An Introduction

Tim Johannessen & Franz Nelißen

GAMS Software GmbH

Company

- Roots: World Bank, 1976
- Went commercial in 1987
- Locations
  - GAMS Development Corporation (Washington)
  - GAMS Software GmbH (Germany)
- Product: The **G**eneral **A**lgebraic **M**odeling **S**ystem

Agenda

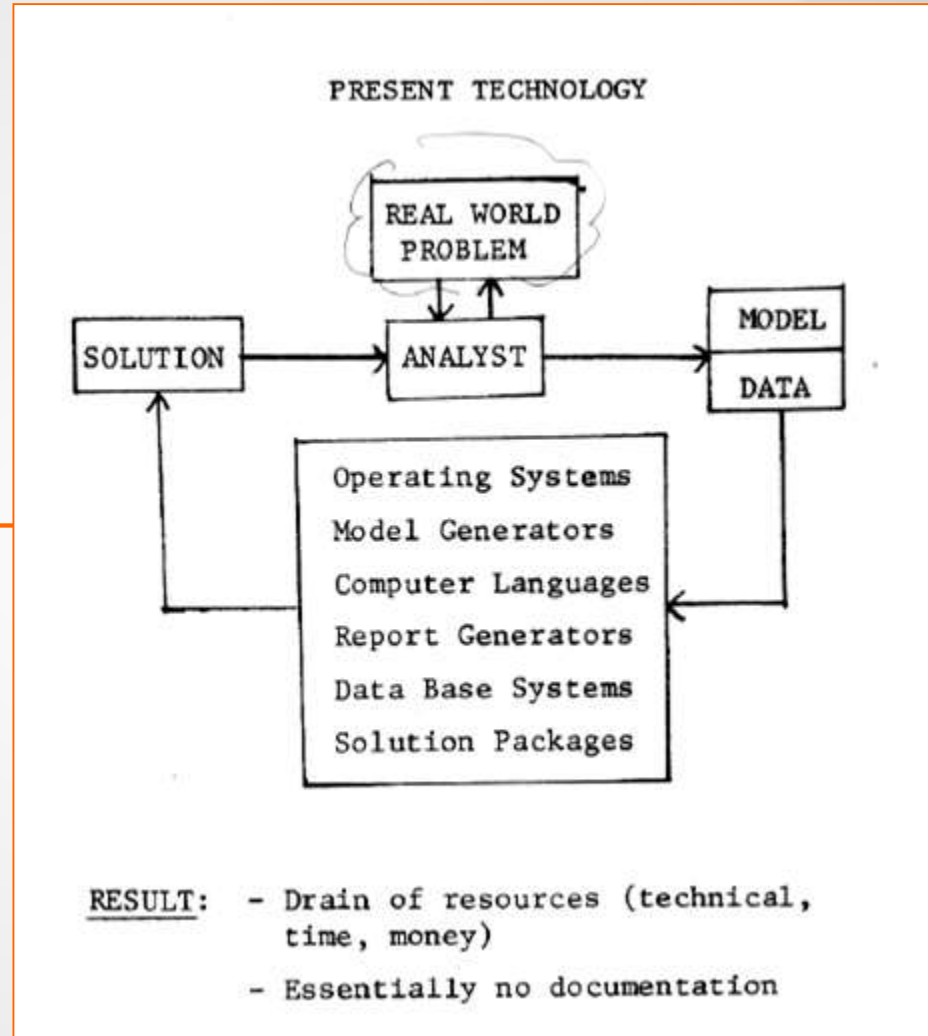
GAMS – Elements and Examples

Scenarios in GAMS

Some Enhancements

1976 - Two World Bank Slides

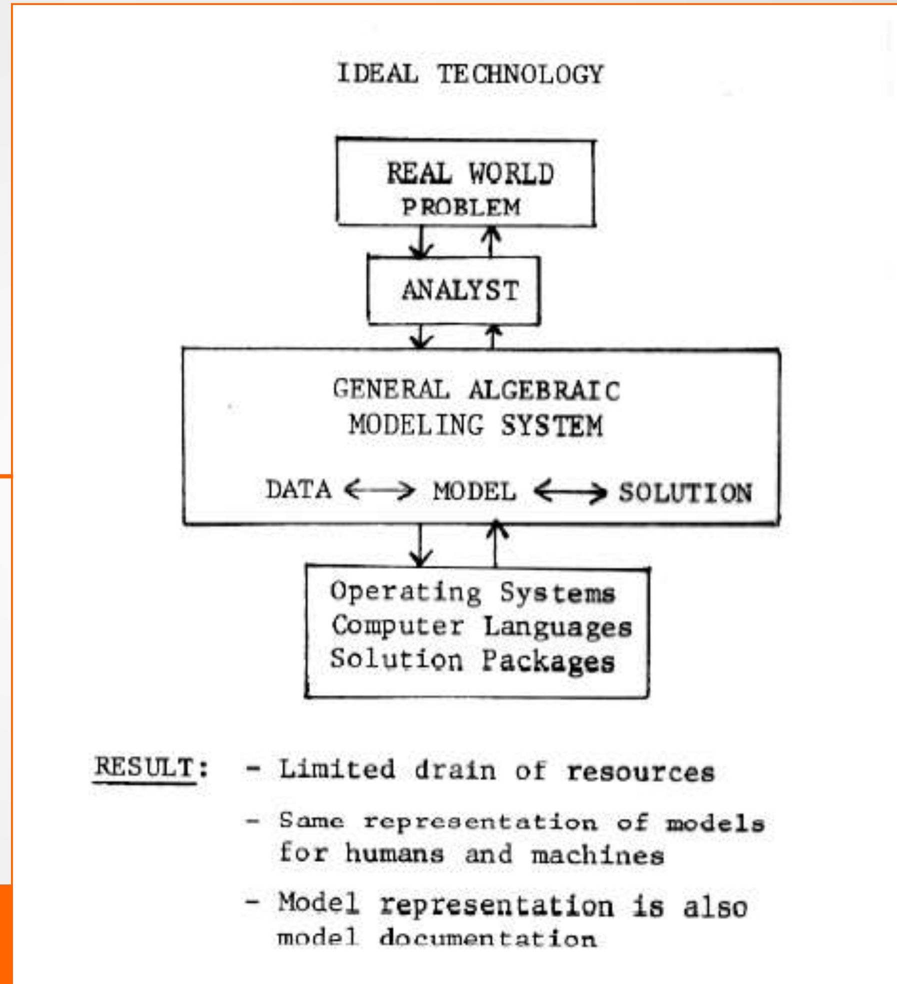
The Origin



GAMS

1976 - Two World Bank Slides

The Vision



GAMS came into being!

What did this give us?

- Simplified model development & maintenance
- Increased productivity tremendously
- Made mathematical optimization available to a broader audience (domain experts)

➢ 2012 INFORMS Impact Prize

GAMS - Evolution

https://en.wikipedia.org/wiki/General_Algebraic_Modeling_System

Languages
Deutsch
Español
Français
Türkçe
Українська
Edit links

Timeline [edit]

- 1976 GAMS idea is presented at the International Symposium on Mathematical Programming (ISMP), Budapest
- 1978 Phase I: GAMS supports linear programming. Supported platforms: Mainframes and Unix Workstations
- 1979 Phase II: GAMS supports nonlinear programming.
- 1987 GAMS becomes a commercial product
- 1988 First PC System (16 bit)
- 1988 Alex Meeraus, the initiator of GAMS and founder of GAMS Development Corporation, is awarded INFORMS Computing Society Prize
- 1990 32 bit Dos Extender
- 1990 GAMS moves to Georgetown, Washington, D.C.
- 1991 Mixed Integer Non-Linear Programs capability (DICOPT)
- 1994 GAMS supports mixed complementarity problems
- 1995 MPSGE language is added for CGE modeling
- 1996 European branch opens in Germany
- 1998 32 bit native Windows
- 1998 Stochastic programming capability (OSL/SE, DECIS)
- 1999 Introduction of the GAMS Integrated development environment (IDE)
- 2000 GAMS World initiative started
- 2001 GAMS Data Exchange (GDX) is introduced
- 2002 GAMS is listed in OR/MS 50th Anniversary list of milestones
- 2003 Conic programming is added
- 2003 Global optimization in GAMS
- 2004 Quality assurance initiative starts
- 2004 Support for Quadratic Constrained programs
- 2005 Support for 64 bit PC Operating systems
- 2006 GAMS supports parallel grid computing
- 2007 GAMS supports open-source solvers from COIN-OR
- 2008 Support for 32 and 64 bit Mac OS X
- 2009 GAMS supports extended mathematical programs (EMP)
- 2010 GAMS is awarded the company award of the German Society of Operations Research (GOR)
- 2012 The Winners of the 2012 INFORMS Impact Prize included Alexander Meeraus. The prize was awarded to the originators of the five most important algebraic modeling languages [1].

1976

Now

Broad User Community and Network

- 11,500+ licenses
- Users: 50% academic, 50% commercial
- GAMS used in more than 120 countries
- Uniform interface to more than 30 solvers

25+ Years
GAMS Development

GAMS

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

GAMS

Foundation of GAMS

Powerful algebraic modeling language

Open architecture and interfaces to other systems, independent layers

Powerful Declarative Language

Similar to mathematical notation

Easy to learn - few basic language elements: sets, parameters, variables, equations, models

Model is executable (algebraic) description of the problem

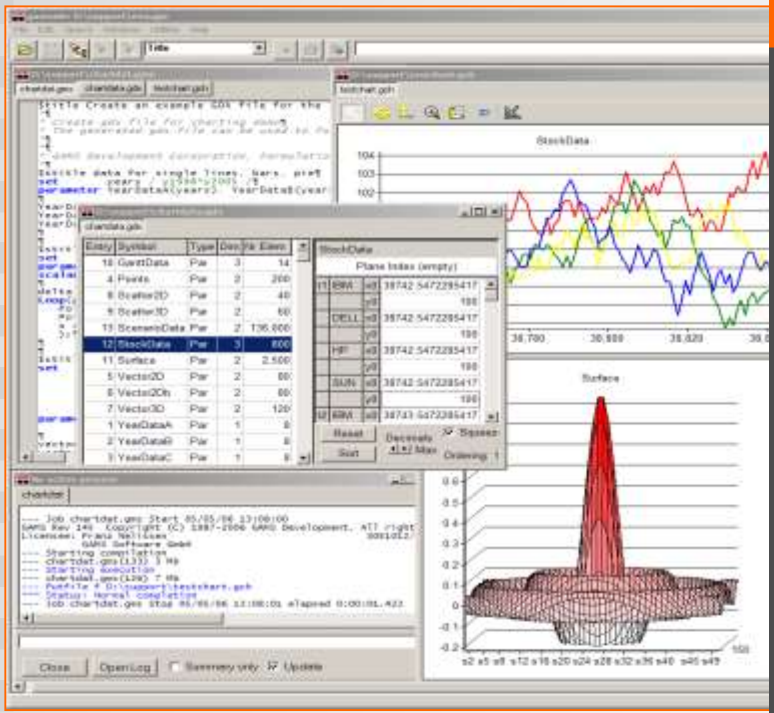
Mix of Declarative and Imperative Elements

Control Flow Statements (e.g. loops, for, if,...), macros and functions

Advantages:

- Build complex problem algorithms within GAMS
- Simplified interaction with other systems:
  - Data exchange
  - GAMS process control

Strong Development Environment

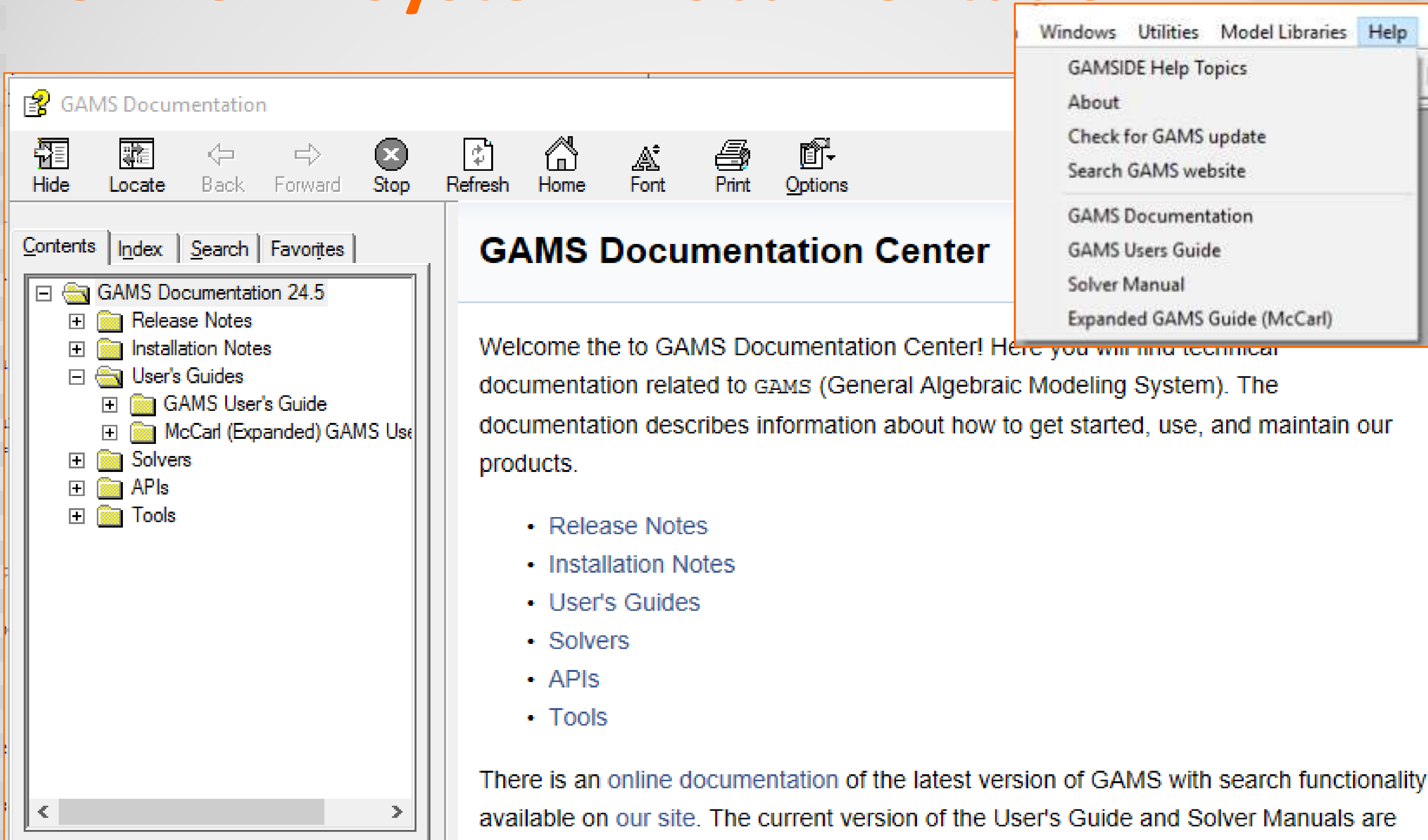


GAMS IDE

- Project management
- Editor / Syntax coloring / Spell checks
- Tree view / Syntax-error navigation
- Model Debugging & Profiling
- Solver selection & setup
- Data viewer
  - Export
  - Charting
- GAMS Processes Control

GAMS

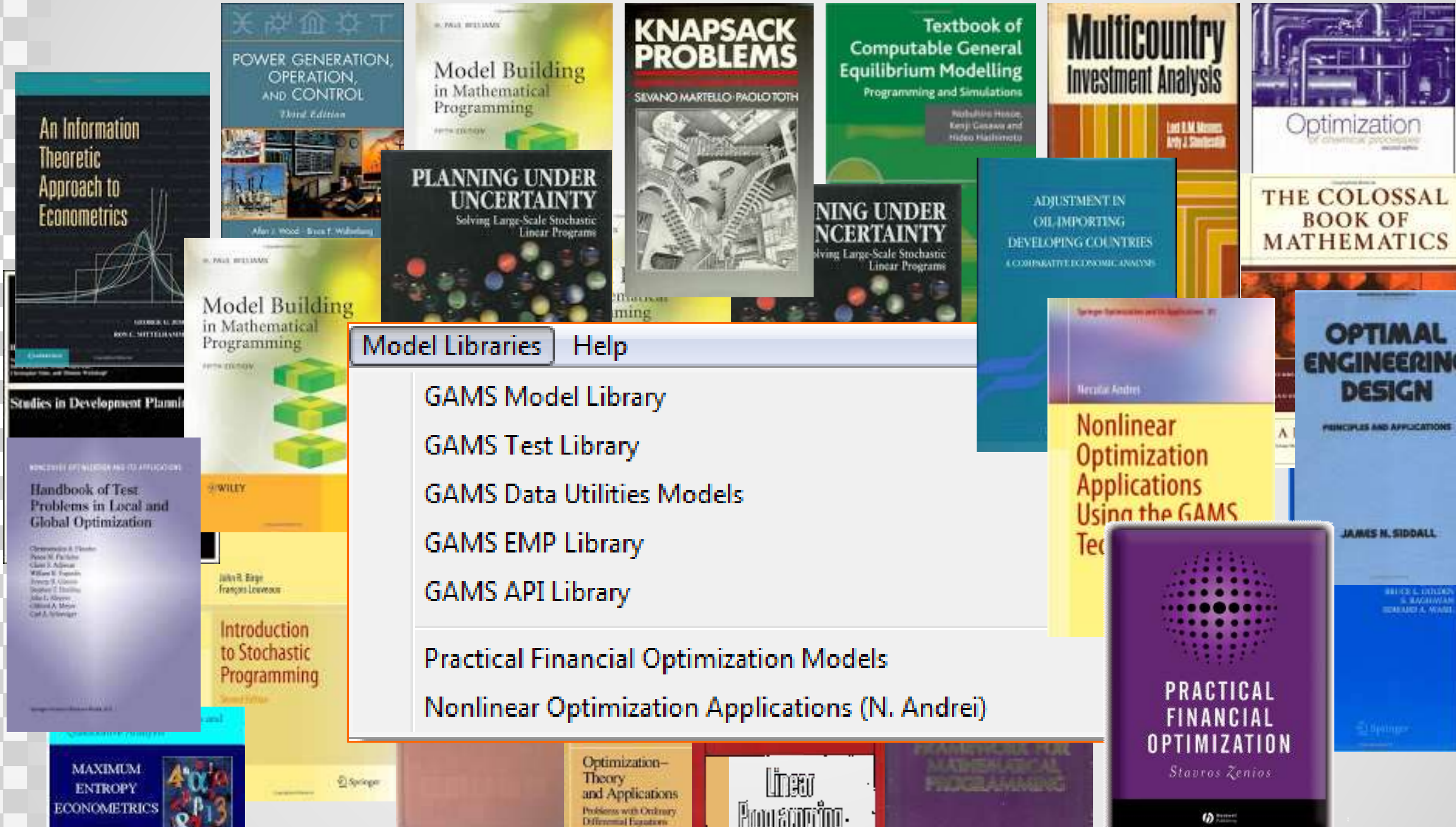
Uniform System Documentation



- Windows Help, HTML, Pdf

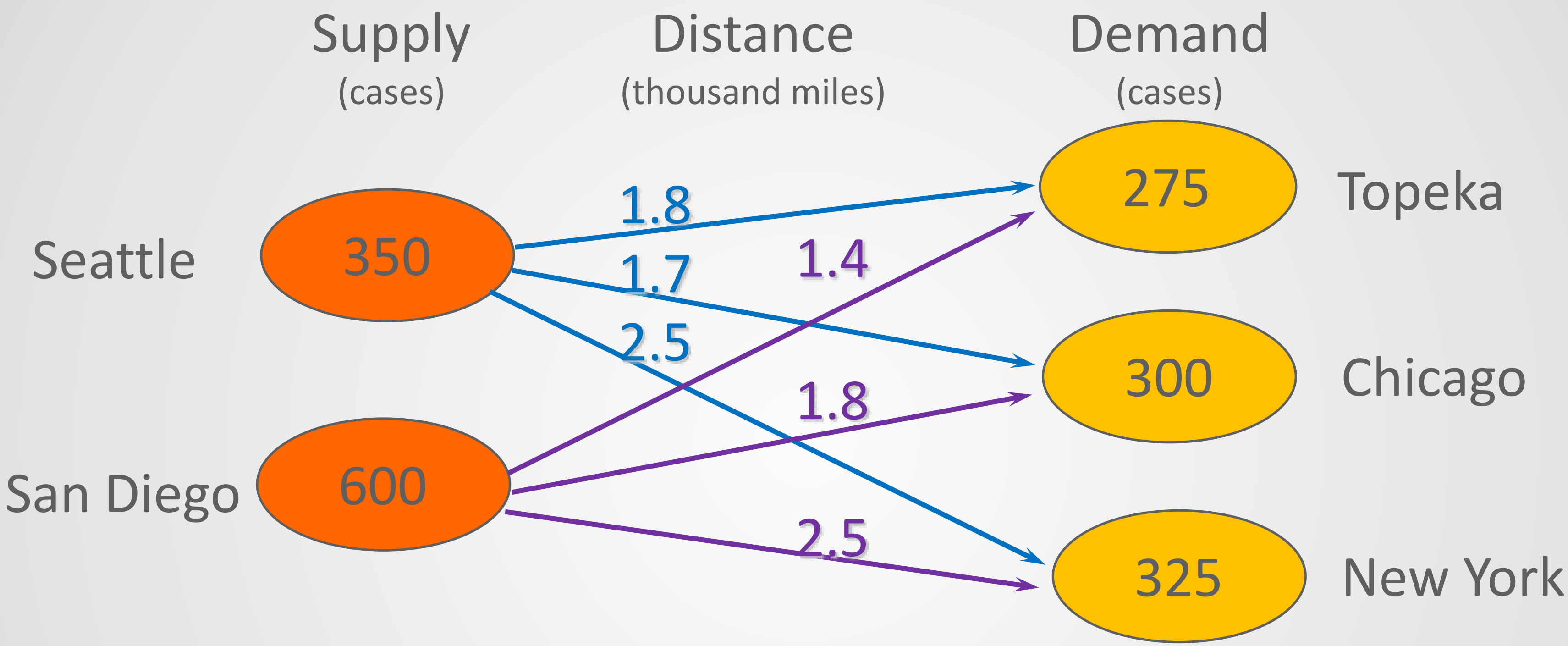
GAMS

Free Model Libraries



- More than 1400 models!

Example: A Transportation Model



Freight: $90 case / thousand miles

Minimize	Transportation cost
subject to	Demand satisfaction at markets
	Supply constraints

GAMS

Example: Mathematical Model

Indices:	i	(Canning plants)
	j	(Markets)
Decision variables:	x_{ij}	(Number of cases to ship)
Data:	c_{ij}	(Transport cost per case)
	a_i	(Capacity in cases)
	b_i	(Demand in cases)

min $\sum_i \sum_j c_{ij} \cdot x_{ij}$ (Minimize total transportation cost)

subject to

$\sum_j x_{ij} \leq a_i \quad \forall\, i$ (Shipments from each plant $\leq$ supply capacity)

$\sum_i x_{ij} \geq b_j \quad \forall\, j$ (Shipments to each market $\geq$ demand)

$x_{ij} \geq 0 \quad \forall i,j$ (Do not ship from market to plant)

$i,j \in \mathbb{N}$

GAMS

Example: GAMS Algebra

D:\Dropbox\GAMS\OR2015-Vienna\pre confernce ws\models\vienna.gms

data.gdx | data.gms | vienna.gms | rep.gdx

```
Sets
   i   canning plants,
   j   markets;
Parameters
       a(i)   capacity of plant i in cases,
       b(j)   demand at market j in cases,
       d(i,j) distance in thousands of miles,
       c(i,j) transport cost in thousands of dollars per case;
Variables
         z       total transportation costs in thousands of dollars,
         x(i,j)  shipment quantities in cases          ;
Positive Variable x;

Equations
       cost        define objective function,
       supply(i)   observe supply limit at plant i,
       demand(j)   satisfy demand at market j;
cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
supply(i) ..    sum(j, x(i,j))  =l=  a(i) ;
demand(j) ..    sum(i, x(i,j))  =g=  b(j) ;

Model modelLP /all/ ;

$include data.gms
* or via gdx
* $gdxin data.gdx
* $load i<d.dim1 j<d.dim2 d b a  c

Solve modelLP using lp minimizing z ;

Parameter rep1(i,j,*)    Shipments between plants and markets
          rep2(*)        Objective value;
rep1(i,j,'lp') = x.l(i,j);
rep2('lp')     = z.l;

display rep1, rep2;
```

GAMS

Example: Data Input

1 `$include data.gms`

conferncee ws\models\data.gms

data.gdx | data.gms | vienna.gms | rep.gdx

```
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   / ;
Table d(i,j)  distance in thousands of miles
                  new-york       chicago      topeka
    seattle          2.5           1.7          1.8
    san-diego        2.5           1.8          1.4  ;
Scalar f  freight in dollars per case per thousand miles  /90/ ;
Parameter c(i,j)  transport cost in thousands of dollars per case ;
          c(i,j) = f * d(i,j) / 1000 ;
```

2
```
$gdxin data.gdx
$load i<d.dim1 j<d.dim2 d b a  c
```

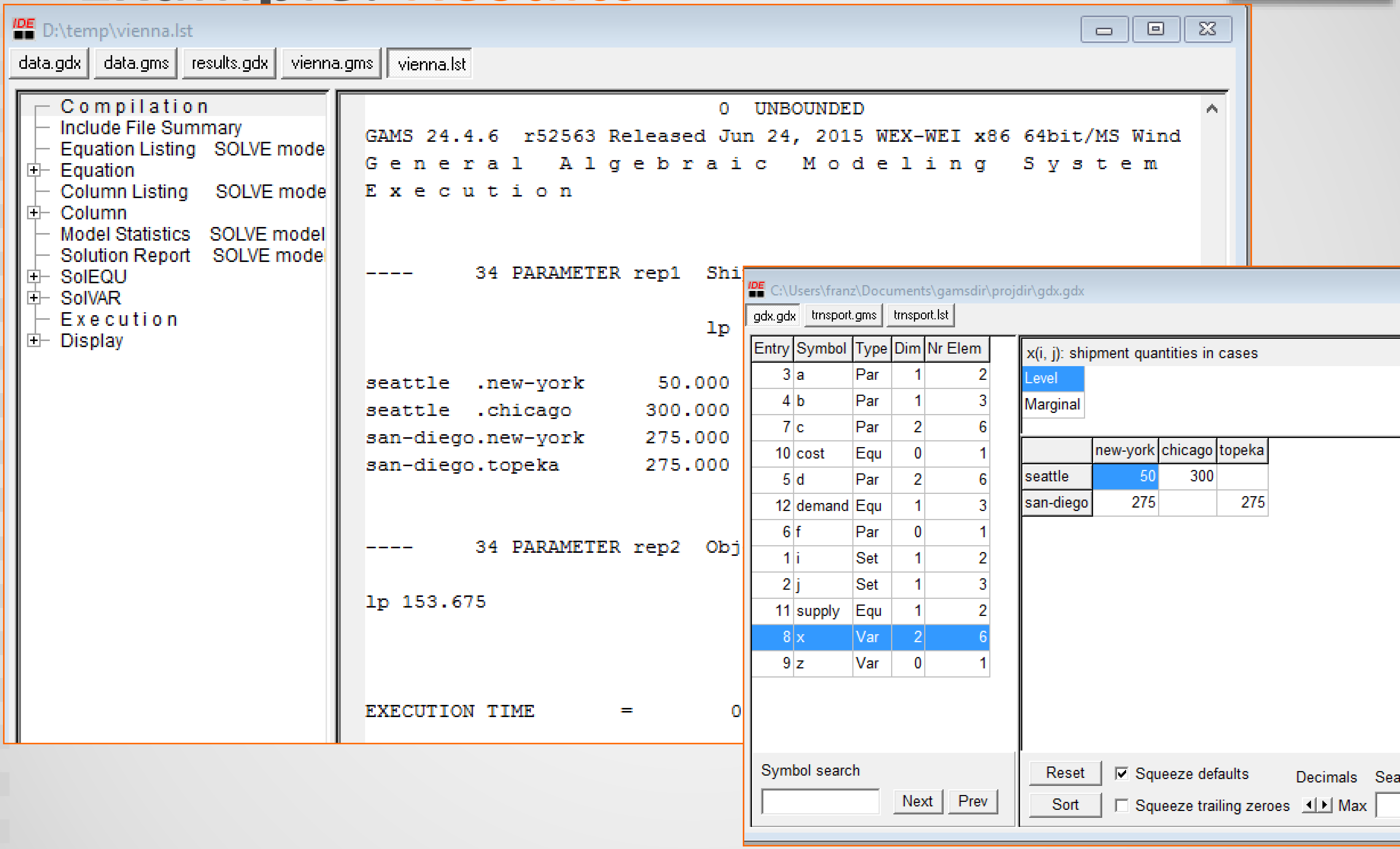
ce ws\models\data.gdx

Entry	Symbol	Type	Dim	Nr Elem
3	a	Par	1	2
4	b	Par	1	3
7	c	Par	2	6
5	d	Par	2	6
6	f	Par	0	1

b(j): demand at market j in cases

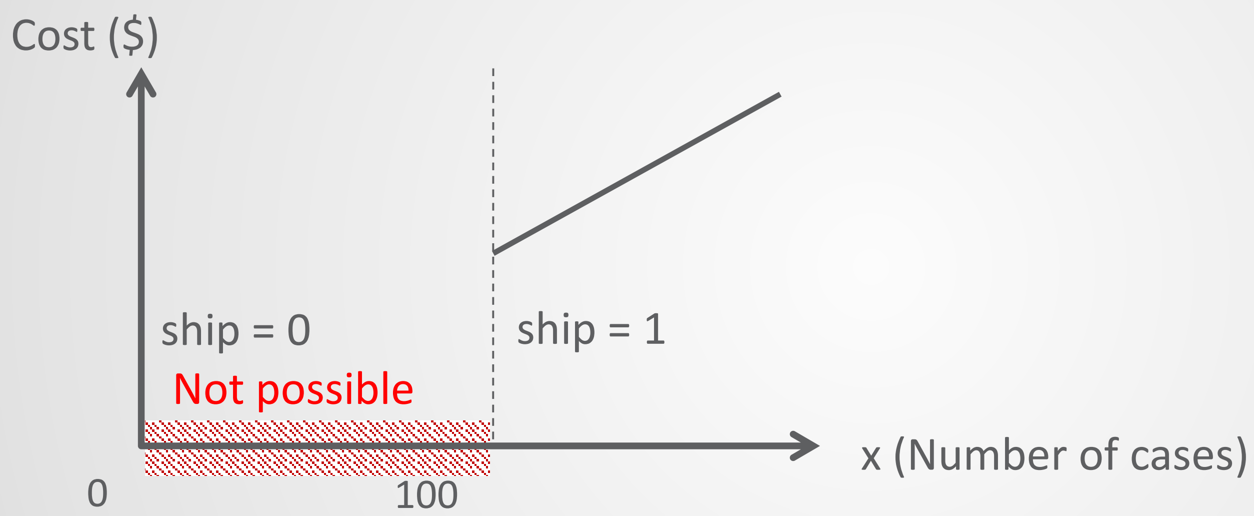
new-yor	325
chicago	300
topeka	275

Example: Results



+ MIP Model: Minimum Shipments

- Shipment volume: **x** (continuous variable)
- Discrete decision: **ship** (binary variable)



add constraints:

$x_{i,j} \geq 100 \cdot ship_{i,j} \quad \forall\, i,j$ (if ship=1, then ship at least 100)

$x_{i,j} \leq bigM \cdot ship_{i,j} \quad \forall\, i,j$ (if ship=0, then do not ship at all)

$ship_{i,j} \in \{0,1\}$

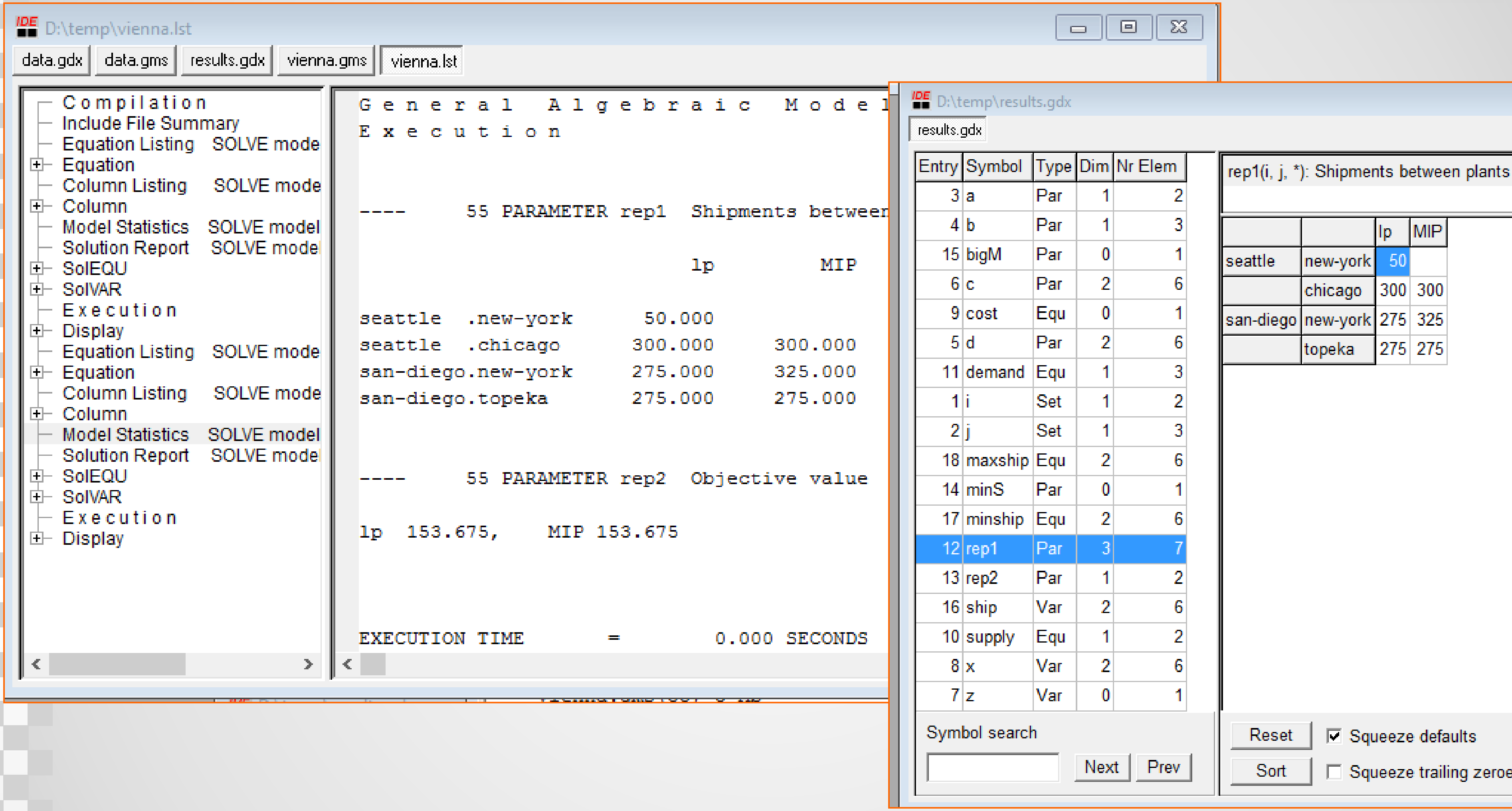
+ MIP Model: GAMS Algebra

D:\temp\vienna.gms

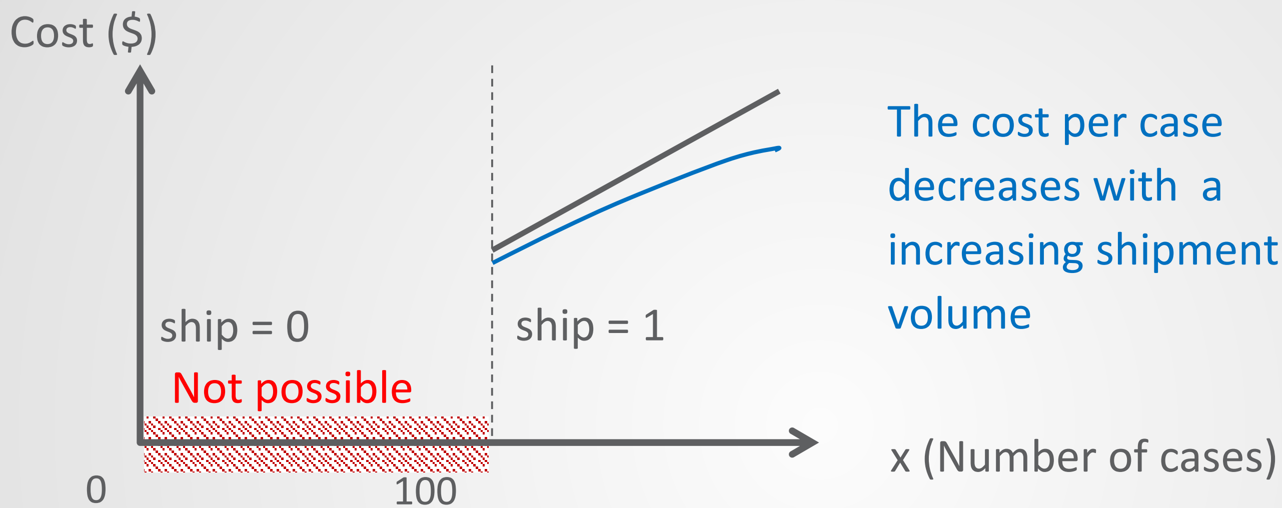
data.gdx | data.gms | results.gdx | vienna.gms | vienna.lst

```
* MIP

scalar minS minimum shipment /100/
       bigM big M;
bigm = min(smax(i,a(i)), smax(j,b(j)));
display bigm;

binary variable ship(i,j) '1 if we ship from i to j, 0 otherwise';
equations
       minship(i,j) minimum shipment
       maxship(i,j) maximum shipment;

  minship(i,j).. x(i,j) =g= mins*ship(i,j);
  maxship(i,j).. x(i,j) =l= bigM*ship(i,j);

Model modelMIP /modelLP, minship, maxship/ ;
option optcr =0;
Solve modelMIP using mip minimizing z;
rep1(i,j,'MIP') = x.l(i,j);
rep2('MIP')     = z.l;
```

+ MIP Model: Results



+ non-linear: Cost Savings



The cost per case decreases with a increasing shipment volume

Replace:

min $\sum_i \sum_j c_{ij} \cdot x_{ij}$ (Minimize total transportation cost)

With

min $\sum_i \sum_j c_{ij} \cdot x_{ij}^{\boldsymbol{beta}}$ (Minimize total transportation cost)

GAMS

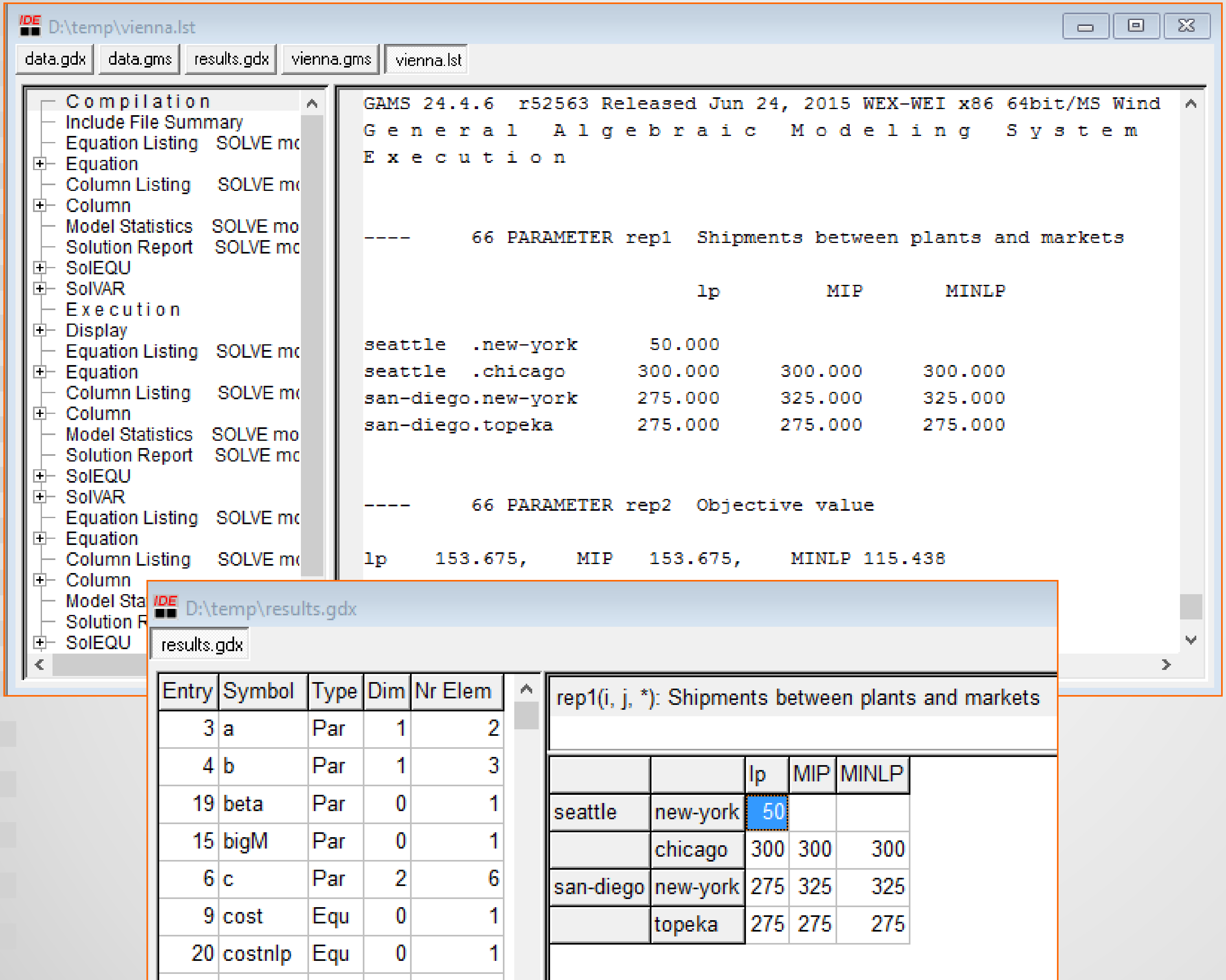
+ non-linear: GAMS Algebra

D:\temp\vienna.gms

data.gdx | data.gms | results.gdx | vienna.gms | vienna.lst

```
* MINLP

scalar beta /0.95/;
Equation costnlp        define non-linera objective function;
costnlp ..          z  =e=  sum((i,j), c(i,j)*x(i,j)**beta) ;

Model modelMINLP /modelMIP - cost + costnlp/ ;
Solve modelMINLP using minlp minimizing z ;
rep1(i,j,'MINLP') = x.l(i,j);
rep2('MINLP')     = z.l;
display rep1, rep2;
```

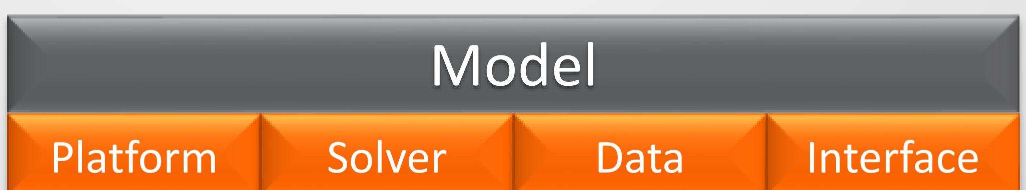
+ non-linear Results



Foundation of GAMS

Powerful algebraic modeling language

Open architecture and interfaces to other systems, independent layers



Independence of Model and Platform

Supported Platforms

Windows, Linux, Mac OS X, Solaris, AIX

➢ **Move models between platforms with ease!**

Model

Platform | Solver | Data | Interface

Independence of Model and Platform

Local and distributed / remote execution

- Distributed Algorithm (CPLEX, GUROBI)
- Grid Computing Facility
- NEOS (Kestrel)

GAMS/Kestrel

- Remote Solver Execution on NEOS Servers
- Stay in your GAMS environment
- Results are being processed as with any local solver

Local Machine ⟷ Remote Cluster (NEOS)

```
Model transport /all/;
Option lp=kestrel;
transport.optfile=1;

$onecho > kestrel.opt
kestrel_solver xpress
$offecho

Solve transport using lp minimizing z;
```

```
--- Executing KESTREL: elapsed 0:00:00.006
Connecting to: http://neos-server.org:3332
NEOS Solver: xpress
NEOS job#=3631352, pass=iJLdAkhP

Check the following URL for progress report :
http://neos-server.org/neos/cgi-bin/nph-neos-
solver.cgi?admin=results&jobnumber=3631352&pass=i
JLdAkhP
Job 3631352 dispatched
password: iJLdAkhP
---------- Begin Solver Output -----------
Job submitted to NEOS HTCondor pool.

FICO-Xpress      24.3.3 r48116 Released Sep 19,
2014 LEG x86 64bit/Linux
```

Independence of Model and Solver

➢ **GAMS is not a Solver!**

GAMS: Model building and interaction with solvers & environments

Solver: Solve an model instance (model + numerical data) using mathematical optimization

Model

Platform | Solver | Data | Interface

Independence of Model and Solver

Solver/Model type availability - 24.4												
	LP	MIP	NLP	MCP	MPEC	CNS	DNLP	MINLP	QCP	MIQCP	*Stoch.*	*Global*
ALPHAECP								✔		✔		
ANTIGONE 1.1			✔			✔	✔	✔	✔	✔		✔*
BARON 14.4	✔	✔	✔			✔	✔	✔	✔	✔		✔*
BDMLP	✔	✔										
BONMIN 1.7								✔		✔		
CBC 2.8	✔	✔										
CONOPT 3	✔		✔			✔	✔		✔			
COUENNE 0.4												
CPLEX 12.6												
DECIS												
DICOPT												
GLOMIQO 2.3												
GUROBI 6.0												
GUSS												
IPOPT 3.11												
KNITRO 9.1												
LGO												
LINDO 9.0												
LINDOGLOBAL 9.												
LOCALSOLVER 5												
MILES												
MINOS												
MOSEK 7												
MPSGE												
MSNLP												
NLPEC												
OQNLP												
PATH												
SBB												
SCIP 3.1												
SNOPT												
SOPLEX 2.0	✔											
SULUM 4.0	✔	✔										
XA	✔	✔										
XPRESS 27.01	✔	✔							✔	✔		

* deterministic global solver

- All major commercial and academic solvers integrated (30+)
- Wide range of available model types (12+)
- Average number of commercial solvers per license:
  - Academic clients: 2.9
  - Commercial clients: 2.2

GAMS

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

FICO · Gurobi Optimization · IBM · mosek · COIN|OR

➢ **More than 30 Solvers integrated!**

Model

Platform | Solver | Data | Interface

Independence of Model and Solver

Solver/Model type availability - 24.4	LP	MIP	NLP	MCP	MPEC	CNS	DNLP	MINLP	QCP	MIQCP	*Stoch.*	*Global*
ALPHAECP								✔		✔		
ANTIGONE 1.1			✔			✔	✔	✔	✔	✔		✔*
BARON 14.4	✔	✔	✔			✔	✔	✔	✔	✔		✔*
BDMLP	✔	✔										
BONMIN 1.7								✔		✔		
CBC 2.8	✔	✔										
CONOPT 3	✔		✔			✔	✔		✔			
COUENNE 0.4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
CPLEX 12.6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
DECIS	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
DICOPT	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
GLOMIQO 2.3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
GUROBI 6.0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
GUSS	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
IPOPT 3.11	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
KNITRO 9.1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
LGO	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
LINDO 9.0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
LINDOGLOBAL 9.	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
LOCALSOLVER 5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MILES	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MINOS	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MOSEK 7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MPSGE	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MSNLP			✔				✔		✔			✔
NLPEC				✔	✔							
OQNLP			✔				✔	✔	✔	✔		✔
PATH				✔		✔						
SBB								✔		✔		
SCIP 3.1		✔	✔			✔	✔	✔	✔	✔		✔*
SNOPT	✔		✔			✔	✔		✔			
SOPLEX 2.0	✔											
SULUM 4.0	✔	✔										
XA	✔	✔										
XPRESS 27.01	✔	✔							✔	✔		

* deterministic global solver

Uniform interface to all major solvers:

- Switching between solvers with one statement: `option solver= ….;`
- Unified Documentation
- Licensing (GAMS as a „license broker“)

Model

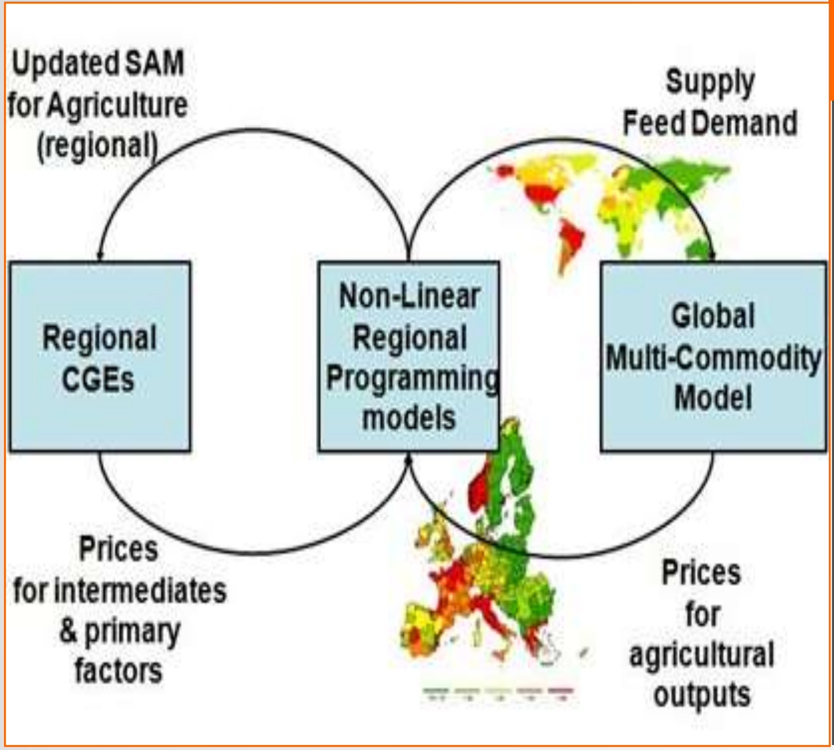
Platform | **Solver** | Data | Interface

Independence of Model and Solver

CyBio Scheduler

- Scheduling software for high throughput screening
- Used in the pharmaceutical industry (drug discovery)
- Model optimizes throughput of robotic screening systems
- Solver
  - COIN for smaller instances
  - Commercial MIP solver for large problems
- Developed by analytikjena

Independence of Model and Solver

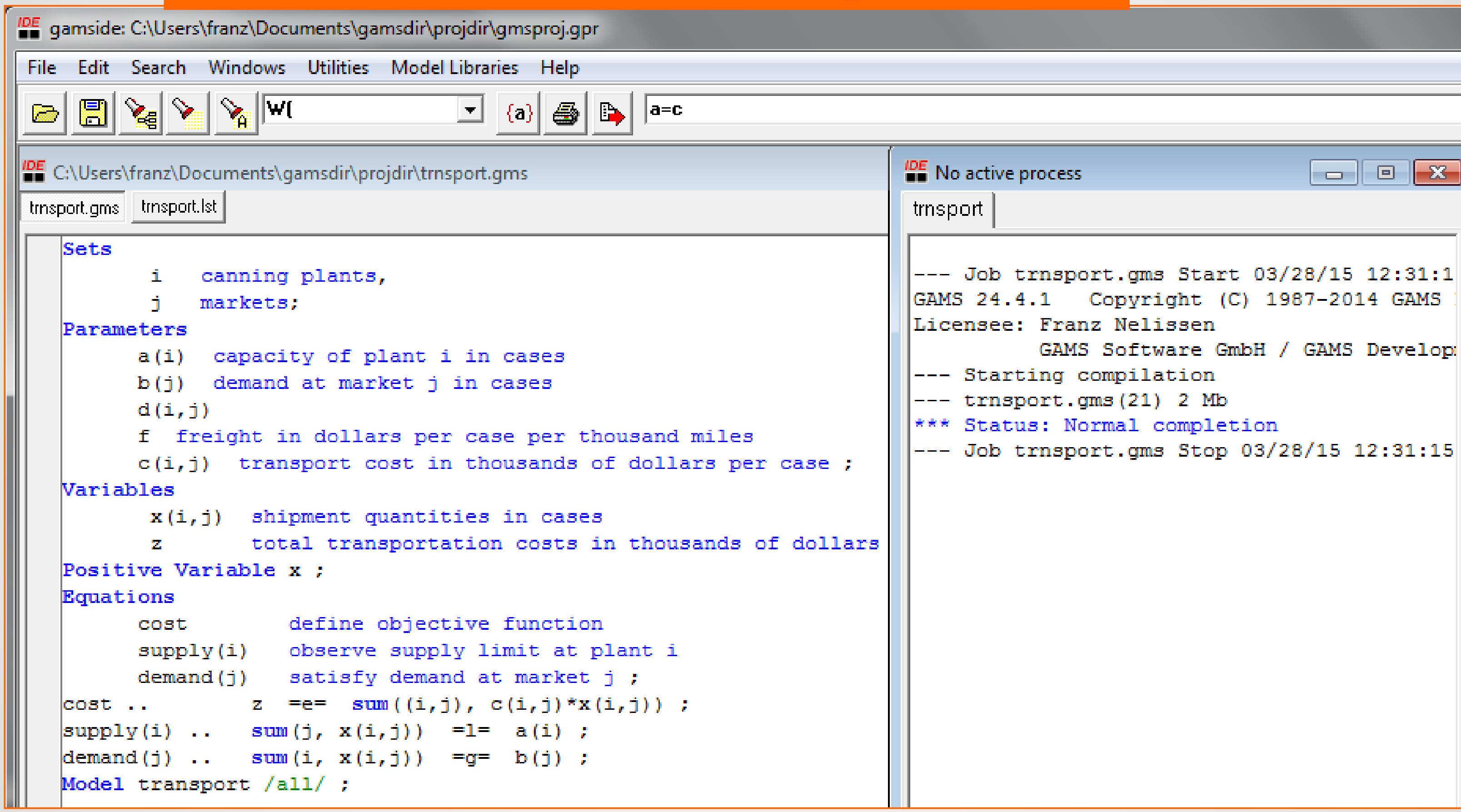


CAPRI

- Global agricultural sector models with focus on EU27 members
- Evaluates impacts of agricultural and trade policies on production income, markets, trade, and the environment
- Different NLP solvers
- Open source approach with an active network of developers and users coordinated by University of Bonn

Independence of Model and Data

➢ Declarative Modeling



Model

Platform | Solver | Data | Interface

Independence of Model and Data

Initial Model Development – ASCII

- Input: Part of model input ($include file.gms)
- Output: Display / Put Command (File)

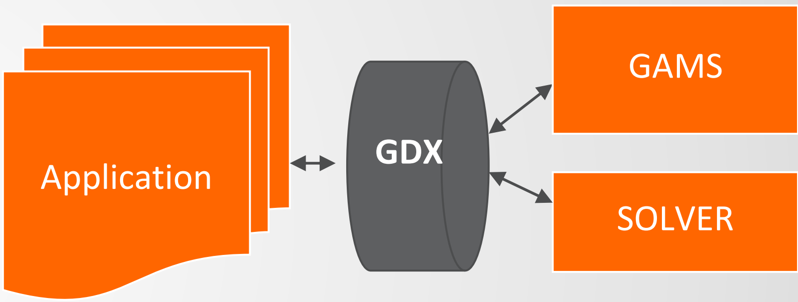
```
file fy /result.csv/;
fy.pc = 5; fy.nd = 4;
loop((i,j)$x.l(i,j),
     put fy i.te(i) j.te(j) x.l(i,j) /;
     );
```

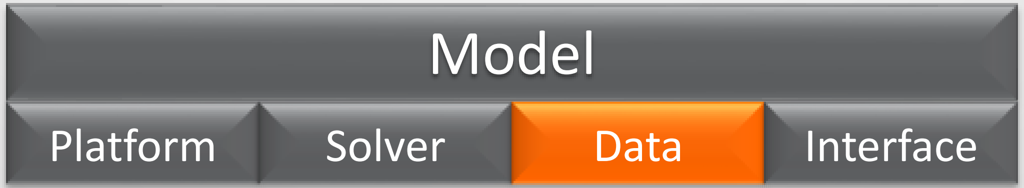
```
"seattle","new-york",50.00
"seattle","chicago",300.00
"san-diego","new-york",275.00
"san-diego","topeka",275.00
```

Independence of Model and Data

GDX: Binary Data layer ("contract") between GAMS and applications

- Fast exchange of data at any stage
- Platform independent
- Direct GDX interfaces and OO APIs
- No license required
- Scenario Management Support





Independence of Model and Data

➤ (GDX) Tools included in the GAMS distribution

- Release Notes
- Installation Notes
- User's Guides
- Solvers
- APIs
- Tools
  - GAMS IDE
  - Data Exchange
    - Excel
    - GDXMRW
    - GDXRRW
    - GDXDUMP
    - CSV2GDX
    - SQL2GMS
    - GDX2Access
    - GDX2SQLITE
    - MDB2GMS
    - MPS2GMS
    - GDXVIEWER
    - GDX2HAR
    - HAR2GDX
    - GDX2VEDA
    - GDXTROLL
  - GDX Service
    - GDXDIFF
    - GDXMERGE
    - GDXCOPY
    - GDXRENAME
  - Data Transformation
    - GDXRANK
    - SCENRED
    - SCENRED2
    - INVERT
    - CHOLESKY
    - EIGENVALUE
    - EIGENVECTOR
    - MCFILTER
    - CSDP
  - Other Tools

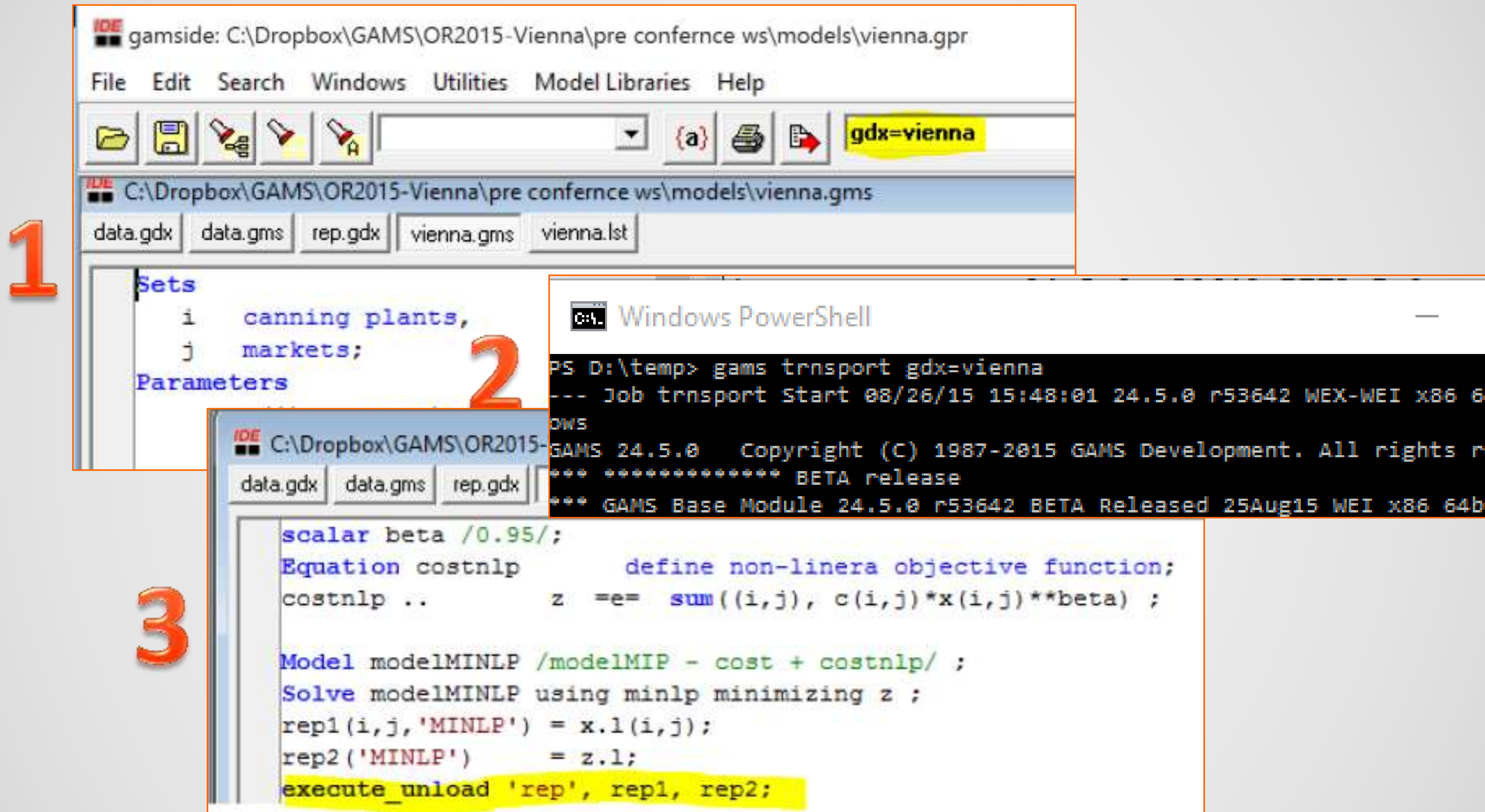
A large number of tools are included in GAMS distribution. Below we give a functional categorization of all tools as well as a brief description of each tool with the Supported Platforms.

- **GAMS IDE** - an integrated model development environment including a general text editor with the ability to launch and monitor the compilation and execution of GAMS models.
- **Data Exchange** - a collection of tools that provide functionality to exchange data between GAMS and other data sources. This category contains tools for popular data sources and high-level programming environment and like databases, Matlab, and R. There are also tools for specialized systems like HAR, TROLL, and VEDA. The tools to communicate with Microsoft Excel are grouped in a subcategory **Excel**. Many of the tools described here use the GAMS Data eXchange facility GDX.
- **GDX Service** - a collection of tools that operate directly on GDX containers to e.g. compare, copy, and merge the content of GDX..
- **Data Transformation** - a collection of tools that perform very specific tasks that are awkward or inefficient to implement in GAMS directly. Through GDX and the execution of the tools in this category allow to perform complex tasks from a GAMS model like identifying eigenvalues or the inverse of a matrix.
- **Other Tools** - a collection of more exotic tools that can become handy in some some special circumstances. Most notably, the collection contains the tool MODEL2TEX to document the model algebra in LaTeX format.

The following table gives an alphabetically sorted list of all available tools.

Tool	Description
ASK	The utility can be used to get input from an user interactively.
BIB2GMS	Analyses BibTeX files with file extension .bib and writes GAMS source files that can be used to create various author, reference and cross reference reports.
CHK4UPD	Checks whether the user can update to a more recent GAMS version.
CHOLESKY	Calculates the Choleksy decomposition of a symmetric positive definite matrix.
CSDP	The semidefinite programming CSDP solver from COIN-OR. The communication with CSDP requires the setup of matrix data structures in a CSDP input file. In a sense a GAMS model functions as a matrix

Creating a GDX file



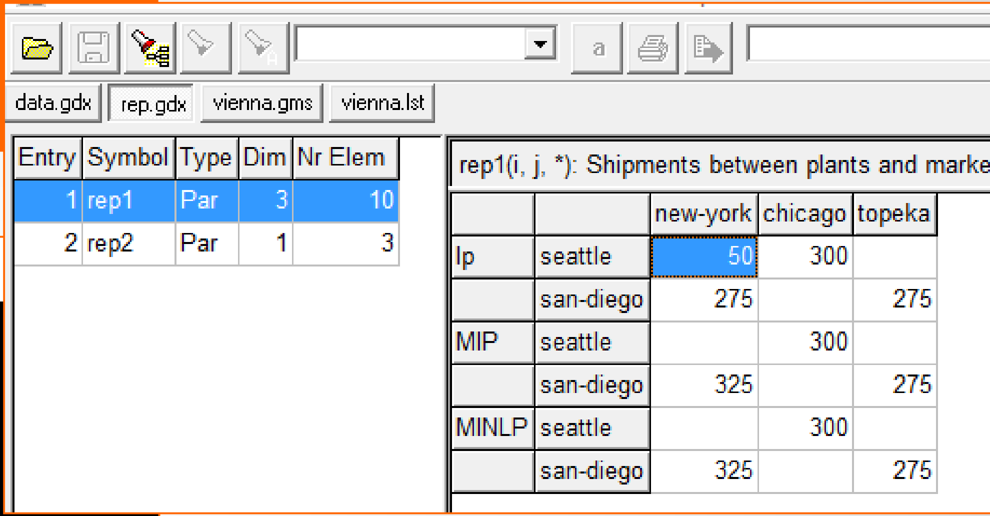
GAMS

Looking at a GDX file

1

GAMS IDE-GDX Browser

- Data cube with Drag&Drop
- Searching and Charting
- Export to Excel, HTML,…



2

Windows PowerShell

```
PS D:\temp> gdxdump .\rep.gdx
$onempty

Parameter rep1(*,*,*) Shipments between plants and markets /
'seattle'.'new-york'.'lp' 50,
'seattle'.'chicago'.'lp' 300,
'seattle'.'chicago'.'MIP' 300,
'seattle'.'chicago'.'MINLP' 300,
'san-diego'.'new-york'.'lp' 275,
'san-diego'.'new-york'.'MIP' 325,
'san-diego'.'new-york'.'MINLP' 325,
'san-diego'.'topeka'.'lp' 275,
'san-diego'.'topeka'.'MIP' 275,
'san-diego'.'topeka'.'MINLP' 275 /;

Parameter rep2(*) Objective value /
'lp' 153.675,
'MIP' 153.675,
'MINLP' 115.437925639658 /;

$offempty
PS D:\temp>
```

GDX2XLS

➢ Dumping a GDX container to Excel

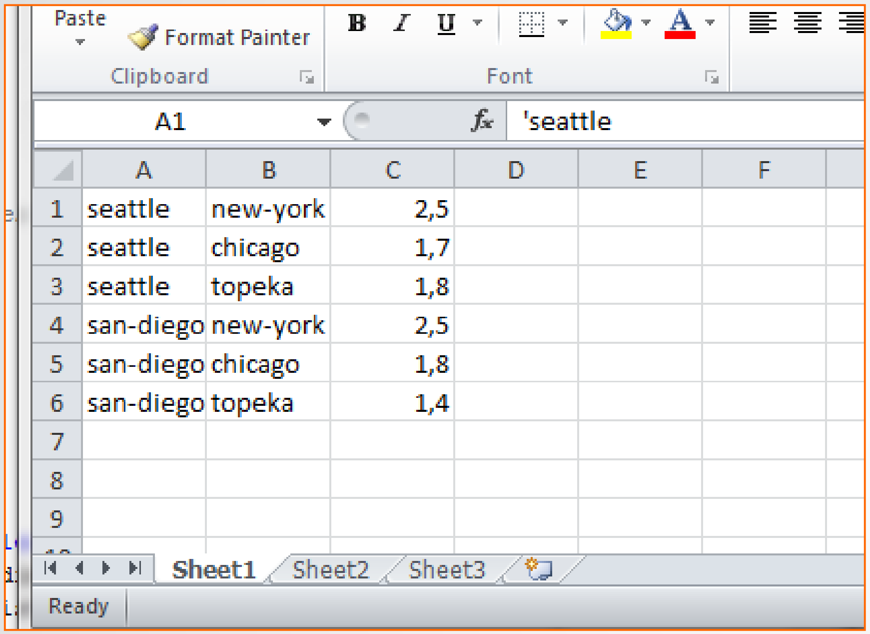
```
gdx2xls rep.gdx
shellexecute rep.xlsx
```

	A	B	C	D	E
1					
2	rep1	(parameter)	Shipments between plants and markets		
3	dim1	dim2	dim3	Value	
4	seattle	new-york	lp	50	
5	seattle	chicago	lp	300	
6	seattle	chicago	MIP	300	
7	seattle	chicago	MINLP	300	
8	san-diego	new-york	lp	275	
9	san-diego	new-york	MIP	325	
10	san-diego	new-york	MINLP	325	
11	san-diego	topeka	lp	275	
12	san-diego	topeka	MIP	275	
13	san-diego	topeka	MINLP	275	

rep1 | rep2

Ready 100%

GDXXRW – Accessing Excel Files



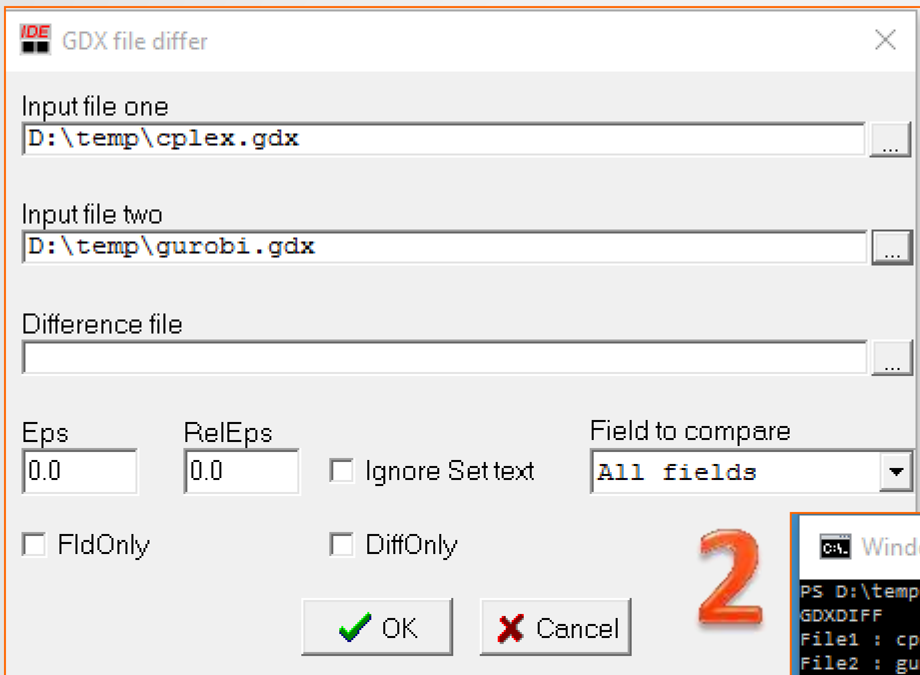
	A	B	C	D	E	F
1	seattle	new-york	2,5			
2	seattle	chicago	1,7			
3	seattle	topeka	1,8			
4	san-diego	new-york	2,5			
5	san-diego	chicago	1,8			
6	san-diego	topeka	1,4			
7						
8						
9						

```
$call GDXXRW dist.xlsx par=d rng=A1 rdim=2 cdim=0
$if errorlevel 1 $abort "Problems with Excel file!"
$gdxin dist
$load d
```

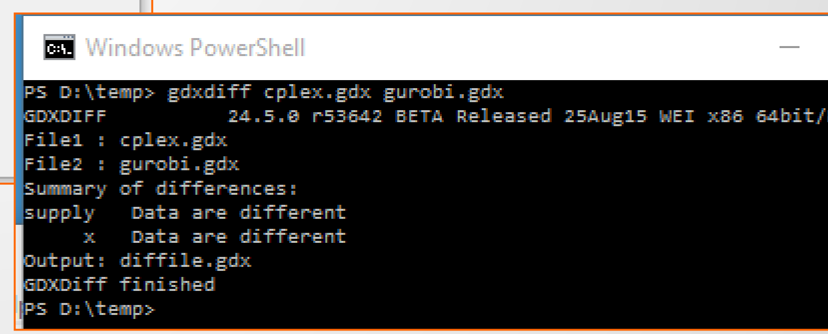
GDXDiff

➢ Comparing two gdx files

1



2



GAMS

GDXDiff - Results

1

diffile.gdx

Entry	Symbol	Type	Dim	Nr Elem
3	FilesCompared	Set	1	2
1	supply	Equ	2	4
2	x	Var	3	4

x(*, *, *): Differences

Level

		dif1	dif2
seattle	new-york	50	
san-diego	new-york	275	325

2

Windows PowerShell

```
$onempty

Equation supply(*,*) Differences /
'seattle'.'dif1'.L 350,
'seattle'.'dif1'.M Eps,
'seattle'.'dif1'.UP 350,
'seattle'.'dif2'.L 300,
'seattle'.'dif2'.UP 350,
'san-diego'.'dif1'.L 550,
'san-diego'.'dif1'.UP 600,
'san-diego'.'dif2'.L 600,
'san-diego'.'dif2'.UP 600 /;

Positive Variable x(*,*,*) Differences /
'seattle'.'new-york'.'dif1'.L 50,
'seattle'.'new-york'.'dif2'.M Eps,
'san-diego'.'new-york'.'dif1'.L 275,
'san-diego'.'new-york'.'dif2'.L 325 /;

Set FilesCompared(*) /
'File1' D:\temp\cplex.gdx,
'File2' D:\temp\gurobi.gdx /;

$offempty
PS D:\temp>
```

GAMS

Independence of Model and User Interface

1 Open architecture and interfaces to other systems
→ No preference for a particular user interface

2 (OO) Application Programming Interfaces

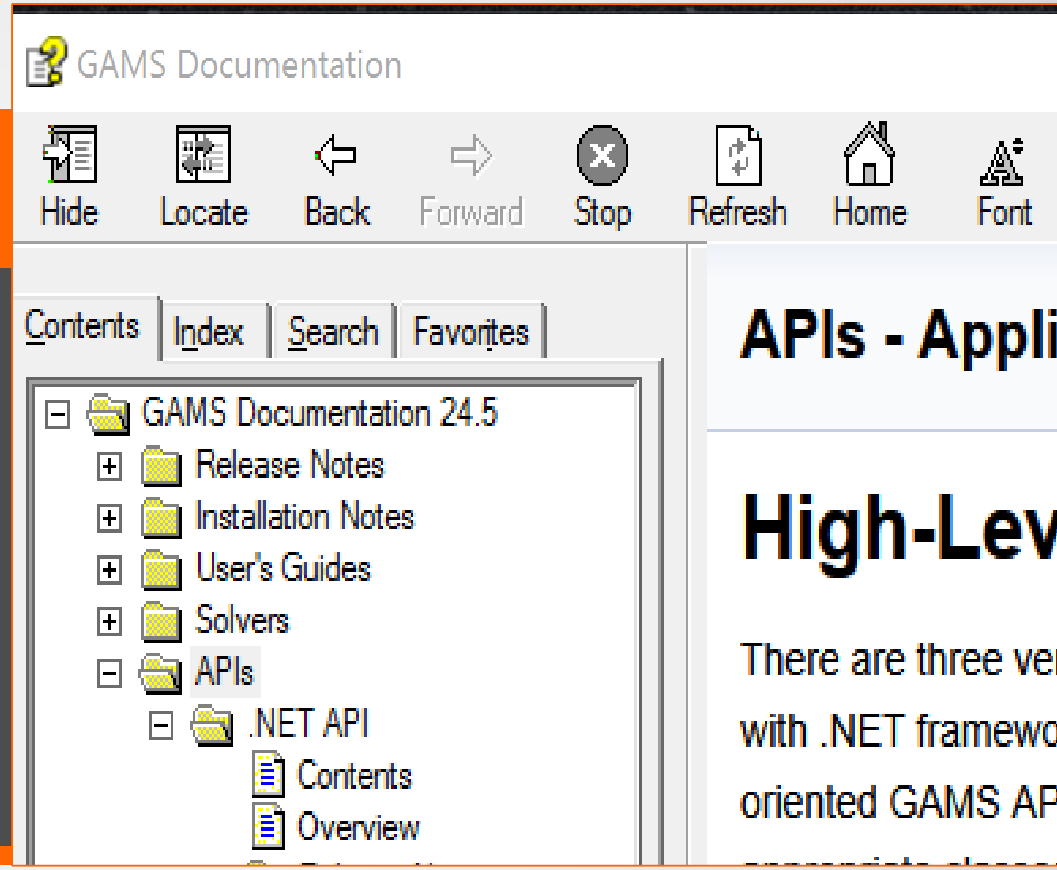
3 Smart Links to popular environments: Excel, MATLAB, R, ...

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



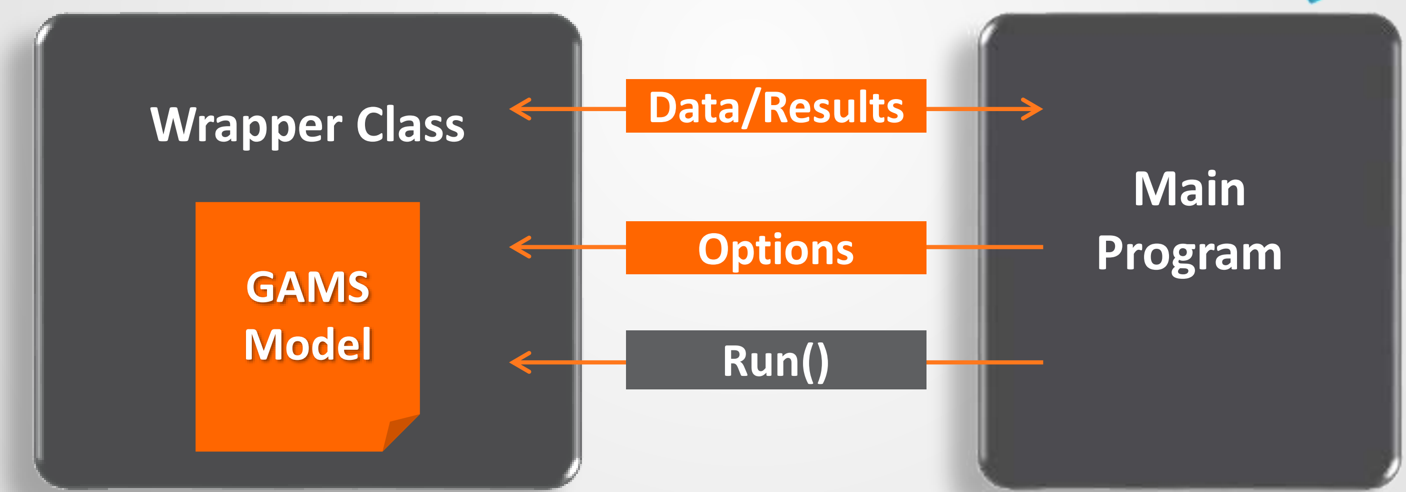
Model

Platform	Solver	Data	Interface

Simple Encapsulation of a GAMS Model

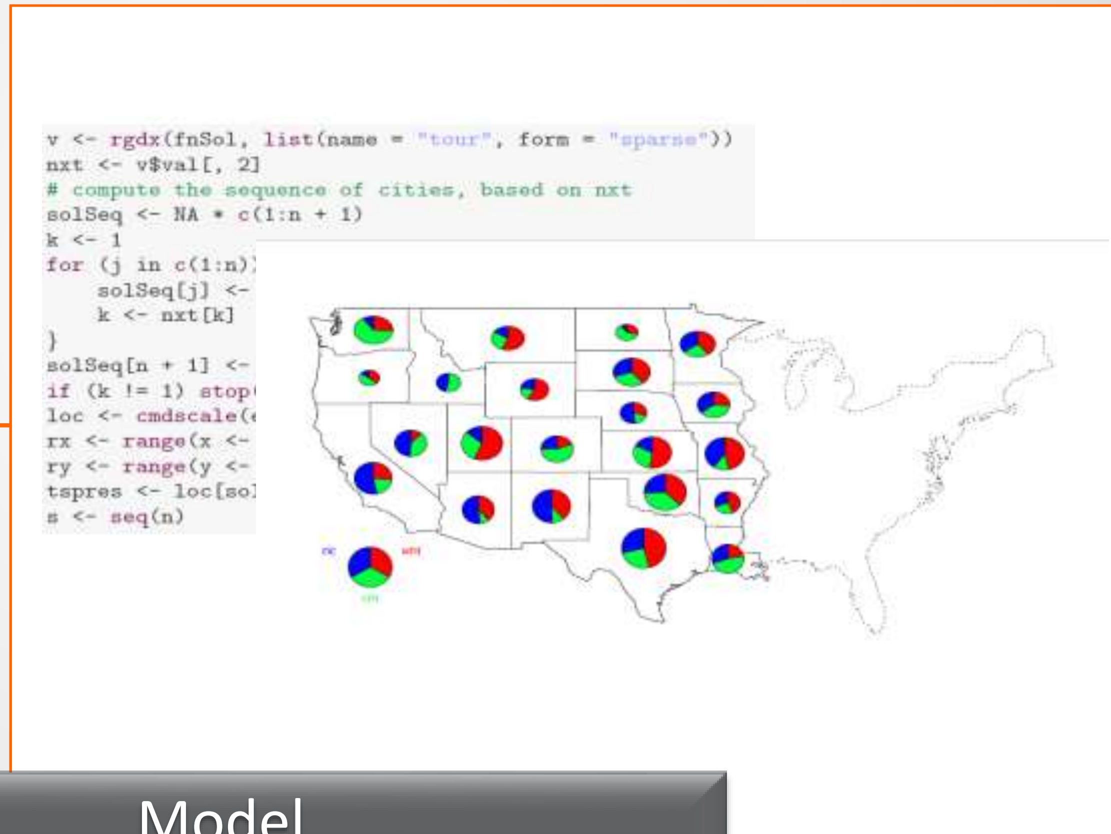
Simple API interface

- Properties to **communicate input data and results**
- Properties to **change options** like the solver to use
- Run() method to **run the model**



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



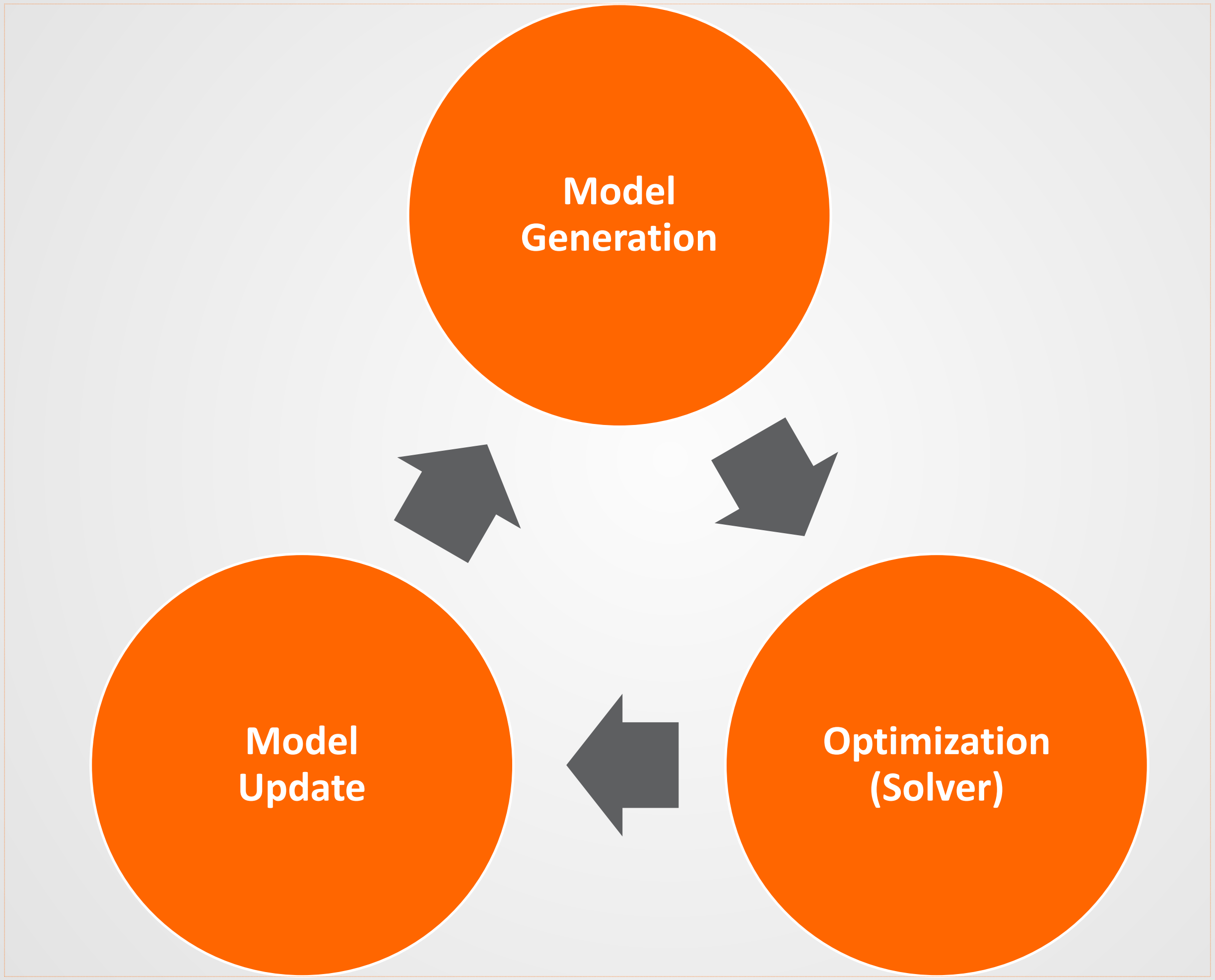
Agenda

GAMS – Elements and Examples

Scenarios in GAMS

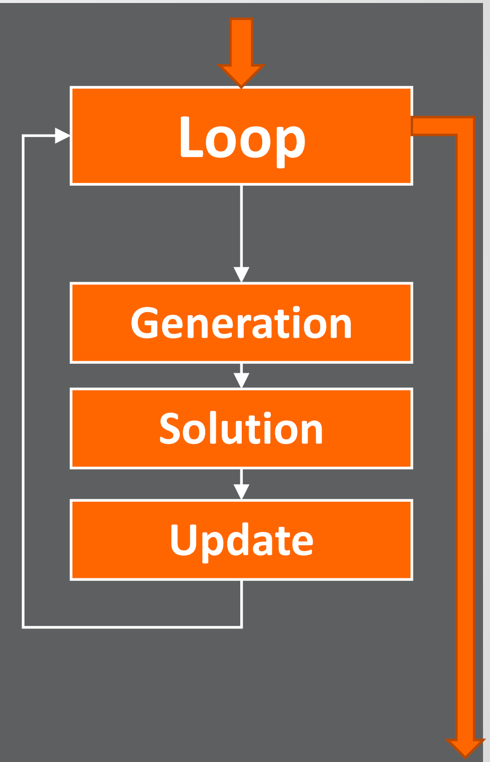
Some Enhancements

Solving Scenarios



Simple Serial Solve Loop

```
Set s /s*s10/;
Parameter
 A_s(s,i,j) “scenario data”,
 xlo_s(s,i,j) ”scenario lower bound 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

Simple Serial Solve - Performance

trnsport.gms (LP) solved 500 times with CPLEX:

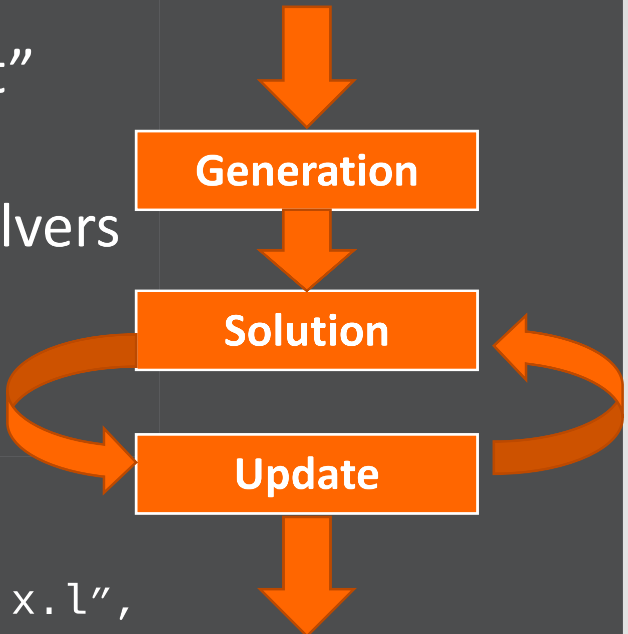
```
Loop(s,
    d(i,j) = dd(s,i,j);
    f = ff(s);
    solve transport using lp minimizing z;
    rep(s) = transport.objval;
);
```

Setting	Solve time (secs)
Solvelink=%Solvelink.ChainScript%	52.221
Solvelink=%Solvelink.CallModule%	37.366
Solvelink=%Solvelink.LoadLibrary%	03.252

Scenario Solver (GUSS)

- Generates model once and updates the algebraic model keeping the model “hot” inside the solver
- Platform independent, works with all solvers
- Performance close to native solver API
- Example:

```
Set s /s*s10/;
Parameter
  A_s(s,i,j) “scenario data”,
  xlo_s(s,i,j) ”scenario lower bound 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;
```



GAMS

Scenario Solver - Performance

Example: Stochastic model with 66,320 linear problems

Setting	Solve time (secs)
Loop: Solvelink=%Solvelink.Chainscript (default)	7,204
Loop: Solvelink=%Solvelink.LoadLibrary%	2,481
GAMS Scenario Solver	392
CPLEX Concert Technology	210

Factor

18.3

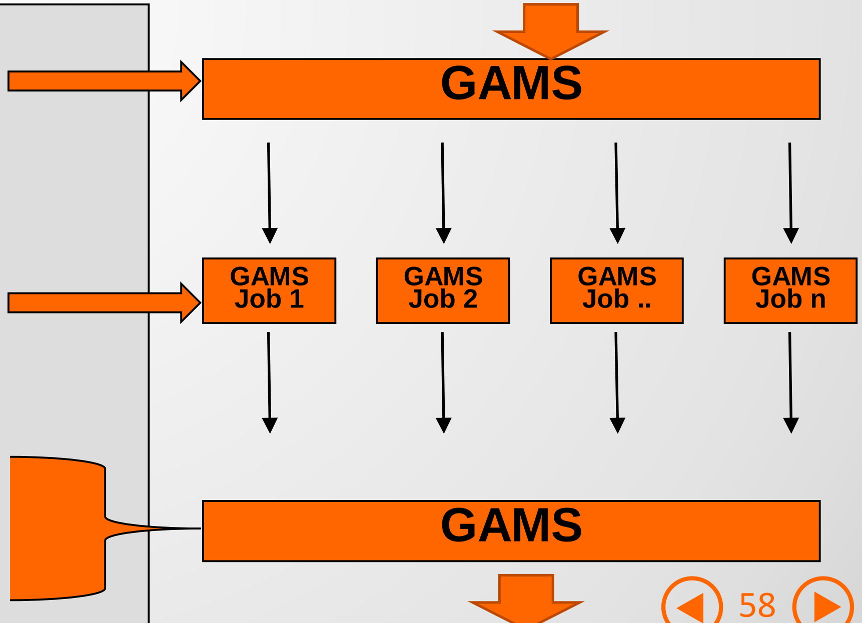
1.86

Grid Computing Facility

GAMS jobs in a **distributed** environment

- Scalable: supports large grids, but also works on local machine
- Platform independent, works with all solvers/model types
- Only minor changes to model required

1. Submission of jobs
2. "Grid Middleware"
   - Distribution of jobs
   - Job execution
3. Collection of solutions
4. Processing of results



GAMS

Grid Computing Facility - Example

➢ GAMS Model Library: Transgrid

D:\temp\trnsgrid.gms

trnsgrid.gms | trnsgrid.lst | vienna.gms

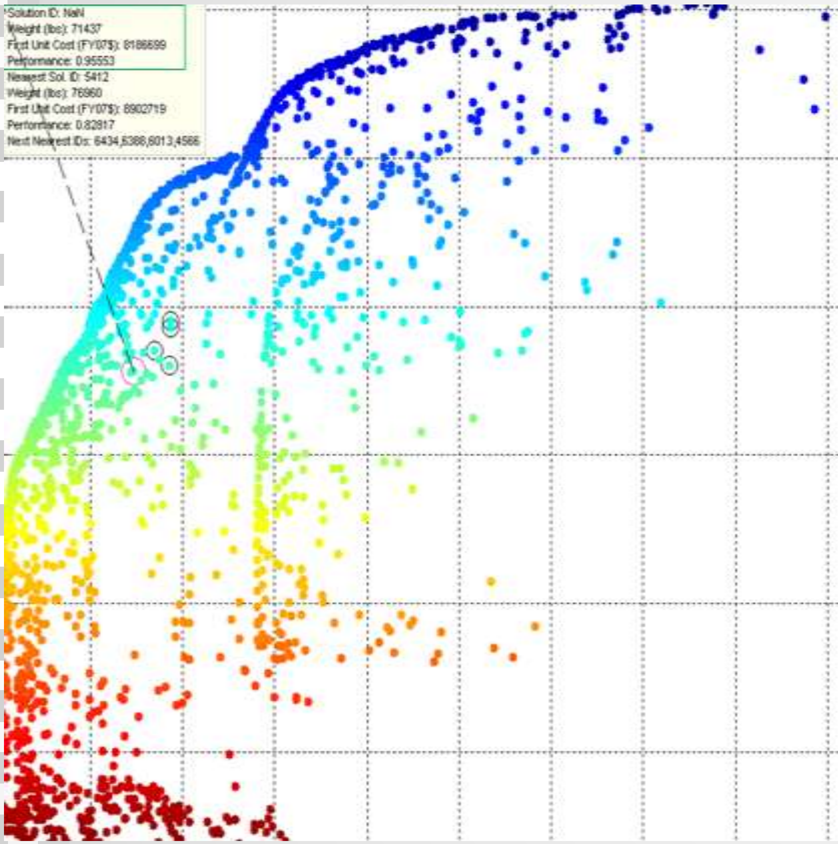
```
transport.solvelink = %solvelink.AsyncGrid%;  // turn on grid optio
transport.limcol    = 0;
transport.limrow    = 0;
transport.solprint  = %solprint.Quiet%;

set s scenarios / 1*5 /;

parameter dem(s,j)  random demand
          h(s)  store the instance handle;

dem(s,j)= b(j)*uniform(.95,1.15); // create some random demands

loop(s,
   b(j) = dem(s,j)
   Solve transport using lp minimizing z;
   h(s) = transport.handle );     // save instance handle

parameter repx(s,i,j) solution report
          repy          summary report;

repy(s,'solvestat') = na;
repy(s,'modelstat') = na;

* we use the handle parameter to indicate that the solution has bee
repeat
   loop(s$handlecollect(h(s)),
      repx(s,i,j) = x.l(i,j);
      repy(s,'solvestat') = transport.solvestat;
      repy(s,'modelstat') = transport.modelstat;
      repy(s,'resusd'   ) = transport.resusd;
      repy(s,'objval')    = transport.objval;
      display$handledelete(h(s)) 'trouble deleting handles' ;
      h(s) = 0 ) ;     // indicate that we have loaded the solution
   display$sleep(card(h)*0.2) 'was sleeping for some time';
until card(h) = 0 or timeelapsed > 10;  // wait until all models ar

display repx, repy;

abort$sum(s$(repy(s,'solvestat')=na),1) 'Some jobs did not return';
```

```
** GAMS Base Module 24.5.0 r53642 BETA Released 25Aug15 WEI x86 64bit/MS
** ************* BETA release
icensee: Franz Nelissen                                  S150326/0001CN-GE
         GAMS Software GmbH                                          DC345
-- Starting compilation
-- trnsgrid.gms(101) 3 Mb
-- Starting execution: elapsed 0:00:00.003
-- trnsgrid.gms(75) 4 Mb
-- Generating LP model transport
-- trnsgrid.gms(77) 4 Mb
-- LOOPS s = 1
--    6 rows  7 columns  19 non-zeroes
-- Submitting model transport with handle grid145000001
-- Executing after solve: elapsed 0:00:00.017
-- trnsgrid.gms(75) 4 Mb
-- Generating LP model transport
-- trnsgrid.gms(77) 4 Mb
-- LOOPS s = 2
--    6 rows  7 columns  19 non-zeroes
-- Submitting model transport with handle grid145000002
-- Executing after solve: elapsed 0:00:00.094
-- trnsgrid.gms(75) 4 Mb
-- Generating LP model transport
-- trnsgrid.gms(77) 4 Mb
-- LOOPS s = 3
--    6 rows  7 columns  19 non-zeroes
-- Submitting model transport with handle grid145000003
-- Executing after solve: elapsed 0:00:00.121
-- trnsgrid.gms(75) 4 Mb
-- Generating LP model transport
-- trnsgrid.gms(77) 4 Mb
-- LOOPS s = 4
--    6 rows  7 columns  19 non-zeroes
-- Submitting model transport with handle grid145000004
-- Executing after solve: elapsed 0:00:00.134
-- trnsgrid.gms(75) 4 Mb
-- Generating LP model transport
-- trnsgrid.gms(77) 4 Mb
-- LOOPS s = 5
--    6 rows  7 columns  19 non-zeroes
-- Submitting model transport with handle grid145000005
-- Executing after solve: elapsed 0:00:00.147
-- trnsgrid.gms(88) 4 Mb
-- GDXin=D:\temp\225a\grid145000001\gmsgrid.gdx
-- Removed handle grid145000001
-- GDXin=D:\temp\225a\grid145000002\gmsgrid.gdx
-- Removed handle grid145000002
-- GDXin=D:\temp\225a\grid145000003\gmsgrid.gdx
-- Removed handle grid145000003
```

Solving "many" Scenarios

1. Small Ratio of solver time / GAMS time → Scenario Solver
2. Large ratio i.e. only solver time is relevant (pre/post processing not critical) → Grid Computing Facility
3. Entire model run including pre processing / optimization / post processing is costly → Parallel execution of entire model in the cloud

Application - Scenario Solver



- Generation of Efficient Frontiers
- GAMS/CPLEX solution pool to collect 1,000,000+ solutions of a small MIP
- Compact representation of solutions
- Uses an ACE algorithm to filter out dominated solutions

Application - Scenario Solver

➢ Scenario Solver and Parallel Combined

Implementation	Number of MIP models	Solve time	Rest of algorithm	Total time
Traditional GAMS loop	100,000	1068 sec	169 sec	1237 sec
Scenario Solver	100,000	293 sec	166 sec	459 sec

Implementation	Number of MIP models	Worker Threads	Parallel sub-problem time	Rest of algorithm (serial)	Total time
Parallel + Scenario Solver	100,000	4	116 sec	67 sec	183 sec

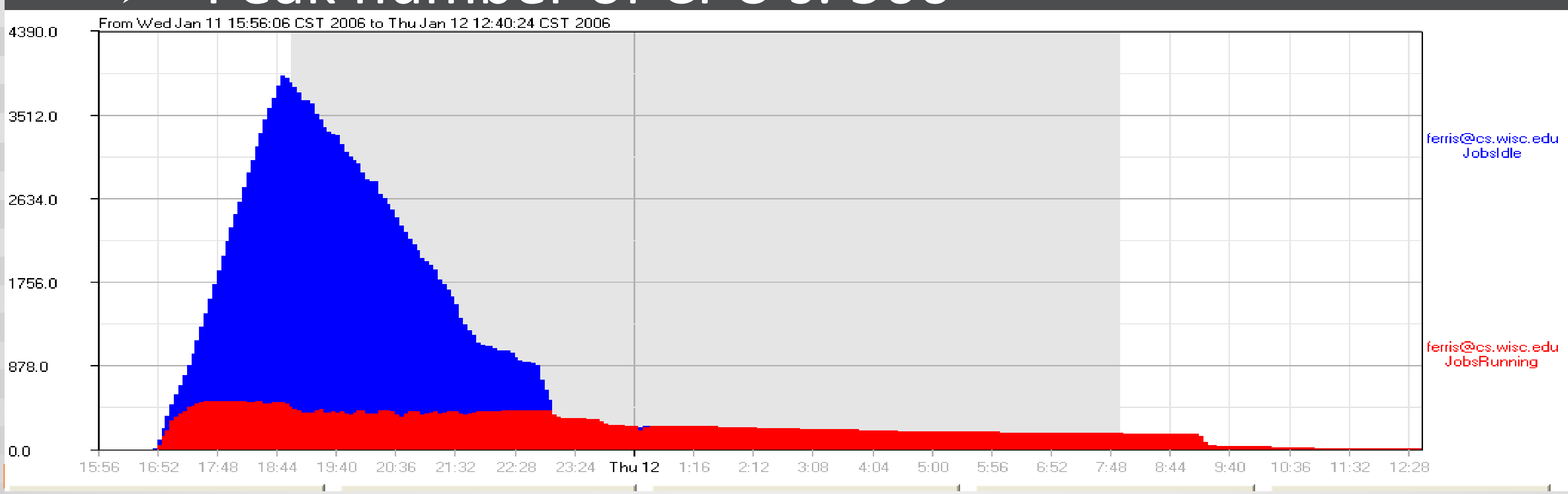
http://yetanothermathprogrammingconsultant.blogspot.de/2012/04/parallel-gams-jobs-2.html

GAMS

Application - Grid Computing Facility

- 4096 MIP models on HT Condor
- Submission started Jan 11,16:00
- All jobs submitted by Jan 11, 23:00
- All jobs returned by Jan 12, 12:40
  - 20 hours wall time, 5,000 CPU hours
  - Peak number of CPU's: 500

HTCondor
High Throughput Computing



GAMS

Application – Cloud Computing

xyz – Energy Company

- Task: Solve 1,000 scenarios (MIPs, ~1 hour per scenario) once a week overnight in parallel
- Implementation:
  - Amazon Cloud: 1000 parallel machines (instances), Python, GAMS + OO API
  - Issues: Automation / Licensing / Security
  - Hardware Costs / run: **$70!**
    (1,000 instances/run * $0.07 instance / hour)

Agenda

GAMS – Elements and Example

Scenarios in GAMS

Some Enhancements

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 and Solution Concepts**

Protect investments of Users

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

New Modeling and Solution Concepts

Examples:

- Bilevel Programs
- Extended Nonlinear Programs
- Stochastic Programming
- Disjunctive Programs

Issues:

- Breakouts of traditional Mathematical Programming classes
- Limited support with common model representation
- Incomplete/experimental solution approaches
- New and interesting solver features driven by implementation choices → May break solver independence of models

Challenge:

- Find a concept that combines the essentials of new features independent of the particular implementation choices.

GAMS

The "GAMS" – Approach

Extended Mathematical Programming

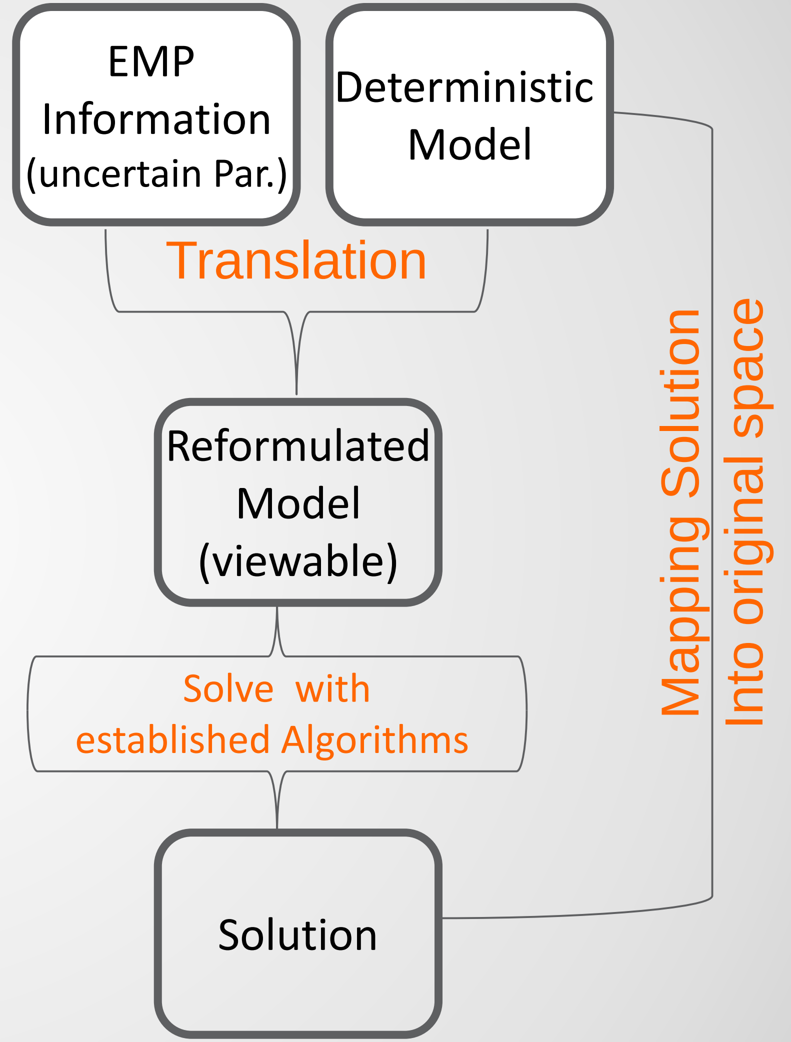
- **Experimental framework** for **automated** mathematical programming **reformulations**
- Keep the language simple: Do **not overload** existing GAMS notation
- Use **existing language features** to specify additional model features, structure, and semantics
- Express **extended model information** in **symbolic (source) form** and **apply existing modeling/solution technology**
- Package new tools with the production system

➢ **GAMS is conservative when it comes to syntax extensions**

Stochastic Programming in GAMS

EMP/SP

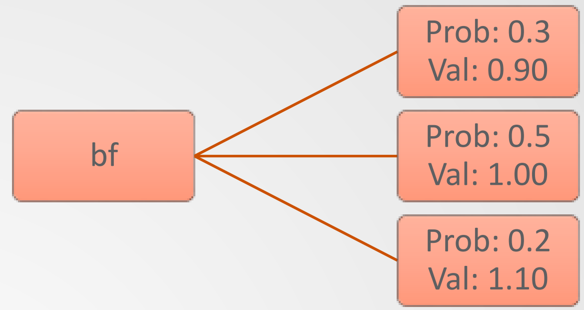
- Simple interface to add uncertainty to existing deterministic models
- (EMP) Keywords to describe uncertainty include: discrete and parametric random variables, stages, chance constraints, Value at Risk, ...
- Available solution methods:
  - Automatic generation of **D**eterministic **E**quivalent (can be solved with any solver)
  - Specialized commercial algorithms (DECIS, LINDO)



Transport Example - Uncertain Demand

b(j): demand at market j in cases	
new-york	325
chicago	300
topeka	275

Uncertain demand factor *bf*



Decisions to make

- First-stage decision: How many units should be shipped “here and now” (without knowing the outcome)
- Second-stage (recourse) decision:
  - How can the model react if we do not ship enough?
  - Penalties for “bad” first-stage decisions, e.g. buy additional cases `u(j)` at the demand location:

```
costsp ..  z  =e=  sum((i,j), c(i,j)*x(i,j))+
sum(j,0.3*u(j));
demandsp(j) .. sum(i, x(i,j))  =g=  bf*b(j) - u(j) ;
```

Uncertain Demand - GAMS Algebra

data.gdx | data.gms | rep.gdx | vienna.gms | vienna.lst

```
* Stochastic

Positive Variable u(j) unmet demand;
scalar bf demand factor /1/;
equation costsp, demandsp;
costsp ..         z  =e=  sum((i,j), c(i,j)*x(i,j)) + sum(j,0.3*u(j));
demandsp(j) ..   sum(i, x(i,j))  =g=  bf*b(j) - u(j) ;

Model modelsp /costsp, demandsp,  supply/ ;
file emp / '%emp.info%' /; put emp '* problem %gams.i%'/;
$onput
randvar bf discrete 0.3 0.9
                    0.5 1.0
                    0.2 1.1
stage 2 bf demandsp u
$offput
putclose emp;

Set scen        scenarios / s1*s3 /;
Parameter
    s_bf(scen)     demand factor realization by scenario
    s_x(scen,i,j)  shipment per scenario
    s_u(scen,j)    cases bought per scenario;

Set dict / scen .scenario.''
           bf   .randvar .s_bf
           x    .level   .s_x
           u    .level   .s_u /;

option solver=lindo;
Solve modelsp using emp minimizing z scenario dict;
Display s_bf, s_x, s_u;
```

GAMS

Uncertain Demand - Results

vienna.lst | rep.gdx

```
----    103 PARAMETER s_bf  demand factor realization by scenario

s1 0.900,    s2 1.000,    s3 1.100


----    103 PARAMETER s_x  shipment per scenario

                new-york     chicago      topeka

s1.seattle        50.000     300.000
s1.san-diego     242.500                 275.000
s2.seattle        50.000     300.000
s2.san-diego     242.500                 275.000
s3.seattle        50.000     300.000
s3.san-diego     242.500                 275.000


----    103 PARAMETER s_u  cases bought per scenario

      new-york     chicago      topeka

s2      32.500
s3      65.000      30.000      27.500
```

Stochastic Programming in GAMS

- Start with a deterministic model and define uncertain model parameters
- EMP/SP replaces these parameters and generates stochastic model
- EMP/SP supports:
  - Discrete and continuous distributions
  - Multi-stage problems
  - Chance constraints
  - Various Risk Measures (EV, VaR, CVaR)
- EMP/SP can use specialized algorithms (DECIS, LINDO) or create the Deterministic Equivalent (free solver DE)
- Further Information: http://www.gams.com/dd/docs/solvers/empsp/index.html

Logical Constraints

- **Example:** `If y=1 => x1 + x2 + x3=L= 0;`

- EMP supports **automatic reformulation** of logical constraints (disjunctive programming) in various formats, e.g.
  - BigM-Formulation
    ```
    constr01.. x1 + x2 + x3 =L= M*(1-y);
    y ∊ {0,1};
    ```
  - Indicator Constraints (CPLEX, XPRESS, SCIP):
    ```
    constr01.. x1 + x2 + x3 =L= 0;
    indic constr01$y 1
    ```
  - *Alternative: SOS1 (GUROBI):*
    ```
    constr01.. x1 + x2 + x3 =L= M;
    SOS1:(y, M);
    ```
- Further Information: http://www.gams.com/dd/docs/solvers/jams/index.html

GAMS

Quality Assurance at GAMS

Solver/Platform availability - 24.4

	x86 32bit MS Windows	x86 64bit MS Windows	x86 64bit Linux	x86 64bit MacOS X	x86 64bit SOLARIS	Sparc 64bit SOLARIS	IBM Power 64bit AIX
ALPHAECP	✔	✔	✔	✔	✔	✔	✔
ANTIGONE 1.1	✔	✔	✔	✔			
BARON 14.4	✔	✔	✔				
BDMLP	✔	✔	✔	✔	✔	✔	✔
BONMIN 1.7	✔	✔	✔	✔	✔		
CBC 2.8	✔	✔	✔	✔	✔		
COUENNE 0.4	✔	✔	✔	✔	✔		
CONOPT 3	✔	✔	✔	✔	✔	✔	✔
CPLEX 12.6	✔	✔	✔	✔	✔	✔	✔
DECIS	✔	✔	✔			✔	
DICOPT	✔	✔	✔	✔	✔	✔	✔
GLOMIQO 2.3	✔	✔	✔	✔			
GUROBI 6.0	✔	✔	✔	✔			✔
					✔		
					8.0		
					8.0		
					✔	✔	
MILES	✔	✔	✔	✔	✔	✔	✔
MINOS	✔	✔	✔	✔	✔	✔	✔
SULUM 4.0	✔	✔	✔				
XA	✔	✔	✔				
XPRESS 27.01	✔	✔	✔	✔	✔	✔	✔

- 7 supported Platforms
- 30+ Integrated Solvers

- What are the impacts of
  - new features
  - updated or new solvers?
- Is the new distribution backward compatible?

Quality Assurance - Motivation

Quality Assurance

- Essential component in most industries
- Important in most software engineering sectors

Mathematical Programming

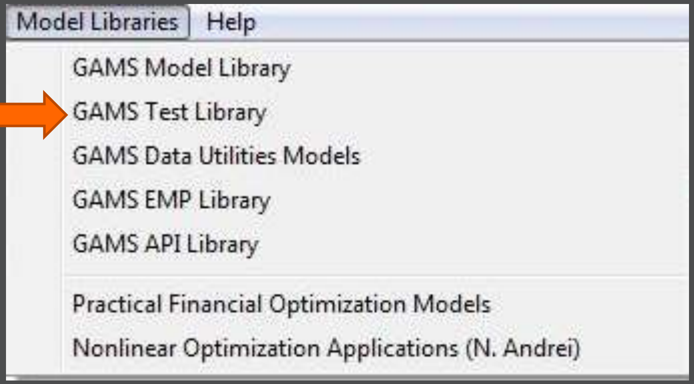
- Less attention to quality assurance (small community)
- Specific QA issues for modeling systems (initially expensive)
- Different focus for industry and academia

Industry ⟷ **Academia**

Focus on reliability — Focus on performance

Quality Test Model Library

- Tests to verify proper behavior of the system
- More than 680 quality test models, each containing numerous pass/fail tests
- Assurance about the basic functionality of the software!
- Give the tools in the hand of the user: Included in any distribution!



- Automatically executed every night for all solver combinations:
  - 13,000+ runs / platform (all tests)
  - Test summaries with different level of information

GAMS

Nightly Quality Tests: Results

Latest GAMS System Builds and Test Results

Sunday 12Apr15 15:22 (UTC)

[Latest Builds | Alpha Builds | Beta Builds | Nightly Builds | α Help System | Glossary]

Comments?

NOTE: The (nightly) alpha builds are internal development versions of the GAMS system. They may have known bugs, unfinished features, beta versions of third-party software, or may not function at all! Not for production use!

nightly α	System	Libraries	Build	Rev	Status and Time (UTC)		Initial Tests		Full Tests	
Saturday	aix	Download	24.5.0	51758	Test done	11Apr2015 09:29:07	385 runs 0 failures (q=0,s=0)	Report	3569 runs 1 failures (q=0,s=1)	Report
Saturday	deg	Download	24.5.0	51758	Test started	11Apr2015 01:47:52	619 runs 1 failures (q=1,s=0)	Report	results pending	
Saturday	leg	Download	24.5.0	51758	Test started	11Apr2015 02:01:21	647 runs 0 failures (q=0,s=0)	Report	results pending	
Saturday	legONdrcooper	Download	24.5.0	51758	Test done	11Apr2015 09:05:50	629 runs 0 failures (q=0,s=0)	Report	10832 runs 0 failures (q=0,s=0)	Report
Saturday	sig	Download	24.5.0	51758S	Test done	12Apr2015 04:24:15	493 runs 0 failures (q=0,s=0)	Report	7110 runs 0 failures (q=0,s=0)	Report
Saturday	vs8	Download	24.5.0	51758	Test started	11Apr2015 02:45:42	654 runs 0 failures (q=0,s=0)	Report	results pending	
Saturday	wei	Download	24.5.0	51758	Test started	11Apr2015 05:06:52	652 runs 0 failures (q=0,s=0)	Report	results pending	

nightly β	System	Libraries	Build	Rev	Status and Time (UTC)		Initial Tests		Full Tests	
Thursday	aix	Download	24.4.4	51738	Test done	09Apr2015 17:39:11	394 runs 2 failures (q=0,s=2)	Report	3615 runs 3 failures (q=0,s=3)	Report
Monday	deg	Download	24.4.4	51709	Test done	08Apr2015 19:01:03	602 runs 0 failures (q=0,s=0)	Report	13030 runs 0 failures (q=0,s=0)	Report

GAMS

Nightly Quality Tests: Reports

GAMS System Builds and Test Results Archive

Sunday 12Apr15 15:26 (UTC)

[Latest Builds | Alpha Builds | Beta Builds | Nightly Builds | Glossary]

```
Total:          3569 runs          1 failures
Quality:        1117 runs          0 failures
Slvtest:        2339 runs          1 failures
EMP:              99 runs          0 failures
Data:             14 runs          0 failures

**** SLVTEST FAILURES (failures_slv.gms)
$call  =gams slvtest lo=2 --prefix=aix --fail=failures_slvI.gms --test=bchtlbas u1="ord(s)=286" --runall=no --c1=sbb --ftr

**** SLVTEST FAILURES DETAIL (slvtest.sum?)
 25020  04/11/15 05:45:16 bchtlbas        MINLP    SBB         Bad ModelSolveStat[13,13] minlp=sbb
```

GAMS/CONVERT

- Transforms a GAMS model instance into a scalar model into different formats: Models can then be passed on to others for investigation without confidentiality being lost
- A way of sharing GAMS test problems for use with other modeling systems or solvers
- More than 25 target formats

Option	Description	
All	Generates all supported file formats	
AlphaECP	Generates AlphaECP input file	
Ampl	Generates Ampl input file	
AmplNLC	Generate Ampl NLC compatible file	
Analyze	Generates three text files for rows columns and matrix	
AnalyzeS	Generates short form of Analyze	
Baron	Generates Baron input file	
CplexLP	Generate CPLEX LP format input file	
CplexMPS	Generate CPLEX MPS format input file	
Dict	Generate Convert to GAMS Dictionary	
DictMap	Generate Convert to GAMS Dictionary Map	
FileList	Generate file list of file formats generated	
FixedMPS	Generate fixed format MPS file	fixed.mps
Gams	Generate GAMS scalar model. This is the default conversion format used.	gams.gms
Lgo	Generate an LGO Fortran file	lgomain.for
LindoMPI	Generate Lindo MPI file	lindo.mpi
Lingo	Generate Lingo input file	lingo.lng
LocalSolver	Generate LocalSolver input file (only with ConvertD)	localsolver.lsp
LSPSol	Generate Output function in LocalSolver input file (only with ConvertD)	lspsol.gms
Memo	Generate a memo file containing model statistics and files created.	memo.txt
Minopt	Generate Minopt input file	minopt.dat
NLP2dual	Generate the Wolfe dual of a smooth optimization model	gamsdual.gms
NLP2MCP	Generates GAMS scalar MCP model	gamsmcp.gms
OSiL	Generates Optimization Services instance Language (OSil) file	osil.xml
Pyomo	Generates Pyomo Concrete scalar model	gams.py
SFS	Generates Solver Foundation Services OML file	sfs.oml
ViennaDag	Generate Vienna Dag input file	vienna.dag

GAMS/CONVERT - Example

GAMS Code → **Scalar Model**

GAMS Code

```
Variables
     x(i,j)  shipment quantities in cases
     z       total transportation costs in thousands
Positive Variable x ;
Equations
     cost        define objective function
     supply(i)   observe supply limit at plant i
     demand(j)   satisfy demand at market j ;
cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
supply(i) ..     sum(j, x(i,j))  =l=  a(i) ;
demand(j) ..     sum(i, x(i,j))  =g=  b(j) ;
Model transport /all/ ;
Solve transport using lp minimizing z ;
```

Scalar Model

```
Variables  x1,x2,x3,x4,x5,x6,x7;
Positive Variables  x1,x2,x3,x4,x5,x6;
Equations  e1,e2,e3,e4,e5,e6;
e1..  - 0.225*x1 - 0.153*x2 - 0.162*x3 -
0.225*x4 - 0.162*x5 - 0.126*x6 + x7=E= 0;
e2..    x1 + x2 + x3 =L= 350;
e3..    x4 + x5 + x6 =L= 600;
e4..    x1 + x4 =G= 325;
e5..    x2 + x5 =G= 300;
e6..    x3 + x6 =G= 275;
Model m / all /;
m.limrow=0; m.limcol=0;
Solve m using LP minimizing x7;
```

Protecting IP and Sensitive Data

Obfuscate or hide sensitive information

- Extrinsic function libraries
- External Equations
- Secure (encrypted) binary source files
- Obfuscated binary file: Obfuscates all names and other documentation related to a specific model instance

GAMS

Obfuscated Binary File - Example

Solver Log

Normal

```
Reduced LP has 5 rows, 6 columns, and 12 nonzeros.
Presolve time = 0.00 sec. (0.00 ticks)

Iteration      Dual Objective            In Variable           Out Variable
     1             73.125000    x(seattle.new-york) demand(new-york) slack
     2            119.025000     x(seattle.chicago)  demand(chicago) slack
     3            153.675000    x(san-diego.topeka)   demand(topeka) slack
     4            153.675000  x(san-diego.new-york)  supply(seattle) slack
LP status(1): optimal
```

Obfuscated

```
Reduced LP has 5 rows, 6 columns, and 12 nonzeros.
Presolve time = 0.00 sec. (0.00 ticks)

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
```

GAMS

Obfuscated Binary File - Example

Looking at Results

Normal

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	

Obfuscated

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	

GAMS

Obfuscated Binary File - Example

New options

- `saveobfuscate` (`so`) and `xsaveobfuscate` (`xso`): generate obfuscated binary file (regular or compressed)
- `restartNamed` (`rn`) : brings back original names when restarting from an obfuscated binary file

Example

- Compile GAMS model into named and obfuscated binary file:
  `gams trnsport a=c s=0named saveobfuscate=0anon`
- Move obfuscated binary file to non-secure machine, execute it, and save (obfuscated) to a gdx file: `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:
  `gams dummy r=1anon restartNamed=0named gdx=res`

Model2Tex

- Translates a GAMS model instance into Latex Format

```
Sets
     i   canning plants    / seattle, san-diego /
     j   markets           / new-york, chicago, topeka
Parameters
     a(i)  capacity of plant i in cases
     b(j)  demand at market j in cases
     c(i,j)  transport cost in thousands of dollars pe
Variables
     x(i,j)  shipment quantities in cases
     z       total transportation costs in thousands o
Equations
     cost         define objective function
     supply(i)    observe supply limit at plant i
     demand(j)    satisfy demand at market j ;

cost ..        z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
supply(i) ..   sum(j, x(i,j))   =l=  a(i) ;
demand(j) ..   sum(i, x(i,j))   =g=  b(j) ;

Model transport /all/ ;
```

gams trnsport docfile=trans
model2tex trans

Symbols

Sets

Name	Domains	Description
i	*	canning plants
j	*	markets

Parameters

Name	Domains	Description
a	i	capacity of plant i in cases
b	j	demand at market j in cases
c	i, j	transport cost in thousands of dollars per case

Variables

Name	Domains	Description
x	i, j	shipment quantities in cases
z		total transportation costs in thousands of dollars

Equations

Name	Domains	Description
cost		define objective function
supply	i	observe supply limit at plant i
demand	j	satisfy demand at market j

Equation Definitions

cost

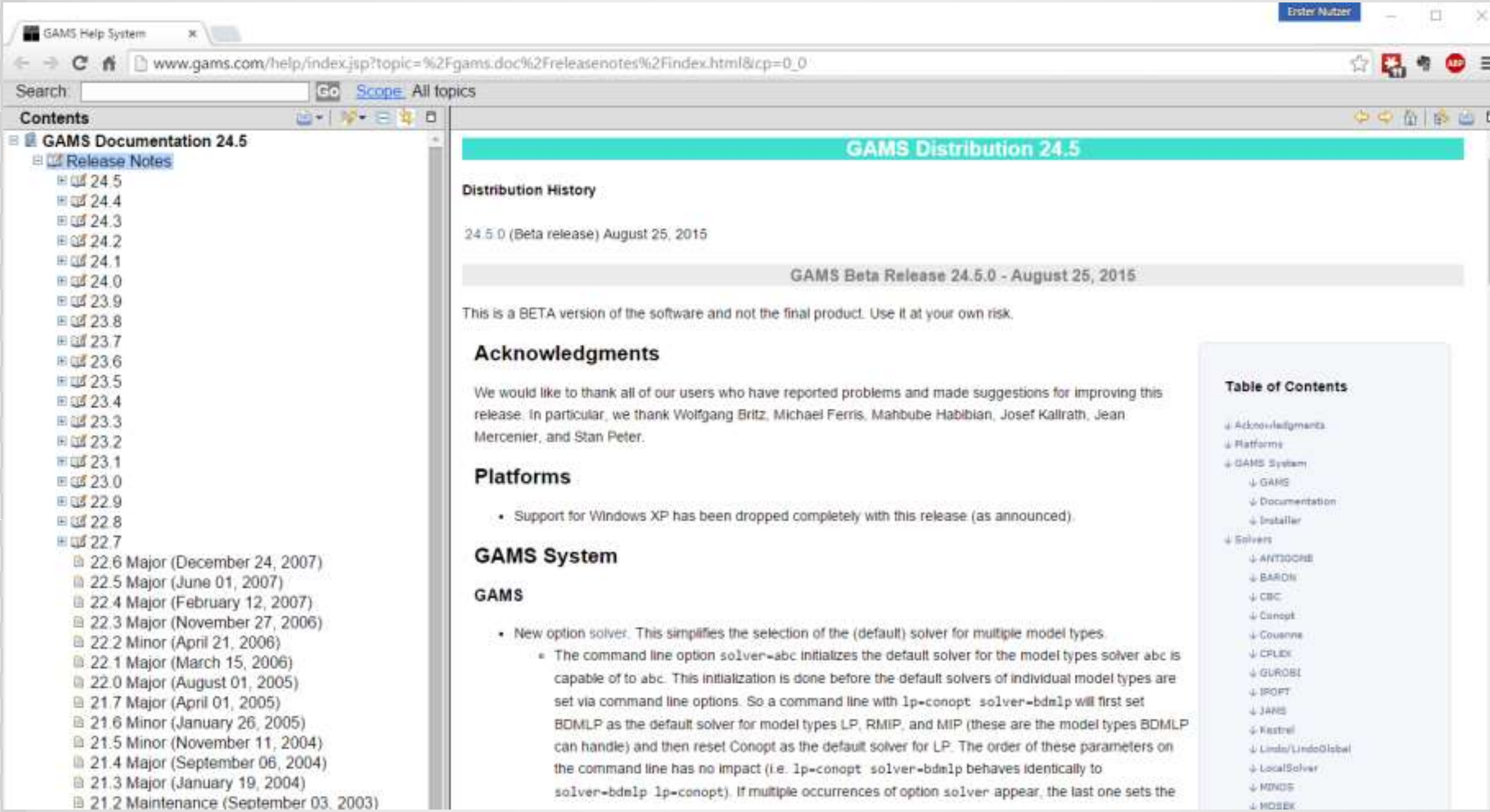
$$z = \sum_{i,j}(c_{i,j} \cdot x_{i,j})$$

$supply_i$

$$\sum_{j}(x_{i,j}) \leq a_i \qquad \forall i$$

$demand_j$

What else is new?



➢ http://www.gams.com/help/index.jsp

Summary

GAMS – Elements, Examples, and Enhancements

- **Simple, but powerful language**
- Designed to **interact,** different **layers** (platform, solver, data, interface)
- **Evolution** through more than 35 years of R&D and user feedback
- **Maturity** through experience and rigorous testing

Striving for Innovation and Compatibility

- Provide **cutting edge technology**
- **Increase productivity**
- Don't **lock** developers and users into a certain environment
- **Protect** investments of users

GAMS

Thank You

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

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