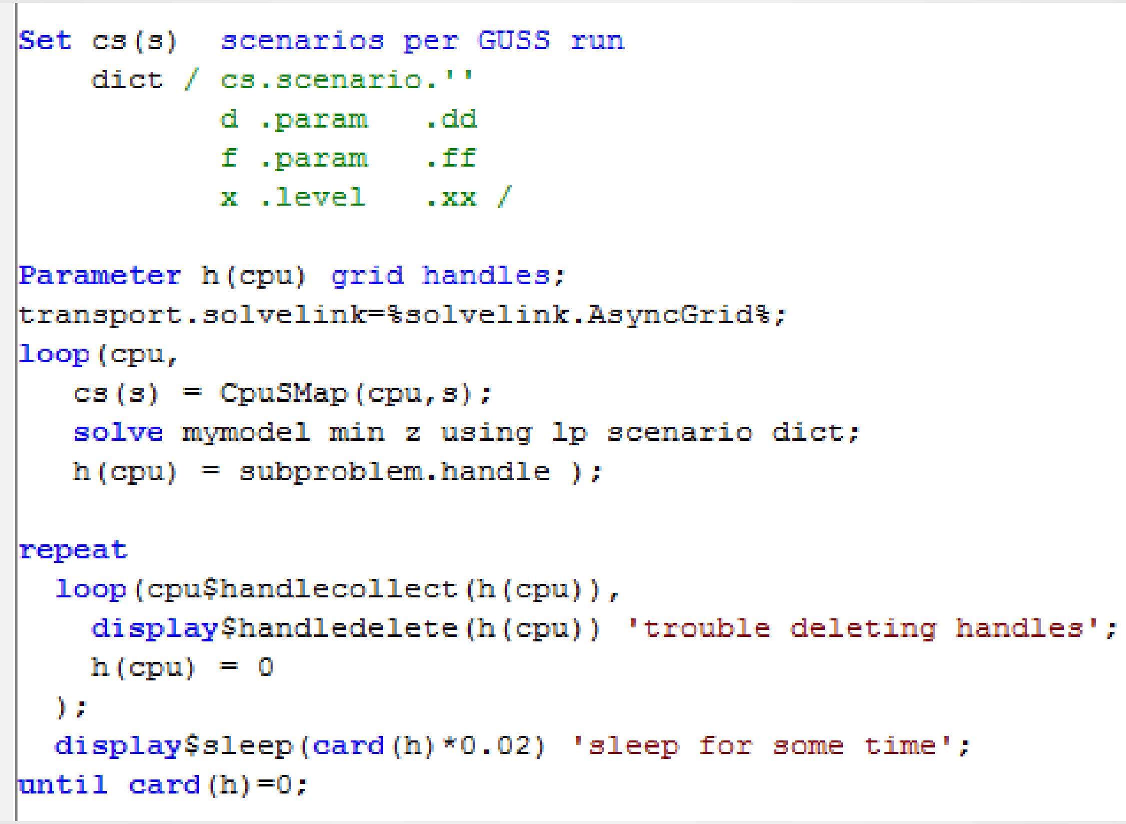


```
subproblem.solvelink = %Solvelink.AsyncGrid%;
loop(s,
    d(i,j) = dd(s,i,j);
    f = ff(s);
    solve mymodel min z using lp;
    h(s) = subproblem.handle;
);


Repeat
  loop(s$handlecollect(h(s)),
    rep(s) = mymodel.objval;
    display$handledelete(h(s)) 'trouble deleting handles';
    h(s) = 0
  );
  display$sleep(card(h)*0.02) 'sleep for some time';
until card(h)=0;
```

- Scalable:
  - Support of massive grids, but also
  - Multi-CPU / Multiple cores desktop machines
- Platform independent
- Only minor changes to model required
- Separation of model and solution method

→ Model stays maintainable

GUSS & Grid (Parallel Scenarios)

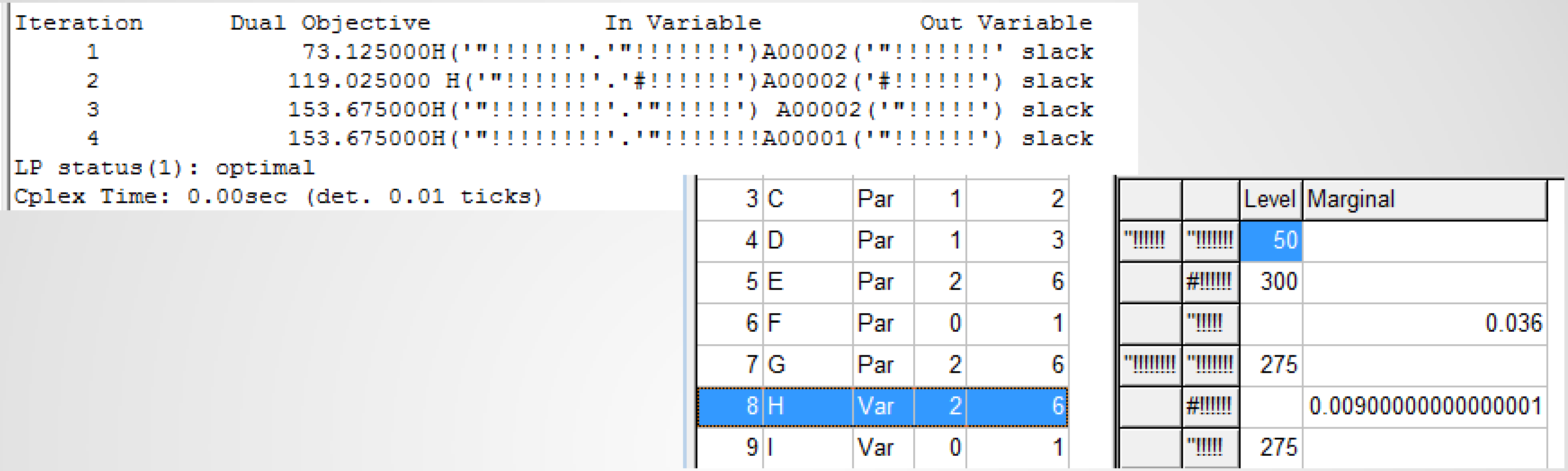


```
Set cs(s)  scenarios per GUSS run
    dict / cs.scenario.''
           d .param   .dd
           f .param   .ff
           x .level   .xx /

Parameter h(cpu) grid handles;
transport.solvelink=%solvelink.AsyncGrid%;
loop(cpu,
   cs(s) = CpuSMap(cpu,s);
   solve mymodel min z using lp scenario dict;
   h(cpu) = subproblem.handle );

repeat
  loop(cpu$handlecollect(h(cpu)),
    display$handledelete(h(cpu)) 'trouble deleting handles';
    h(cpu) = 0
  );
  display$sleep(card(h)*0.02) 'sleep for some time';
until card(h)=0;
```

Obfuscated Save/Restart File

- Special Save/Restart File
- Symbol and UEL names are obfuscate
- New options `saveobfuscate` (`so`) and `xsaveobfuscate` (`xso`) to generate obfuscated Save/Restart file (regular or compressed)
- New option `restartNamed` (`rn`) to bring back original names when restarting from an obfuscated Save/Restart file

Obfuscated Save/Restart File



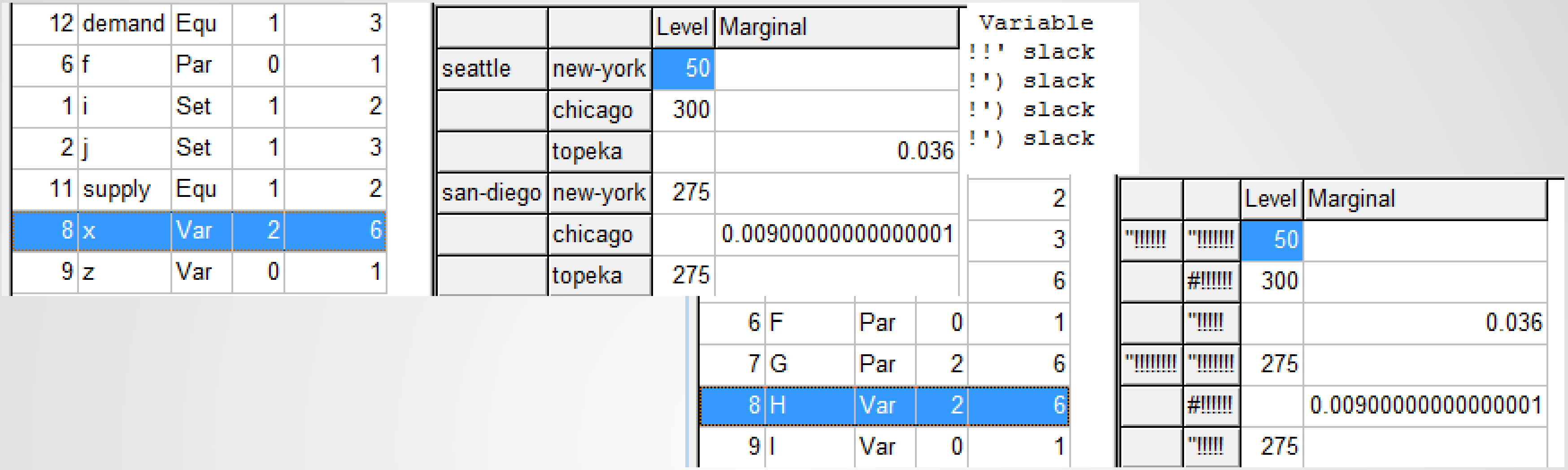
```
Iteration      Dual Objective            In Variable           Out Variable
     1              73.125000H('"!!!!!!'.'"!!!!!!!')A00002('"!!!!!!!' slack
     2             119.025000 H('"!!!!!!'.'#!!!!!!')A00002('#!!!!!!') slack
     3             153.675000H('"!!!!!!!!'.'"!!!!!') A00002('"!!!!!') slack
     4             153.675000H('"!!!!!!!!'.'"!!!!!!!A00001('"!!!!!!') slack
LP status(1): optimal
Cplex Time: 0.00sec (det. 0.01 ticks)
```

3	C	Par	1	2
4	D	Par	1	3
5	E	Par	2	6
6	F	Par	0	1
7	G	Par	2	6
8	H	Var	2	6
9	I	Var	0	1

		Level	Marginal
"!!!!!!	"!!!!!!!	50	
	#!!!!!!	300	
	"!!!!!		0.036
"!!!!!!!!!	"!!!!!!!	275	
	#!!!!!!		0.00900000000000001
	"!!!!!	275	

- Intended use:
  - Compile (only) GAMS model into named and obfuscated save file:
    `$call gams trnsport a=c s=0named saveobfuscate=0anon`
  - Move obfuscated save file to non-secure machine and execute it there:
    `$call gams dummy r=0anon s=1anon gdx=demo`

Obfuscated Save/Restart File



12	demand	Equ	1	3
6	f	Par	0	1
1	i	Set	1	2
2	j	Set	1	3
11	supply	Equ	1	2
8	x	Var	2	6
9	z	Var	0	1

		Level	Marginal
seattle	new-york	50	
	chicago	300	
	topeka		0.036
san-diego	new-york	275	
	chicago		0.00900000000000001
	topeka	275	

```
 Variable
!!' slack
!') slack
!') slack
!') slack
```

				2
				3
				6
6	F	Par	0	1
7	G	Par	2	6
8	H	Var	2	6
9	I	Var	0	1

		Level	Marginal
"!!!!!	"!!!!!!	50	
	#!!!!!	300	
	"!!!!		0.036
"!!!!!!!	"!!!!!!	275	
	#!!!!!		0.00900000000000001
	"!!!!	275	

- Intended use:
  - Compile (only) GAMS model into named and obfuscated save file:
    `$call gams trnsport a=c s=0named saveobfuscate=0anon`
  - Move obfuscated save file to non-secure machine and execute it there:
    `$call gams dummy r=0anon s=1anon gdx=demo`
  - Bring new (still obfuscated) save file with results back to safe machine and do continued compilation with reporting and export:
    `$call gams dummy r=1anon restartNamed=0named gdx=res`

➔Data Security

Agenda

What is GAMS?

Enhancements in GAMS

Enhancements around GAMS

- MINLPLib 2
- GAMS Lessons

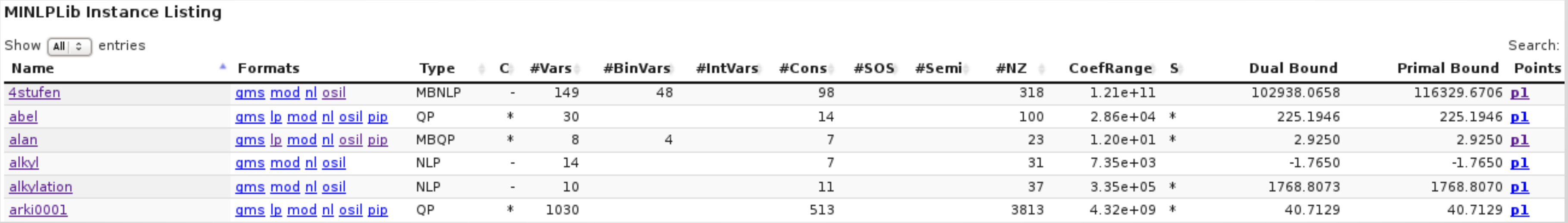
Summary

MINLPLib and GLOBALLib

- http://www.gamsworld.org
- Initiated in 2001 (as part of GamsWorld/MinlpWorld/GlobalWorld):
- M. Bussieck, A. Drud, and A. Meeraus
  MINLPLib – A Collection of Test Models for Mixed-Integer Nonlinear Programming
  INFORMS Journal on Computing 15, 114–119 (2003)
- Frequently used for testing, but also benchmarking
- Scalar GAMS format
- Varying from small scale (great for debugging!) to large scale real world instances (agricultural economics, chemical-, civil-, and electrical engineering, finance, management, OR)
- Intentionally including instances from badly formulated models or different formulations of the same problem
- Including solution points for many instances

MINLPLib 2

- Tasks:
  - Adding new problem instances:
    - Both convex and nonconvex problems
    - (MI)QPs, (MI)QCQPs, and (MI)NLPs
    - Easy solvable, solvable, difficult to solve, but not trivial
  - Categorizing instances
    - Convexity
    - Problem type (quadratic, polynomial, general nonlinear)
    - Function types (powers, exp/log, trigonometric, ...)
    - Solved to global optimality?
  - Providing feasible best known solutions
- Work in progress, alpha version publicly available: http://www.gamsworld.org/minlp/minlplib2/html/index.html



MINLPLib Instance Listing

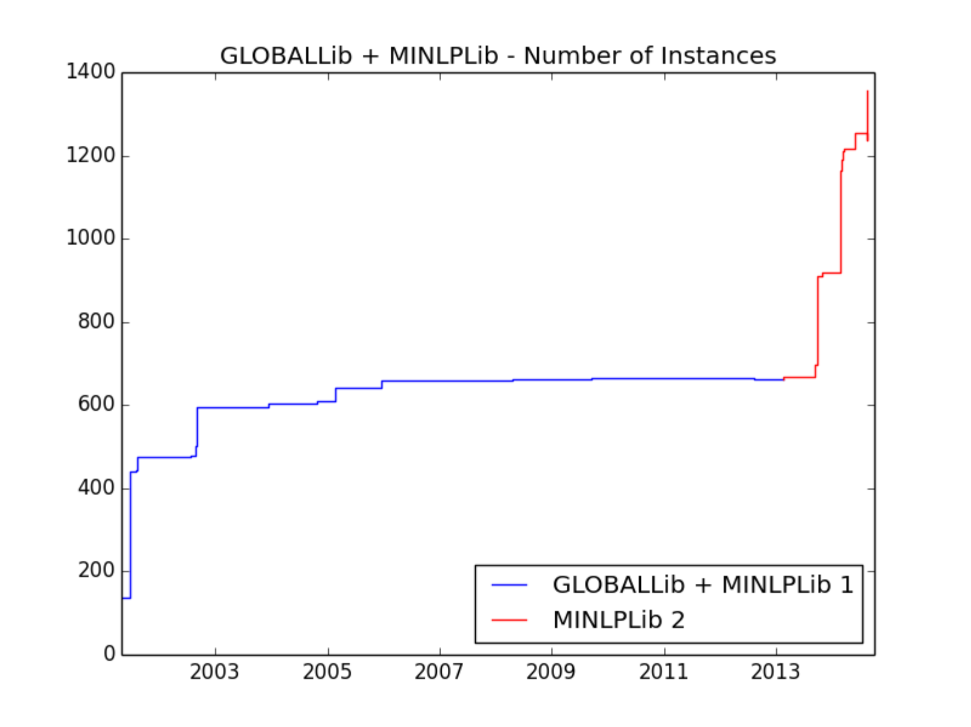
Show All entries

Search:

Name	Formats	Type	C	#Vars	#BinVars	#IntVars	#Cons	#SOS	#Semi	#NZ	CoefRange	S	Dual Bound	Primal Bound	Points
4stufen	gms mod nl osil	MBNLP	-	149	48		98			318	1.21e+11		102938.0658	116329.6706	p1
abel	gms lp mod nl osil pip	QP	*	30			14			100	2.86e+04	*	225.1946	225.1946	p1
alan	gms lp mod nl osil pip	MBQP	*	8	4		7			23	1.20e+01	*	2.9250	2.9250	p1
alkyl	gms mod nl osil	NLP	-	14			7			31	7.35e+03		-1.7650	-1.7650	p1
alkylation	gms mod nl osil	NLP	-	10			11			37	3.35e+05	*	1768.8073	1768.8070	p1
arki0001	gms lp mod nl osil pip	QP	*	1030			513			3813	4.32e+09	*	40.7129	40.7129	p1

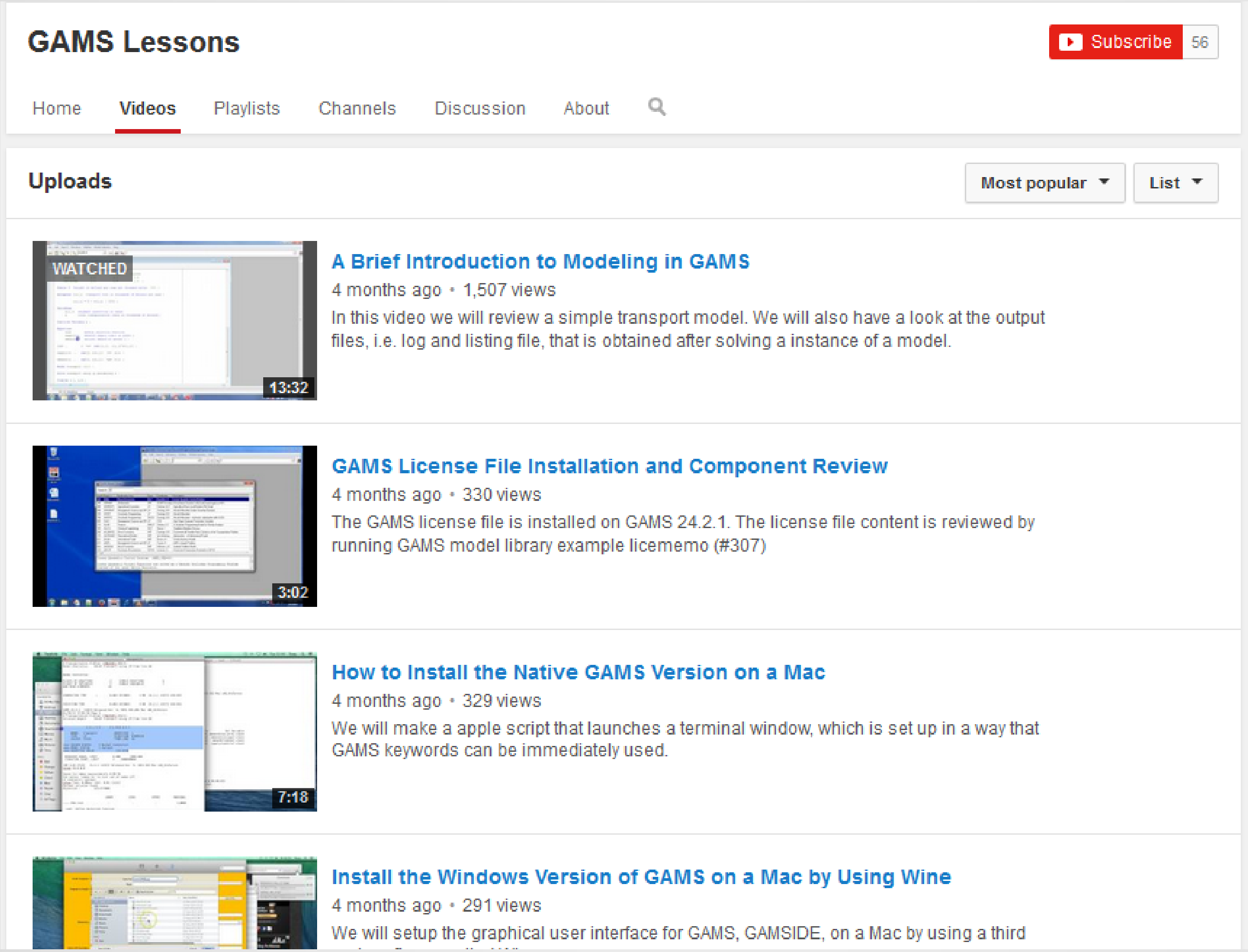
MINLPLib 2 – New Instances

- Mainly from:
  - CMU-IBM open source MINLP project (convex MINLPs)
  - minlp.org
  - POLIP (polynomial MINLPs)
- Future Work:
  - Add more NLPs (merge in PrincetonLib, ...)
  - Semi-automatic identification of duplicates
  - More structure recognition, e.g., second-order cones
  - Define interesting subsets, especially a benchmark set for global solvers



If you have interesting instances, please consider contributing.

YouTube Channel: GAMS lessons



A Brief Introduction to Modeling in GAMS
4 months ago • 1,507 views
In this video we will review a simple transport model. We will also have a look at the output files, i.e. log and listing file, that is obtained after solving a instance of a model.

GAMS License File Installation and Component Review
4 months ago • 330 views
The GAMS license file is installed on GAMS 24.2.1. The license file content is reviewed by running GAMS model library example licememo (#307)

How to Install the Native GAMS Version on a Mac
4 months ago • 329 views
We will make a apple script that launches a terminal window, which is set up in a way that GAMS keywords can be immediately used.

Install the Windows Version of GAMS on a Mac by Using Wine
4 months ago • 291 views
We will setup the graphical user interface for GAMS, GAMSIDE, on a Mac by using a third

www.youtube.com/user/GAMSLessons

Agenda

What is GAMS?

Enhancements in GAMS

Enhancements around GAMS

Summary

Summary

- Improvements on all frontiers:
  - Extended Syntax
    - Singleton Sets
  - Solution concepts
    - Stochastic Programming
  - Multithreading
    - Guss/Grid
  - Data Security
    - Obfuscated Save/Restart files
  - Quality Assurance and Benchmarking
    - MINLPLib 2
  - Documentation
    - YouTube Channel

Fields of Fuel - A Multiplayer, Web-based Simulation Game

A complex system of GAMS models is a centerpiece of this free web-based simulation game, which allows players to explore sustainability issues associated with bioenergy crop production. Biofuels and agronomic experts assisted in creating an accurate and realistic depiction of the system dynamics.

WISCONSIN INSTITUTES FOR DISCOVERY AT THE UNIVERSITY OF WISCONSIN–MADISON

GREAT LAKES BIOENERGY RESEARCH CENTER

- Players take on the role of farmers working to sustainably grow crops as energy resources, earn income and improve ecosystem services.
- Automated 'bot' players communicate with the optimization models via the GAMS Java API to evaluate which options will maximize their overall game score.
- The game can be played in a variety of settings, but was primarly designed for use in high school and undergraduate classes.

For further information please visit http://www.fieldsoffuel.org
or contact Steven Wangen - srwangen@wisc.edu

GAMS

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