

# GAMS – An Introduction

---

**Steven Dirkse & Lutz Westermann**

GAMS Development Corp. / GAMS Software GmbH

# Agenda

**GAMS at a Glance**

**GAMS – Hands On Examples**

**Outlook**

- **Deployment of GAMS Models**

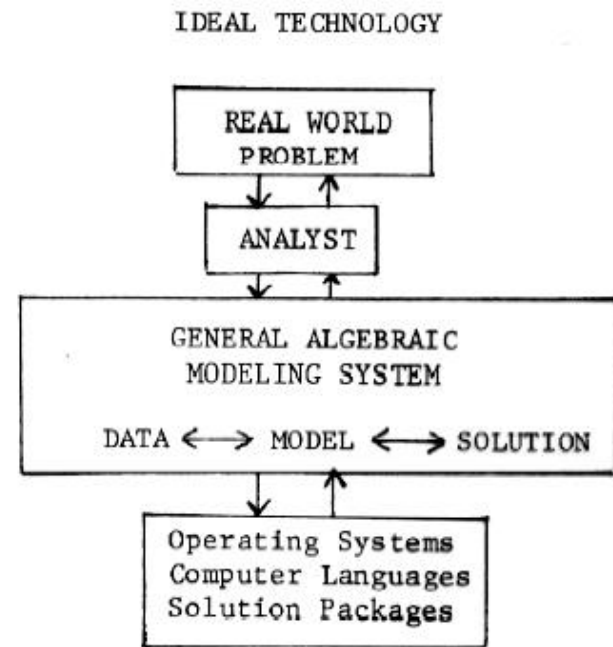
- APIs – Application Programming Interfaces to GAMS
  - Low Level APIs
- Object Oriented APIs
  - Using R/Shiny to deploy and visualize GAMS models in a Web Interface

# GAMS at a Glance

# 1976 - A World Bank Slide

## The Vision

GAMS came into being!



- RESULT:
- Limited drain of resources
  - Same representation of models for humans and machines
  - Model representation is also model documentation

***The aim of this system is to provide one representation of a model which is easily understood by both humans and machines.***

[J. Bisschop, A. Meeraus, On the Development of a General Algebraic Modeling System in a Strategic Planning Environment. *Mathematical Programming Study* 20 (**1982**) 1-29.]

***Self-documenting model. A GAMS model is a machine-executable documentation of an optimization problem.***

[M. Bussieck & A. Meeraus, Algebraic Modeling for IP and MIP (GAMS), *Annals of Operations Research* 149(1): *History of Integer Programming: Distinguished Personal Notes and Reminiscences*, Guest Editors: Kurt Spielberg and Monique Guignard-Spielberg, February, **2007**, pp. 49-56 ]

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

# Company

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

# Broad User Community and Network

14,000+ licenses

Users: 50% academic, 50% commercial

GAMS used in more than 120 countries

Uniform interface to ~40 solvers



**30+ Years**  
GAMS Development



# Broad Range of **Application Areas**

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

# Foundation of GAMS



Powerful algebraic modeling language

Open architecture and interfaces to other systems, independent layers

# Declarative and Procedural Language Elements

## Declarative elements

- Similar to mathematical notation
- Easy to learn - few basic language elements: sets, parameters, variables, equations, models
- Model is executable (algebraic) description of the problem

## Procedural elements

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

# Independence of Model and Operating System



**Platforms supported by GAMS:**

**➡ Models can be moved between platforms with ease!**

# Cross Platform GUI – GAMS Studio

- Open source Qt project (Mac/Linux/Win)
  - Published on GitHub under GPL
- Released in beta status
- Group Explorer
- Editor / Syntax coloring / Spell checks

- Tree view / Syntax-error navigation
- Option Editor
- Integrated Help
- Model Debugging & Profiling
- Solver selection & setup
- Data viewer
- GAMS Processes Control

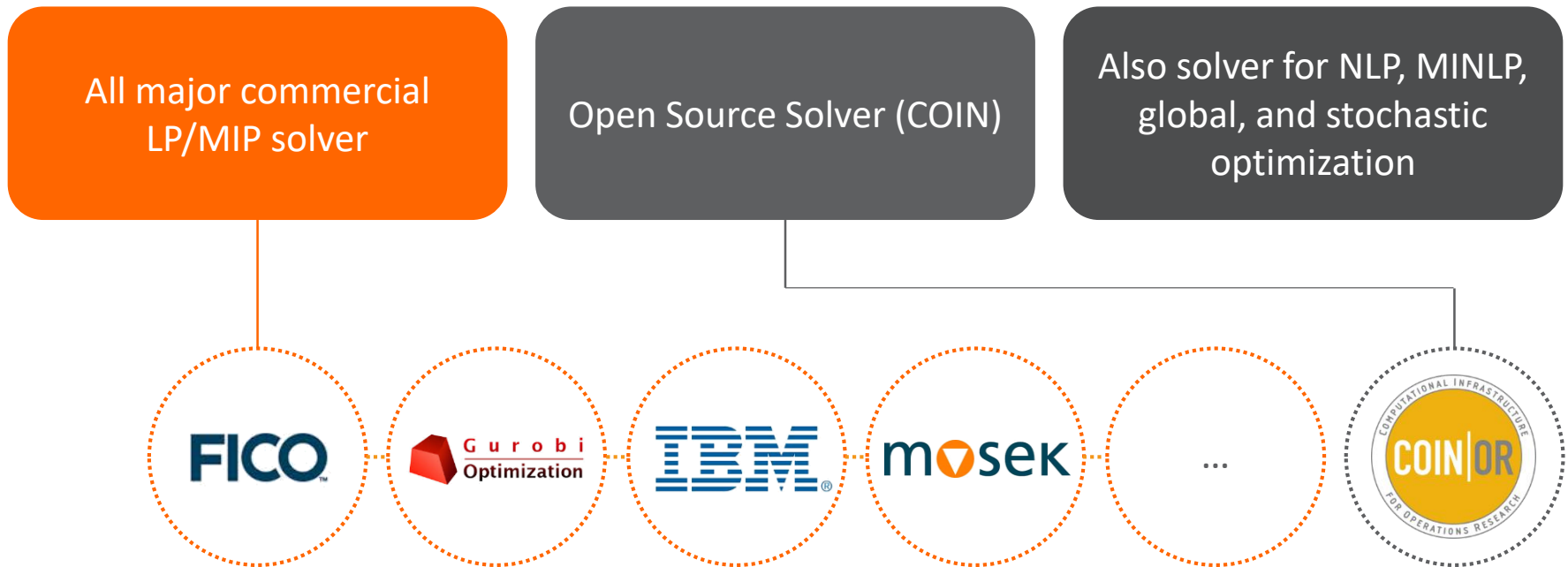
The screenshot displays the GAMS Studio interface with the following components:

- Option Editor:** Shows 'SolveLink' set to 5 and 'LP' set to 'Xpress'.
- Explorer:** A tree view on the left showing a project structure with folders like 'bendersMPI', 'spBendersSeq', and 'tnsport', and files like 'tnsport.gms'.
- Editor:** The central window showing the GAMS code for 'tnsport.gms'. The code includes parameters for freight cost, shipment quantities, and total transportation costs, along with equations for the objective function and supply/demand constraints.
- Output:** The rightmost window showing the solver's output. It reports 12 non-zero elements, 0 entities, and 0 sets. The final objective value is 153.675, and the solution is found after 3 simplex iterations.

At the bottom, a status bar indicates the file path 'C:\Users\User\Documents\GAMSStudio\workspace\tnsport.gms', line 67, and encoding 'UTF-8'.

# Independence of Model and Solver

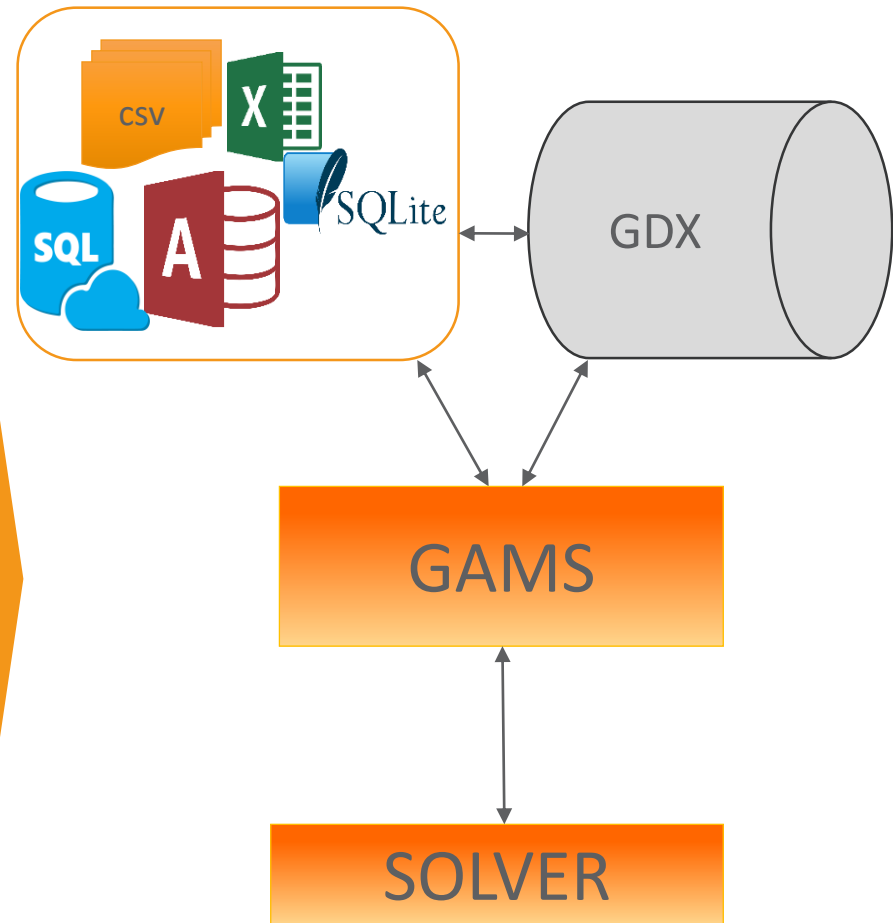
One environment for a wide range of model types and solvers



➔ 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
  - ...



# Independence of Model and User Interface

## API's

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





# Free Model Libraries

**Model Libraries**   **Help**

- GAMS Model Library
- GAMS Test Library
- GAMS Data Utilities Models
- GAMS EMP Library
- GAMS API Library
- Practical Financial Optimization Models
- Nonlinear Optimization Applications (N. Andrei)

➤ **More than 1400 models!**

◀ 17 ▶

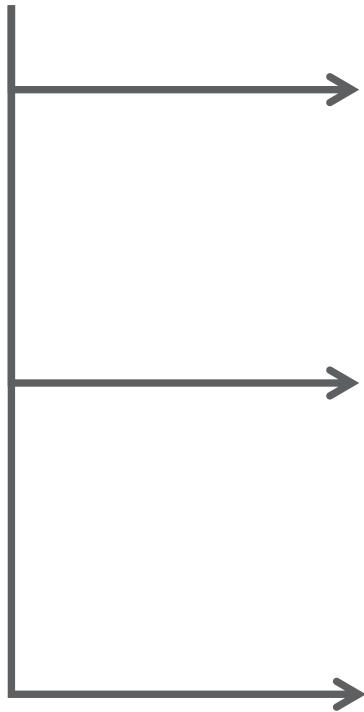
# Why GAMS?

- Experience of 30+ years
- Broad user community from different areas
- Lots of model templates
- Strong development interface
- Consistent implementation of design principles
  - Simple, but powerful modeling language
  - Independent layers
  - Open architecture: Designed to interact with other applications
- Open for new developments
- Protecting investments of users

# GAMS – Hands On Examples

# A Simple Transportation Problem

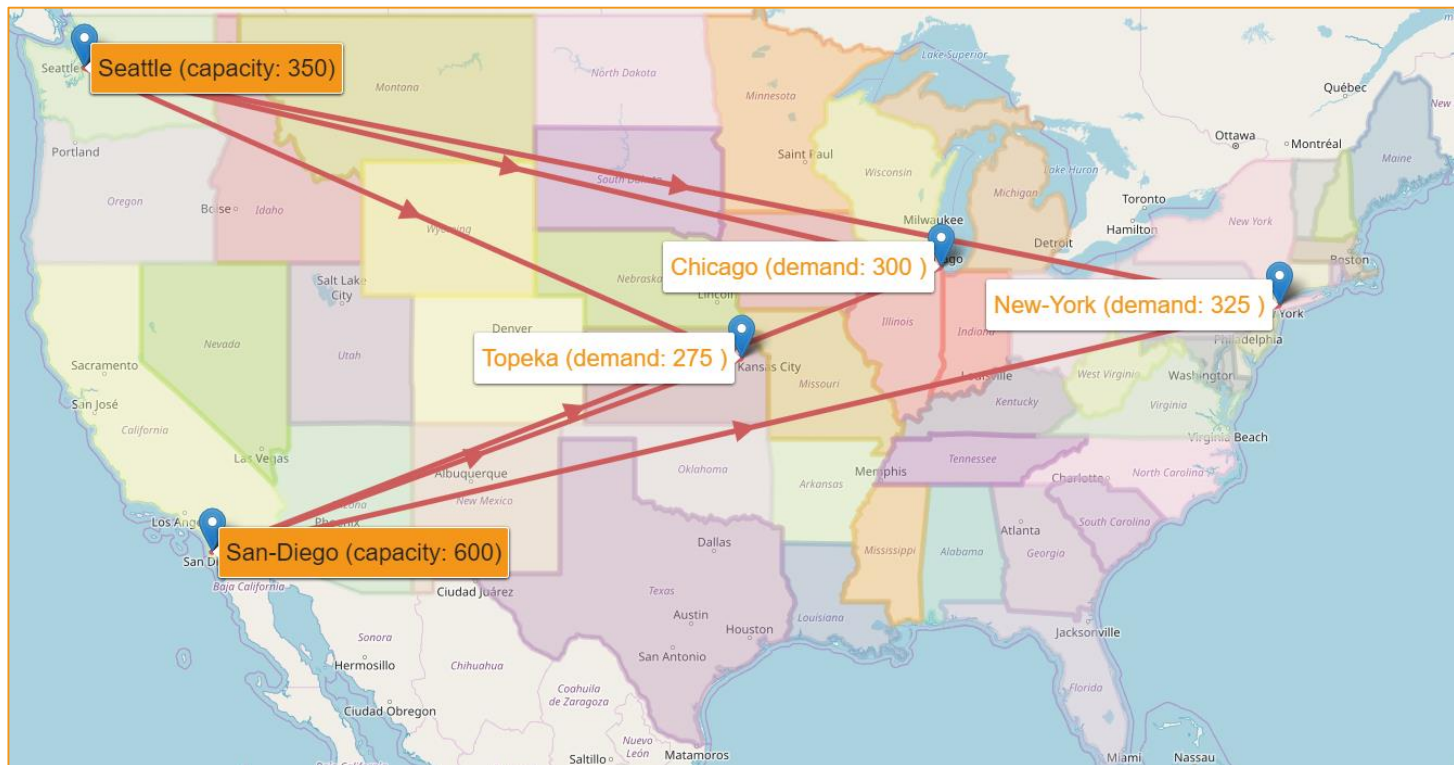
- What does this example show?



- It gives a first glimpse of how a problem can be formulated in GAMS
- It shows some basics of data exchange with GAMS
- It shows how easy it is to change model type and, consequently, solver technology

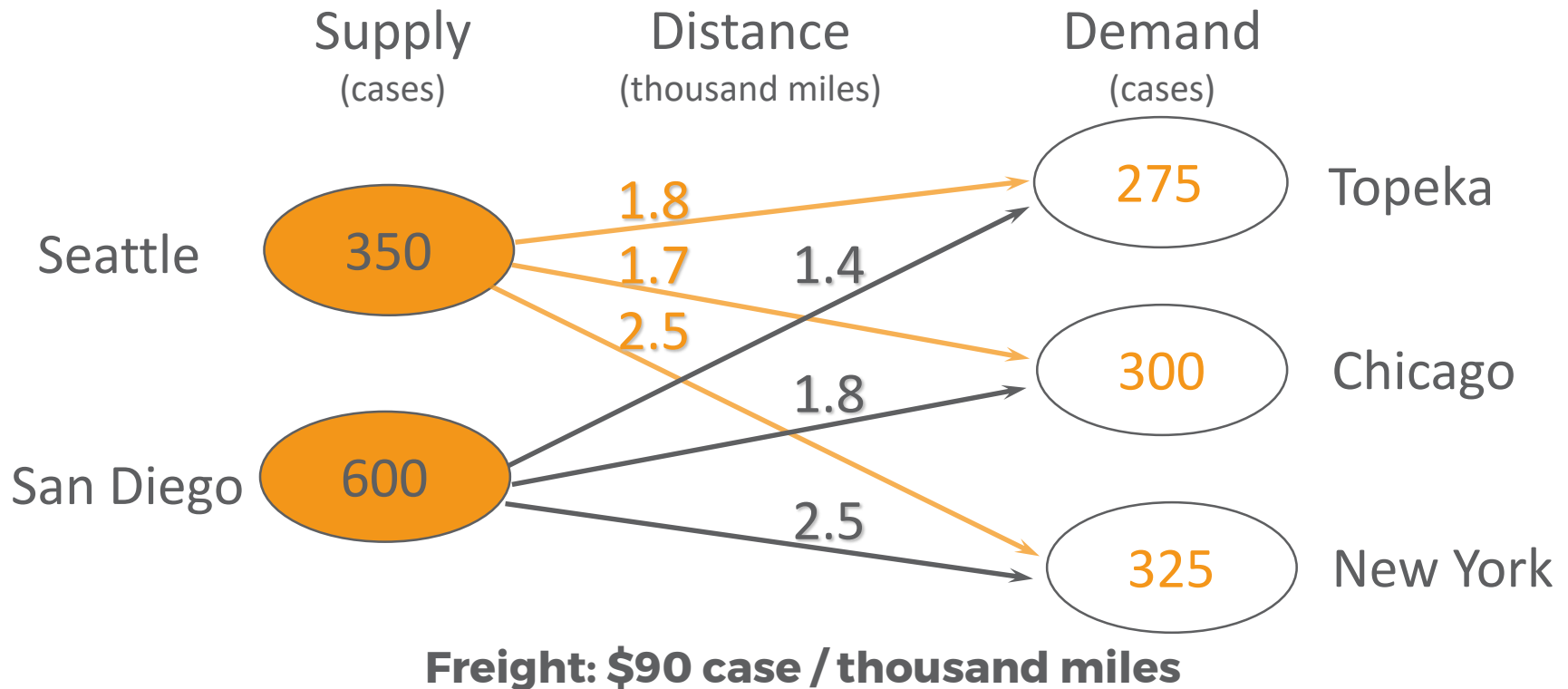


# A Simple Transportation Problem



**Freight: \$90 case / thousand miles**

# A Simple Transportation Problem



Minimize  
subject to

Transportation cost  
Demand satisfaction at markets  
Supply constraints

# Mathematical Model Formulation

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)

$b_j$  (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)

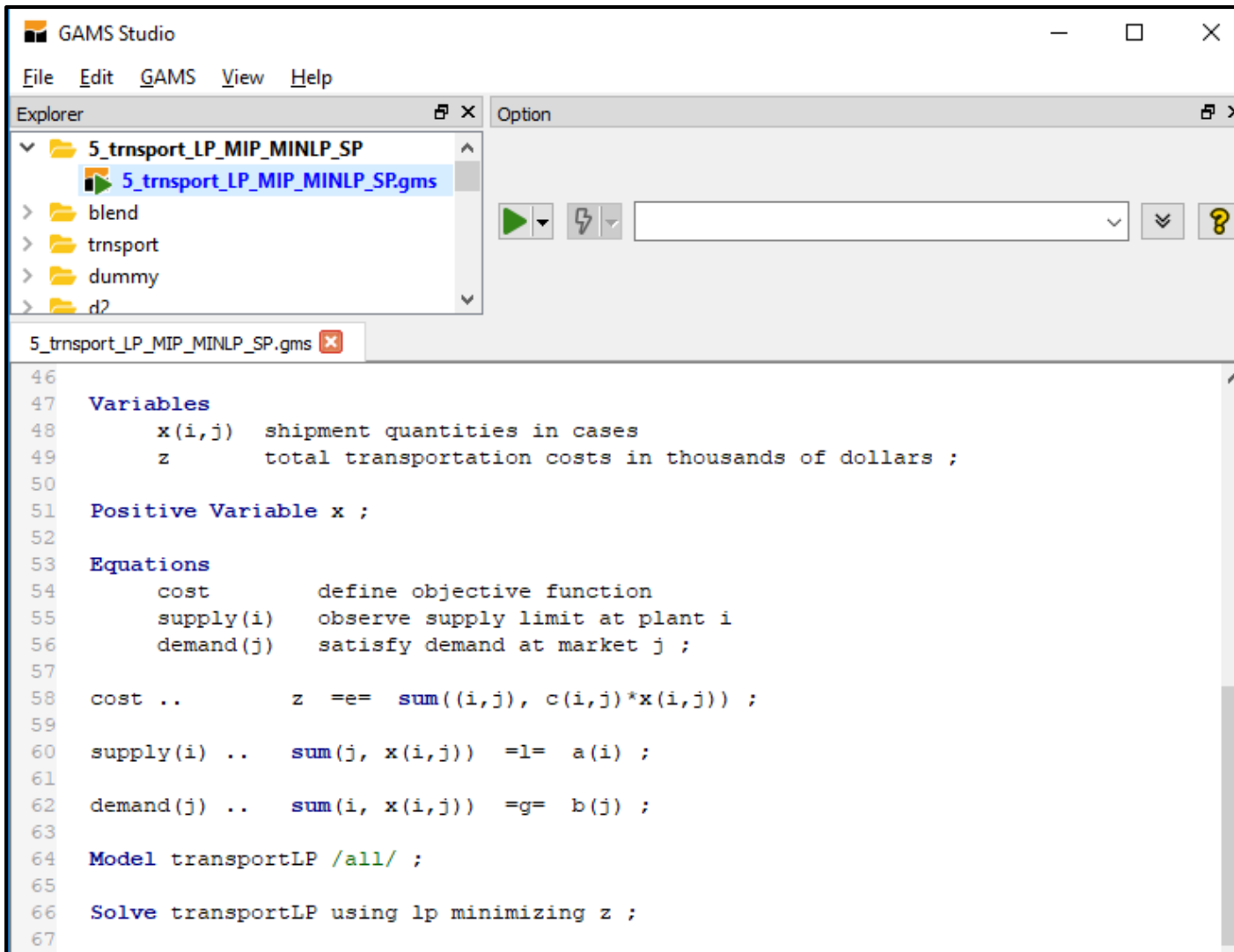
$\sum_i x_{ij} \geq b_j \quad \forall j$  (Shipments to each market  $\geq$  demand)

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

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



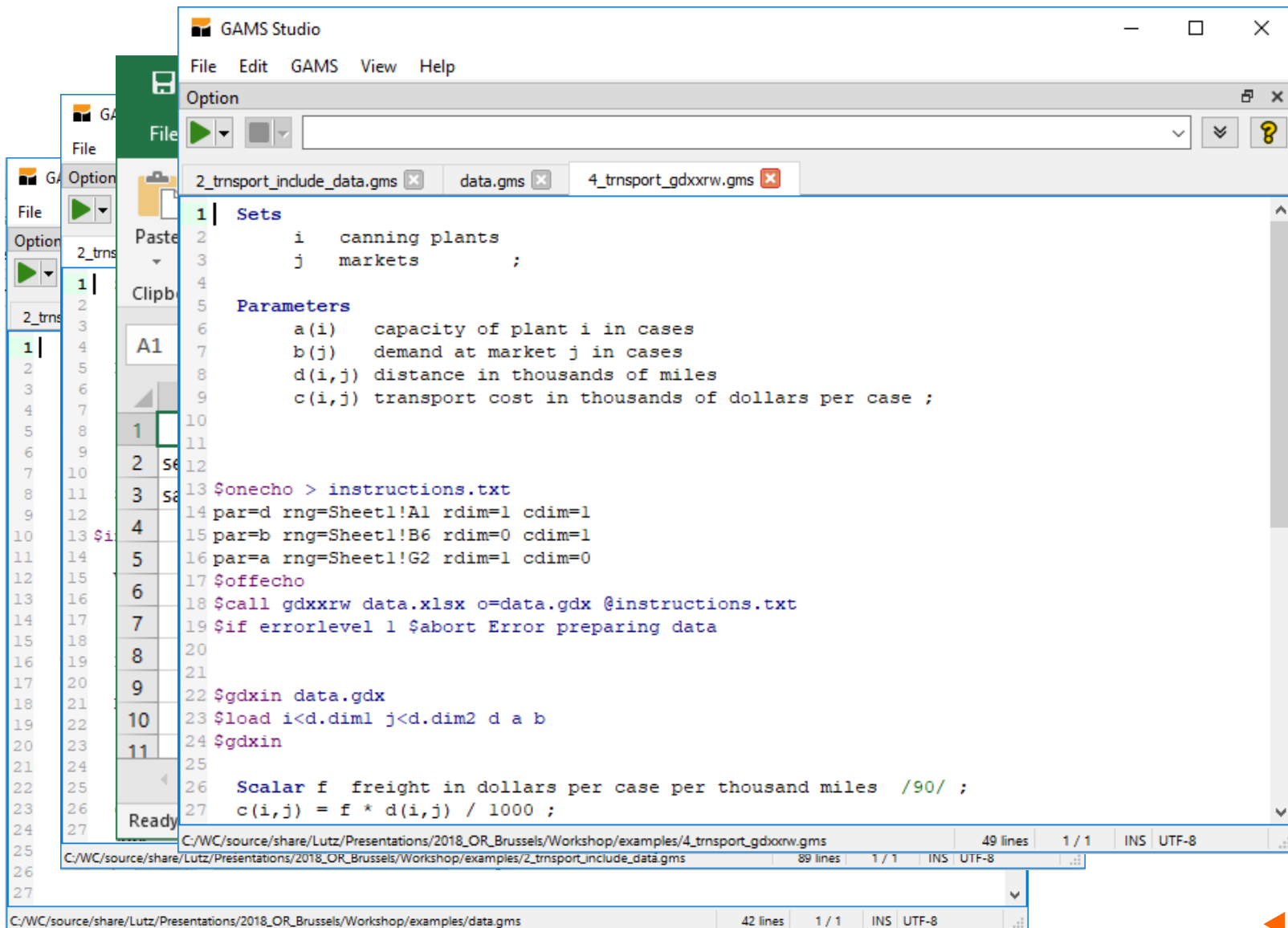
# GAMS Syntax (LP Model)



The screenshot displays the GAMS Studio application window. The 'Explorer' pane on the left shows a project structure with folders 'blend', 'transport', 'dummy', and 'd2', and a file '5\_trnsport\_LP\_MIP\_MINLP\_SP.gms' selected. The 'Option' pane on the right contains execution controls. The main code editor shows the following GAMS script:

```
46
47 Variables
48     x(i,j)  shipment quantities in cases
49     z        total transportation costs in thousands of dollars ;
50
51 Positive Variable x ;
52
53 Equations
54     cost      define objective function
55     supply(i) observe supply limit at plant i
56     demand(j) satisfy demand at market j ;
57
58 cost ..      z  =e= sum((i,j), c(i,j)*x(i,j)) ;
59
60 supply(i) ..  sum(j, x(i,j))  =l= a(i) ;
61
62 demand(j) ..  sum(i, x(i,j))  =g= b(j) ;
63
64 Model transportLP /all/ ;
65
66 Solve transportLP using lp minimizing z ;
67
```

# GAMS Syntax (Data Input)



```
1 | Sets
2     i   canning plants
3     j   markets      ;
4
5 | Parameters
6     a(i)   capacity of plant i in cases
7     b(j)   demand at market j in cases
8     d(i,j) distance in thousands of miles
9     c(i,j) transport cost in thousands of dollars per case ;
10
11
12
13 $onecho > instructions.txt
14 par=d rng=Sheet1!A1 rdim=1 cdim=1
15 par=b rng=Sheet1!B6 rdim=0 cdim=1
16 par=a rng=Sheet1!G2 rdim=1 cdim=0
17 $offecho
18 $call gdxrw data.xlsx o=data.gdx @instructions.txt
19 $if errorlevel 1 $abort Error preparing data
20
21
22 $gdxin data.gdx
23 $load i<d.dim1 j<d.dim2 d a b
24 $gdxin
25
26 Scalar f   freight in dollars per case per thousand miles  /90/ ;
27 c(i,j) = f * d(i,j) / 1000 ;
```

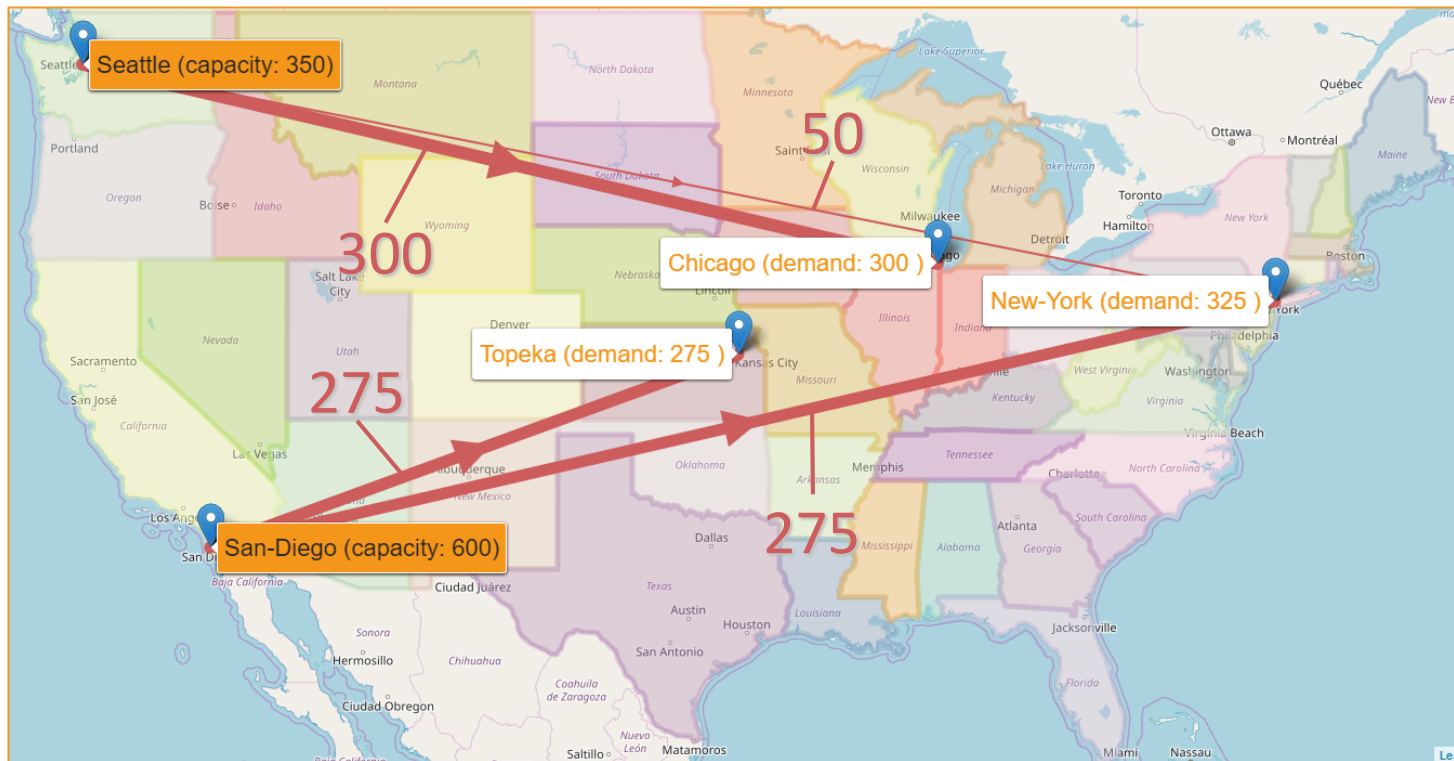
The screenshot shows the GAMS Studio interface with three open files: 2\_trnsport\_include\_data.gms, data.gms, and 4\_trnsport\_gdxxrw.gms. The main editor displays the code for 4\_trnsport\_gdxxrw.gms. The code defines sets 'i' (canning plants) and 'j' (markets), and parameters 'a(i)' (capacity), 'b(j)' (demand), 'd(i,j)' (distance), and 'c(i,j)' (transport cost). It then uses the \$onecho command to execute a program 'gdxrw' with specific data loading instructions, followed by \$gdxin to load the data into the model. The status bar at the bottom shows the file paths and line counts for each open file.

# Solution to LP model

**Canning Plants (supply)**

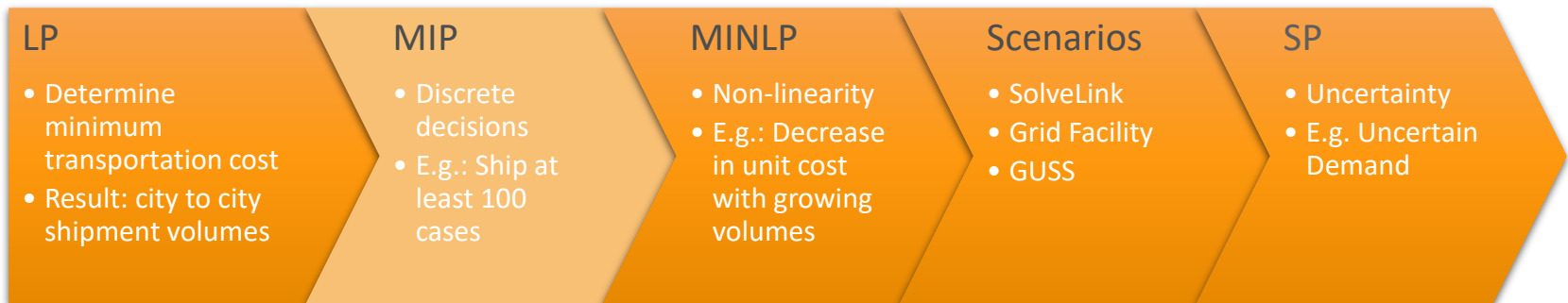
shipments  
→  
(Number of cases)

**Markets (demand)**



**Freight: \$90 case / thousand miles**

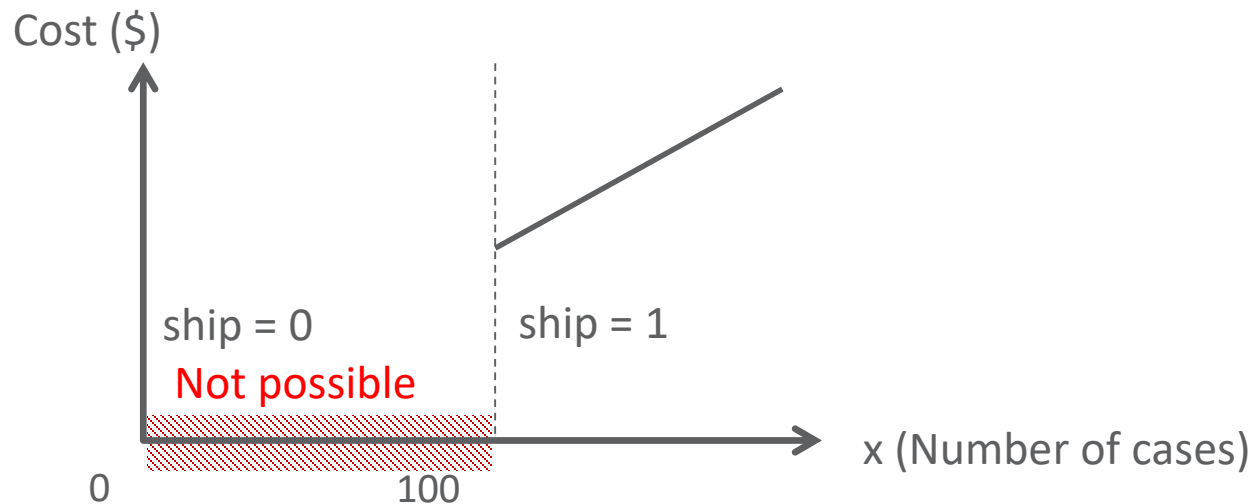
**Total cost: \$153,675**



# MIP Model:

## Minimum Shipment of 100 cases

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



add constraints:

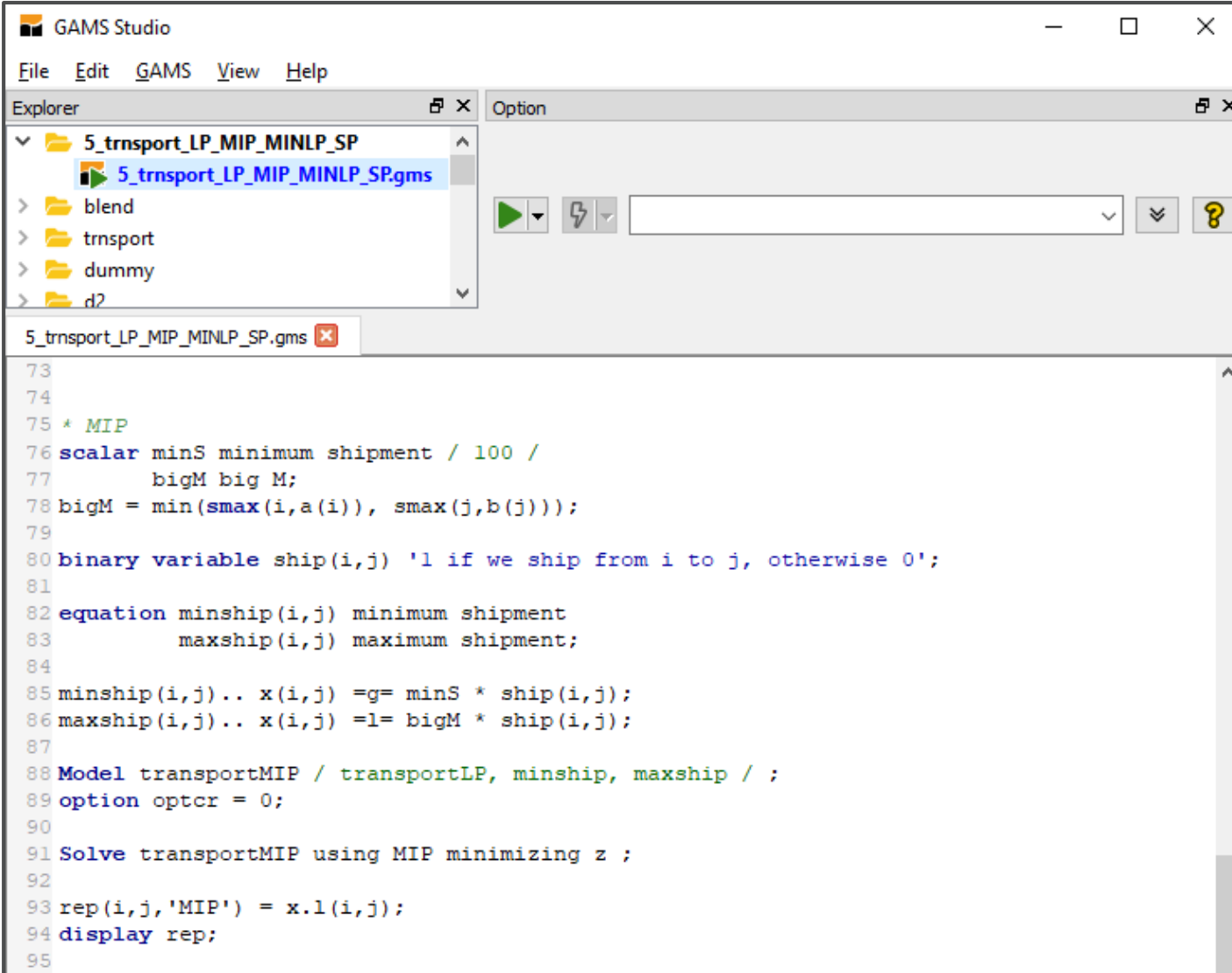
$$x_{i,j} \geq 100 \cdot ship_{i,j} \quad \forall i,j \quad (\text{if } ship=1, \text{ then ship at least 100})$$

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

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

**Hands-On**

# MIP Model: GAMS Syntax



The screenshot displays the GAMS Studio application window. The 'Explorer' pane on the left shows a project structure with folders 'blend', 'transport', 'dummy', and 'd2', and a file '5\_trnsport\_LP\_MIP\_MINLP\_SP.gms' selected. The 'Option' pane on the right contains execution controls like a play button, a lightning bolt icon, a dropdown menu, and a help icon. The main code editor shows the following GAMS code:

```
73
74
75 * MIP
76 scalar minS minimum shipment / 100 /
77      bigM big M;
78 bigM = min(smax(i,a(i)), smax(j,b(j)));
79
80 binary variable ship(i,j) '1 if we ship from i to j, otherwise 0';
81
82 equation minship(i,j) minimum shipment
83      maxship(i,j) maximum shipment;
84
85 minship(i,j).. x(i,j) =g= minS * ship(i,j);
86 maxship(i,j).. x(i,j) =l= bigM * ship(i,j);
87
88 Model transportMIP / transportLP, minship, maxship / ;
89 option optcr = 0;
90
91 Solve transportMIP using MIP minimizing z ;
92
93 rep(i,j,'MIP') = x.l(i,j);
94 display rep;
95
```

# MIP Model: Results

GAMS Studio

File Edit GAMS View Help

Explorer

- 5\_trnsport\_LP\_MIP\_MINLP\_SP
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.lst
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.gms
- blend
- transport

Option

5\_trnsport\_LP\_MIP\_MINLP\_SP.gms 5\_trnsport\_LP\_MIP\_MINLP\_SP.lst 5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx

```

481 seattle .chicago      .      1.0000      1.0000      EPS
482 seattle .topeka        .      1.0000      1.0000      EPS
483 san-diego.new-york     .      1.0000      1.0000      EPS
484 san-diego.chicago     .      1.0000      1.0000      EPS
485 san-diego.topeka      .      1.0000      1.0000      EPS
486
487
488 **** REPORT SUMMARY :      0      NONOPT
489                          0      INFEASIBLE
490                          0      UNBOUNDED
491 GAMS 25.2.0 r67480 ALFA Released 2Aug18 WEX-
492 A Transportation Problem (TRANSPORT,SEQ=1)
493 E x e c u t i o n
494
495
496 ----      94 PARAMETER rep      report parameter
497
498                          LP      MIP
499
500 seattle .new-york        50.000
501 seattle .chicago        300.000      300.000
502 san-diego.new-york       275.000      325.000
503 san-diego.topeka        275.000      275.000
504
505
506
507 EXECUTION TIME      =      0.000 SECONDS
  
```

C:/WC/source/share/Lutz/Presentations/2018\_OR\_Brussels/Workshop/examples/5\_trnsport\_LP\_MIP\_MINLP\_SP.lst

GAMS Studio

File Edit GAMS View Help

Explorer

- 5\_trnsport\_LP\_MIP\_MINLP\_SP
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.lst
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.gms
- blend
- transport

Option

5\_trnsport\_LP\_MIP\_MINLP\_SP.gms 5\_trnsport\_LP\_MIP\_MINLP\_SP.lst 5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx

Entry	Name	Type	Dim	Records	Text	i	j	LP	Value
3	a	Parameter	1	2	capacity of plant i in c...	seattle	new-york	LP	50
4	b	Parameter	1	3	demand at market j in ...	seattle	chicago	LP	300
15	bigM	Parameter	0	1	big M	san-diego	new-york	LP	275
7	c	Parameter	2	6	transport cost in thous...	san-diego	topeka	LP	275
10	cost	Equation	0	1	define objective functi...	seattle	chicago	MIP	300
5	d	Parameter	2	6	distance in thousands ...	san-diego	new-york	MIP	325
12	demand	Equation	1	3	satisfy demand at mar...	san-diego	topeka	MIP	275
6	f	Parameter	0	1	freight in dollars per ca...				
1	i	Set	1	2	canning plants				
2	j	Set	1	3	markets				
18	maxship	Equation	2	6	maximum shipment				
14	minS	Parameter	0	1	minimum shipment				
17	minship	Equation	2	6	minimum shipment				
13	rep	Parameter	3	7	report parameter				
16	ship	Variable	2	6	1 if we ship from i to j, ...				
11	supply	Equation	1	2	observe supply limit at...				
8	x	Variable	2	6	shipment quantities in ...				
9	z	Variable	0	1	total transportation co...				

Symbol Search:  ☐ All Columns ☐ Squeeze Defaults

C:/WC/source/share/Lutz/Presentations/2018\_OR\_Brussels/Workshop/examples/5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx

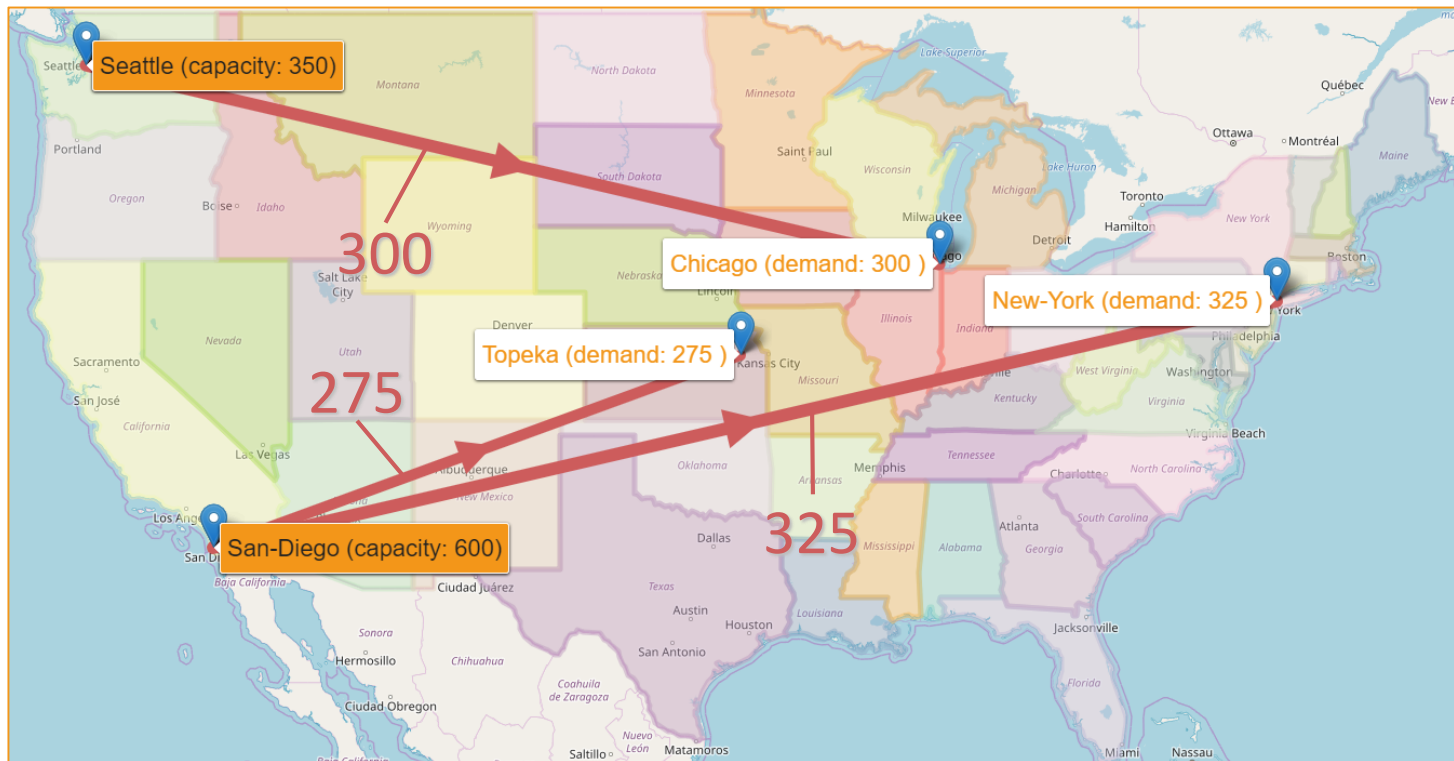
# MIP Model: Solution

**Canning Plants (supply)**

shipments

(Number of cases)

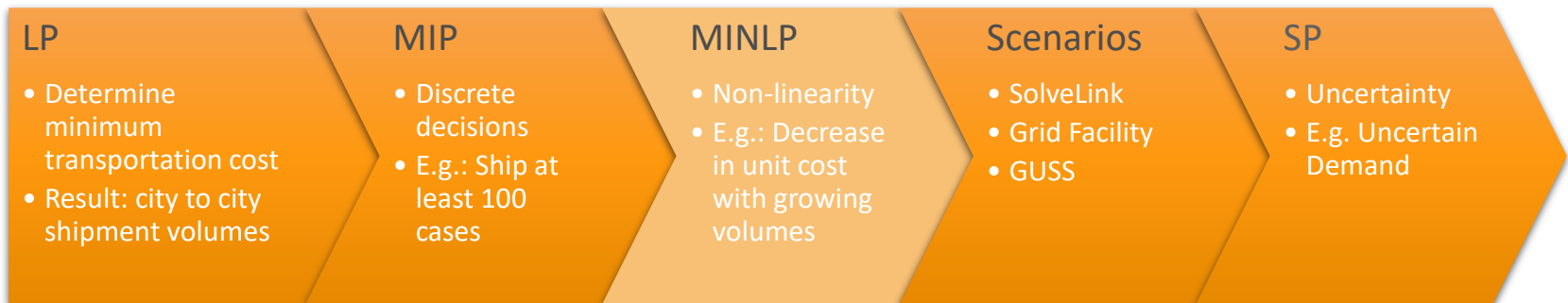
**Markets (demand)**



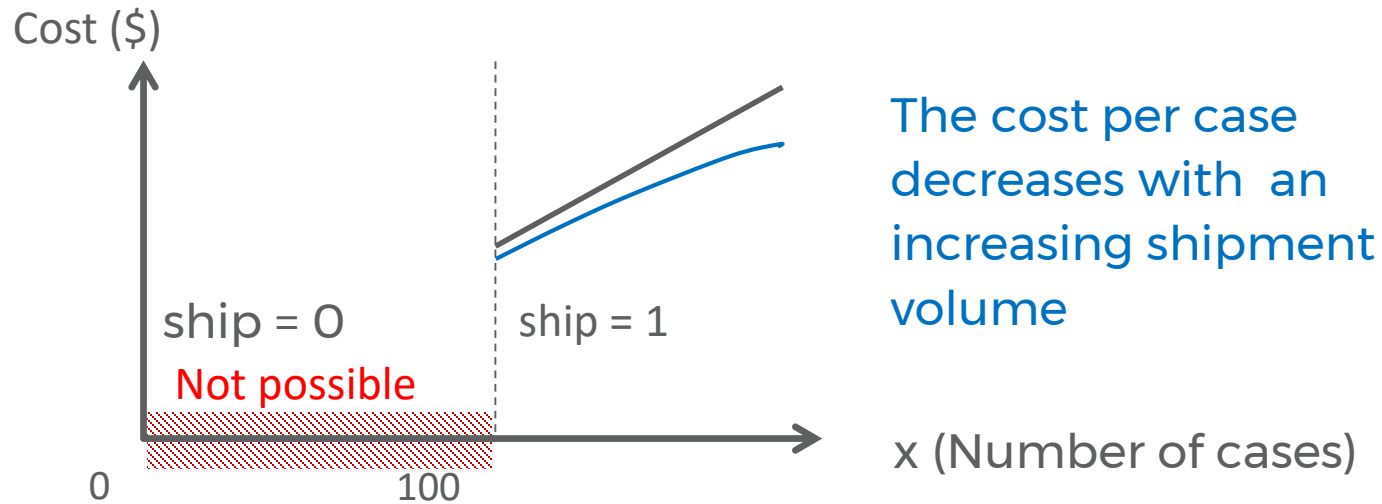
Freight: \$90 case / thousand miles

Total cost: \$153,675





# MINLP: Cost Savings



## Replace:

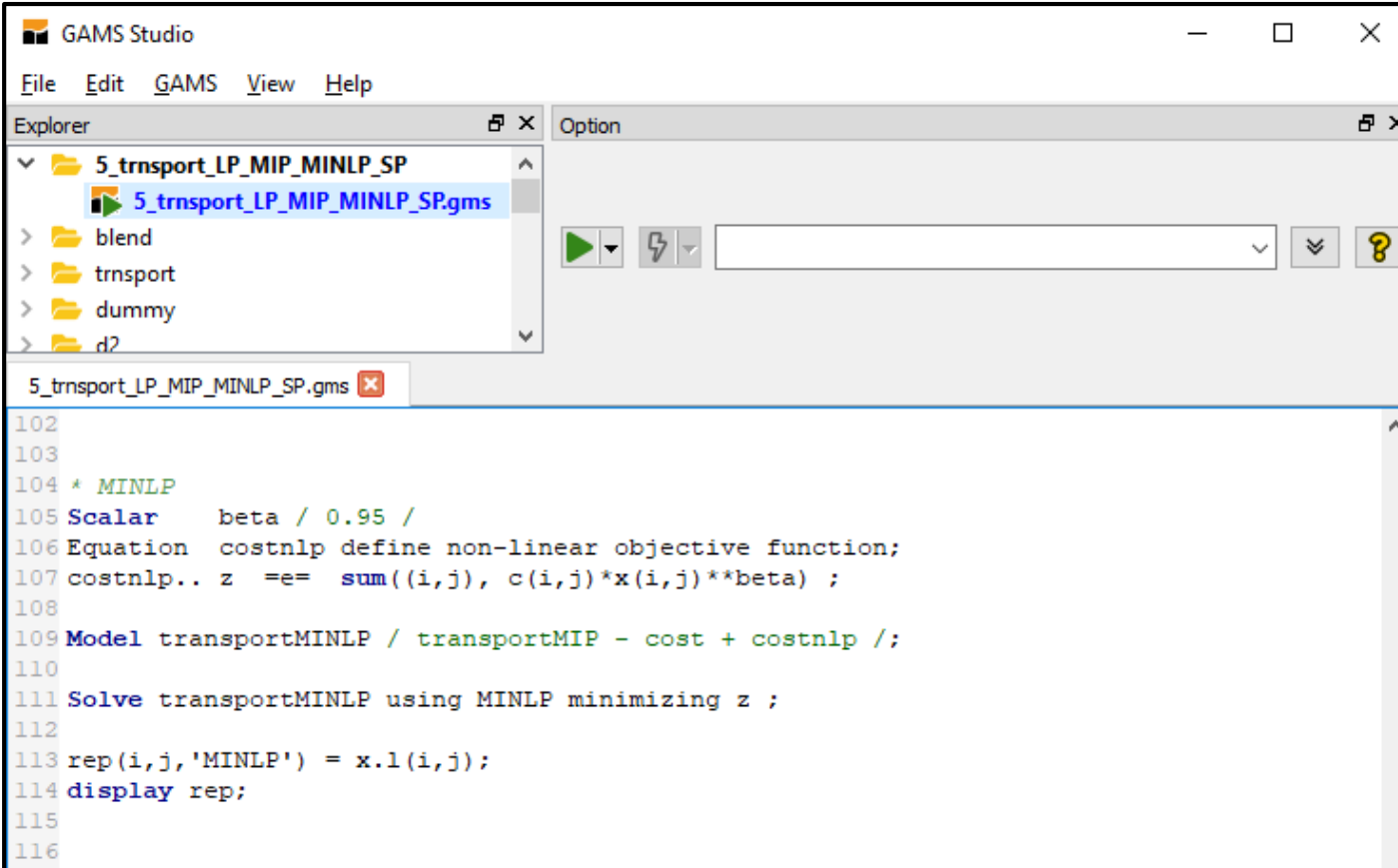
$\min \sum_i \sum_j c_{ij} \cdot x_{ij}$  (Minimize total transportation cost)

With

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

**Hands-On**

# MINLP Model: GAMS Syntax



The screenshot displays the GAMS Studio application window. The 'Explorer' pane on the left shows a project structure with folders 'blend', 'transport', 'dummy', and 'd2', and a file '5\_transport\_LP\_MIP\_MINLP\_SP.gms' selected. The 'Option' pane on the right contains a dropdown menu and a search icon. The main code editor shows the following GAMS code:

```
102
103
104 * MINLP
105 Scalar    beta / 0.95 /
106 Equation  costnlp define non-linear objective function;
107 costnlp.. z  =e= sum((i,j), c(i,j)*x(i,j)**beta) ;
108
109 Model transportMINLP / transportMIP - cost + costnlp /;
110
111 Solve transportMINLP using MINLP minimizing z ;
112
113 rep(i,j,'MINLP') = x.l(i,j);
114 display rep;
115
116
```

# MINLP Model: Results

GAMS Studio

File Edit GAMS View Help

Explorer

- 5\_trnsport\_LP\_MIP\_MINLP\_SP
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.lst
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.gms
- blend
- transport

Option

5\_trnsport\_LP\_MIP\_MINLP\_SP.gms 5\_trnsport\_LP\_MIP\_MINLP\_SP.lst 5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx

```

773 seattle .topeka . . 1.0000 EPS
774 san-diego.new-york . . 1.0000
775 san-diego.chicago . .
776 san-diego.topeka . . 1.0000
777
778
779 **** REPORT SUMMARY :      0      NONOPT
780                          0      INFEASIBLE
781                          0      UNBOUNDED
782                          0      ERRORS
783 GAMS 25.2.0 r67480 ALFA Released 2Aug18 WEX-VS8 x86 32b
784 A Transportation Problem (TRANSPORT,SEQ=1)
785 E x e c u t i o n
786
787
788 ---- 113 PARAMETER rep report parameter
789
790                      LP      MIP      MINLP
791
792 seattle .new-york      50.000
793 seattle .chicago      300.000      300.000      300.000
794 san-diego.new-york      275.000      325.000      325.000
795 san-diego.topeka      275.000      275.000      275.000
796
797
798
799 EXECUTION TIME      =      0.000 SECONDS      3 MB 25
  
```

C:\WC\source\share\Lutz\Presentations\2018\_OR\_Brussels\Workshop\examples\5\_trnsport\_LP\_MIP\_MINLP\_SP.lst

GAMS Studio

File Edit GAMS View Help

Explorer

- 5\_trnsport\_LP\_MIP\_MINLP\_SP
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.lst
  - 5\_trnsport\_LP\_MIP\_MINLP\_SP.gms
- blend
- transport

Option

5\_trnsport\_LP\_MIP\_MINLP\_SP.gms 5\_trnsport\_LP\_MIP\_MINLP\_SP.lst 5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx

Entry	Name	Type	Dim	Records	Text
3	a	Parameter	1	2	capacity of plant i i...
4	b	Parameter	1	3	demand at market j...
19	beta	Parameter	0	1	
15	bigM	Parameter	0	1	big M
7	c	Parameter	2	6	transport cost in th...
10	cost	Equation	0	1	define objective fu...
20	costnlp	Equation	0	1	define non-linear o...
5	d	Parameter	2	6	distance in thousan...
12	demand	Equation	1	3	satisfy demand at ...
6	f	Parameter	0	1	freight in dollars pe...
1	i	Set	1	2	canning plants
2	j	Set	1	3	markets
18	maxship	Equation	2	6	maximum shipment
14	minS	Parameter	0	1	minimum shipment
17	minship	Equation	2	6	minimum shipment
13	rep	Parameter	3	10	report parameter
16	ship	Variable	2	6	1 if we ship from i t...
11	supply	Equation	1	2	observe supply limi...

i	j		Value
seattle	new-york	LP	50
seattle	chicago	LP	300
san-diego	new-york	LP	275
san-diego	topeka	LP	275
seattle	chicago	MINLP	300
san-diego	new-york	MINLP	325
san-diego	topeka	MINLP	275
seattle	chicago	MIP	300
san-diego	new-york	MIP	325
san-diego	topeka	MIP	275

Symbol Search:  ☐ All Columns ☐ Squeeze Defaults

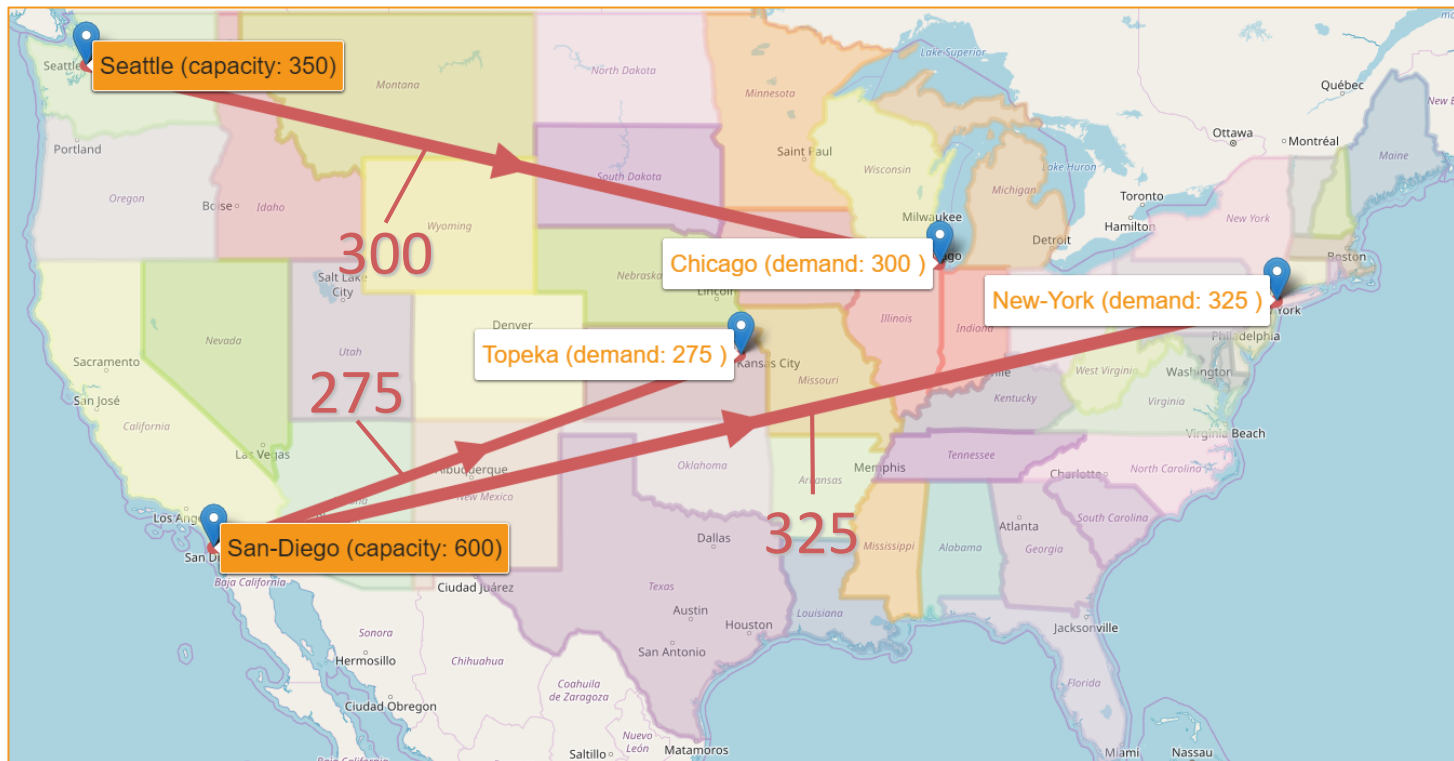
C:\WC\source\share\Lutz\Presentations\2018\_OR\_Brussels\Workshop\examples\5\_trnsport\_LP\_MIP\_MINLP\_SP.gdx

# MINLP Model: **Solution**

**Canning Plants** (supply)

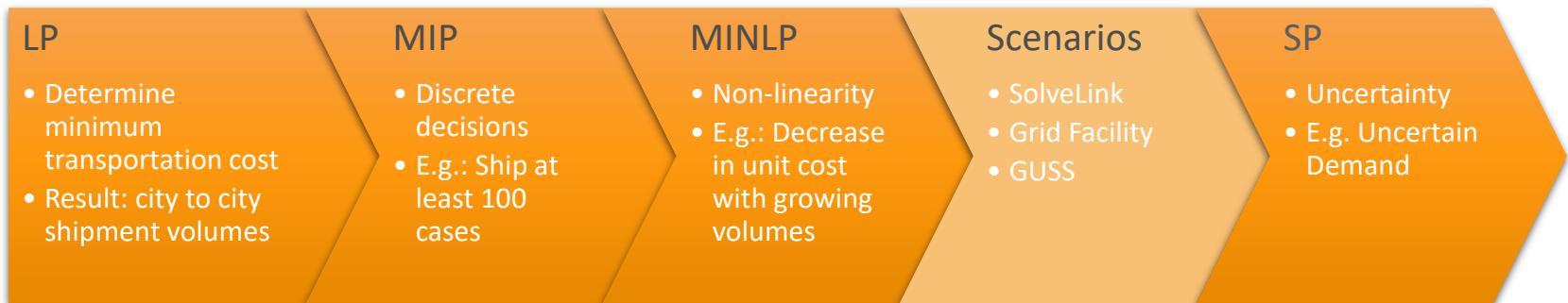
shipments  
→  
(Number of cases)

**Markets** (demand)



**Freight: \$90 case / thousand miles**

**Total cost: \$115,438**



# Motivation

- Solving challenging real-world problems often involves the solution of many optimization problems
  - Decomposition Methods
  - Scenario Analysis
  - Heuristics
  - ...
- Such approaches are often chosen, if solving the problem at hand does not work with a single monolithic model, e.g.
  - Due to its size and the required resources (e.g. memory)
  - Due to time restrictions (Problem should be solved in minutes but it takes days)
  - ...

→GAMS Grid Facility

→Gather-Update-Solve-Scatter

# SolveLink Option

Controls GAMS function when linking to solver.

```
Model transport /all/ ;  
transport.solveLink={0 %SolveLink.ChainScript%,  
1 %SolveLink.CallScript%,  
2 %SolveLink.CallModule%,  
3 %SolveLink.AsyncGrid%,  
4 %SolveLink.AsyncSimulate%,  
5 %SolveLink.LoadLibrary%,  
6 %solveLink.aSyncThreads%,  
7 %solveLink.threadsSimulate%};  
solve transport using lp minimizing z;
```



# SolveLink Option – Sequential Solves

- ChainScript [0]: Solver process, GAMS vacates memory
  - + Maximum memory available to solver
  - + protection against solver failure (*hostile* link)
  - swap to disk
- Call{Script [1]/Module [2]}: Solver process, GAMS stays live
  - + protection against solver failure (*hostile* link)
  - + no swap of GAMS database
  - file based model communication
- LoadLibrary [5]: Solver DLL in GAMS process
  - + fast memory based model communication
  - + update of model object inside the solver (hot start)
  - not supported by all solvers

# SolveLink Option Sequential – Exercise

- Generate 100 distance scenarios with random data
  - Hint: Look, e.g., at the GAMS function `uniform`
- Solve these scenarios with the solveLink values...
  - ChainScript [0]
  - CallModule [2]
  - LoadLibrary [5]
- Compare the execution time of solving all scenarios with different solveLink settings
  - Hint: Look at the GAMS function `jNow`

# SolveLink Option – Sequential Solves

```
Welcome x 6_transport_sovelink_seq.gms x 6_transport_sovelink_seq.lst x
64 Model transport /all/ ;
65
66 set s scenarios / s1*s100 /
67     sl solvelink / ChainScript, CallModule, LoadLibrary /;
68 parameter dd(s,i,j) distance by scenario
69     time(*) time for 100 scenarios
70     sl_val(sl) solvelink value / ChainScript %solveLink.chainScript%,
71                               CallModule %solveLink.CallModule%,
72                               LoadLibrary %solveLink.loadLibrary% /;
73 scalar tmp;
74
75 dd(s,i,j) = uniform(0.9,1.1)*d(i,j);
76
77 option limrow=0, limcol=0, solprint=silent;
78
79 * SERIAL SOLVE
80 loop(sl,
81     tmp = jnow;
82     transport.sovelink=sl_val(sl);
83     loop(s,
84         d(i,j) = dd(s,i,j);
85         Solve transport using lp minimizing z ;
86     );
87     time(sl) = (jnow-tmp)*24*60*60;
88 );
89 display time;
90
```

```
----      88 PARAMETER time    time for 100 scenarios
ChainScript 6.710,      CallModule 2.694,      LoadLibrary 0.578
```

# SolveLink Option – Asynchronous Solves

- aSyncGrid [3]: GAMS starts the solution and continues in a Grid computing environment
- aSyncThreads [6]: The problem is passed to the solver in core without use of temporary files, GAMS does not wait for the solver to come back

# SolveLink Option Asynchronous – Exercise

- Generate 100 distance scenarios with random data
- Solve these scenarios with the solveLink values...
  - aSyncGrid [3]
  - aSyncThreads [6]
- Compare the execution time of solving all scenarios with different solveLink settings
  - Hint: Check the log for output about solveLink
    - → Use solver CplexD instead of Cplex
  - Hint: Look at the following GAMS functions:
    - `readyCollect`
    - `handleCollect`
    - `handleDelete`

# SolveLink Option – Asynchronous Solves

```

Welcome x 6_transport_solveLink_seq.gms x 6_transport_solveLink_seq.lst x 7_transport_solveLink_async.gms x 7_transport_solveLink_async.lst x
64 Model transport /all/ ;
65
66 set s scenarios / s1*s100 /
67 sl solvelink / aSyncGrid, aSyncThreads /;
68 parameter dd(s,i,j) distance by scenario
69 time(*) time for 100 scenarios
70 sl_val(sl) solvelink value / aSyncGrid %solveLink.aSyncGrid%,
71 aSyncThreads %solveLink.aSyncThreads% /
72 scalar h(s) scenario handle;
73 scalar tmp;
74
75 dd(s,i,j) = uniform(0.9,1.1)*d(i,j);
76 option limrow=0, limcol=0, solprint=silent, lp=cplexd;
77 * Async SOLVE
78 loop(sl,
79 tmp = jnow;
80 transport.solvelink=sl_val(sl);
81
82 loop(s,
83 d(i,j) = dd(s,i,j);
84 Solve transport using lp minimizing z ;
85 h(s) = transport.handle; // save instance handle
86 );
87 repeat
88 display$readycollect(h) 'Waiting for next instance to complete';
89 loop(s$hndlecollect(h(s)),
90 display$hndledelete(h(s)) 'trouble deleting handles';
91 h(s) = 0; // indicate that we have loaded the solution
92 );
93 until card(h) = 0 or timeelapsed > 180; // wait until all models are loaded
94 time(sl) = (jnow-tmp)*24*60*60;
95 );
96 display time;
97|

```

ChainScript	6.710
CallModule	2.694
LoadLibrary	0.578
aSyncGrid	4.259
aSyncThreads	0.496

```

----          96 PARAMETER time    time for 100 scenarios
aSyncGrid      4.259,      aSyncThreads 0.496

```

# SolveLink Option – Asynchronous Solves

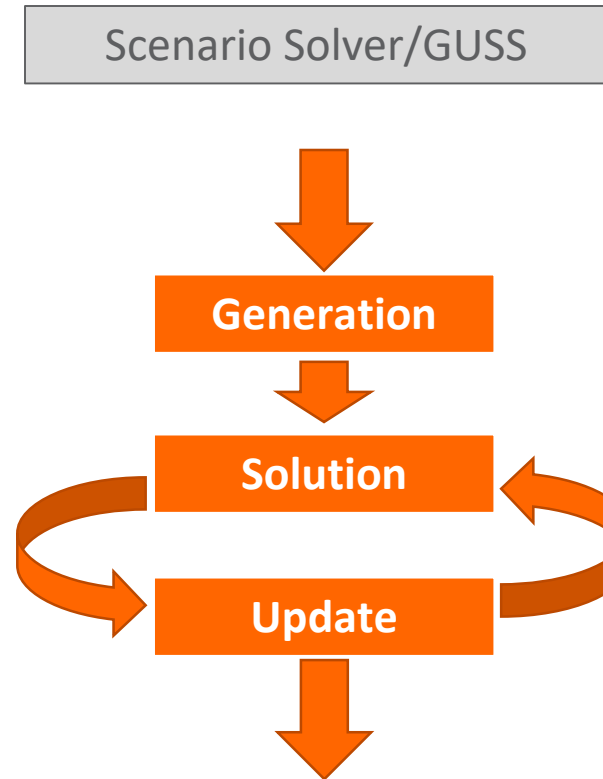
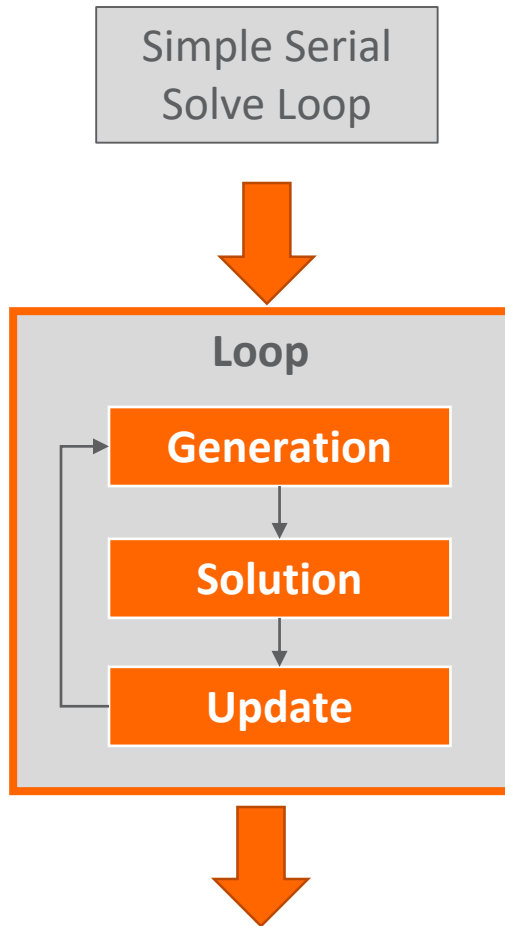
- Helpful, if large ratio of solver time / GAMS time

```
7_dicex_solveLink.gms 7_dicex_solveLink.lst
95
96 * SEQUENTIAL SOLVE
97 loop(seq(sl),
98     tmp = jnow;
99     dice2.solveLink=sl_val(sl);
100     loop(s,
101         solve dice2 using mip maximizing wnx;
102     );
103     time(sl) = (jnow-tmp)*24*60*60;
104 );
105
106 * Async SOLVE
107 loop(async(sl),
108     tmp = jnow;
109     dice2.solveLink=sl_val(sl);
110     loop(s,
111         solve dice2 using mip maximizing wnx;
112         h(s) = dice2.handle;    // save instance handle
113     );
114     repeat
115         display$readycollect(h) 'Waiting for next instance to complete';
116         loop(s$handlecollect(h(s)),
117             display$handledelete(h(s)) 'trouble deleting handles';
118             h(s) = 0;    // indicate that we have loaded the solution
119         );
120     until card(h) = 0 or timeelapsed > 180; // wait until all models are loaded
121     time(sl) = (jnow-tmp)*24*60*60;
122 );
123 option time:3:0:1;
124 display time;
```

ChainScript	29.807
CallModule	30.004
LoadLibrary	28.864
aSyncGrid	9.112
aSyncThreads	7.901

# GUSS – Gather-Update-Solve-Scatter

aka Scenario Solver



Generates model once and updates the algebraic model **keeping the model “hot”** inside the solver.



# GUSS – Gather-Update-Solve-Scatter

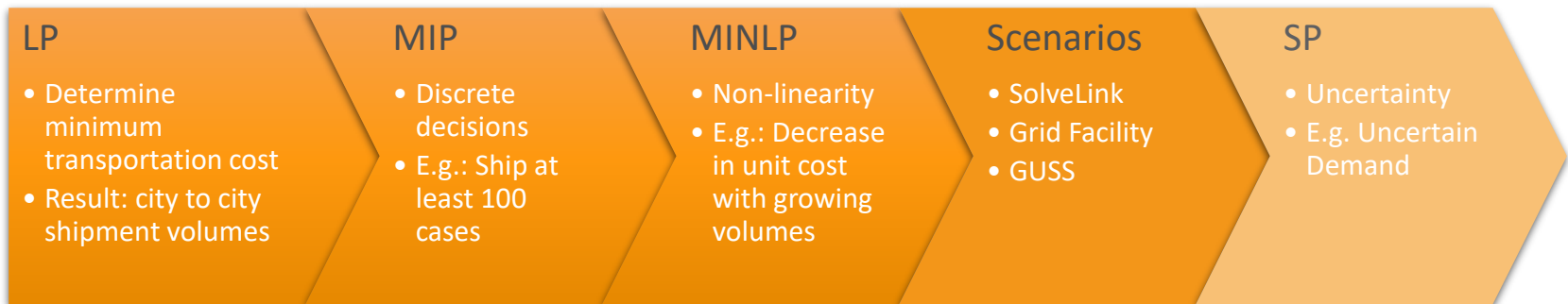
aka Scenario Solver

```
8_transport_GUSS_solvelink.gms* 8_transport_GUSS_solvelink.lst
69 parameter dd(s,i,j) distance by scenario
70           ff(s)    freight cost by scenario
71           time(*)   time for 100 scenarios;
72 scalar      tmp;
73
74 dd(s,i,j) = uniform(0.9,1.1)*d(i,j);
75 ff(s)      = uniform(0.9,1.1)*f;
76 option limrow=0, limcol=0, solprint=off;
77
78 * GUSS
79 transport.solvelink = 0;
80 tmp = jnow;
81 Set mattrib / system.GUSSModelAttributes /;
82 Parameter
83   xxGUSS(s,i,j) collector for level of x
84   srep(s, mattrib) model attributes like modelstat etc
85   o(*)           GUSS options / SkipBaseCase 1 /
86
87 Set dict / s . scenario.'
88           o . opt      .srep
89           d . param     .dd
90           f . param     .ff
91           x . level     .xxGUSS /
92
93 Solve transport using lp minimizing z scenario dict;
94 time('GUSS') = (jnow-tmp)*24*60*60;
95
96 display time;
97
```

ChainScript	6.710
CallModule	2.694
LoadLibrary	0.578
aSyncGrid	4.259
aSyncThreads	0.496
GUSS	0.273

## Grid & GUSS – Examples from the model library

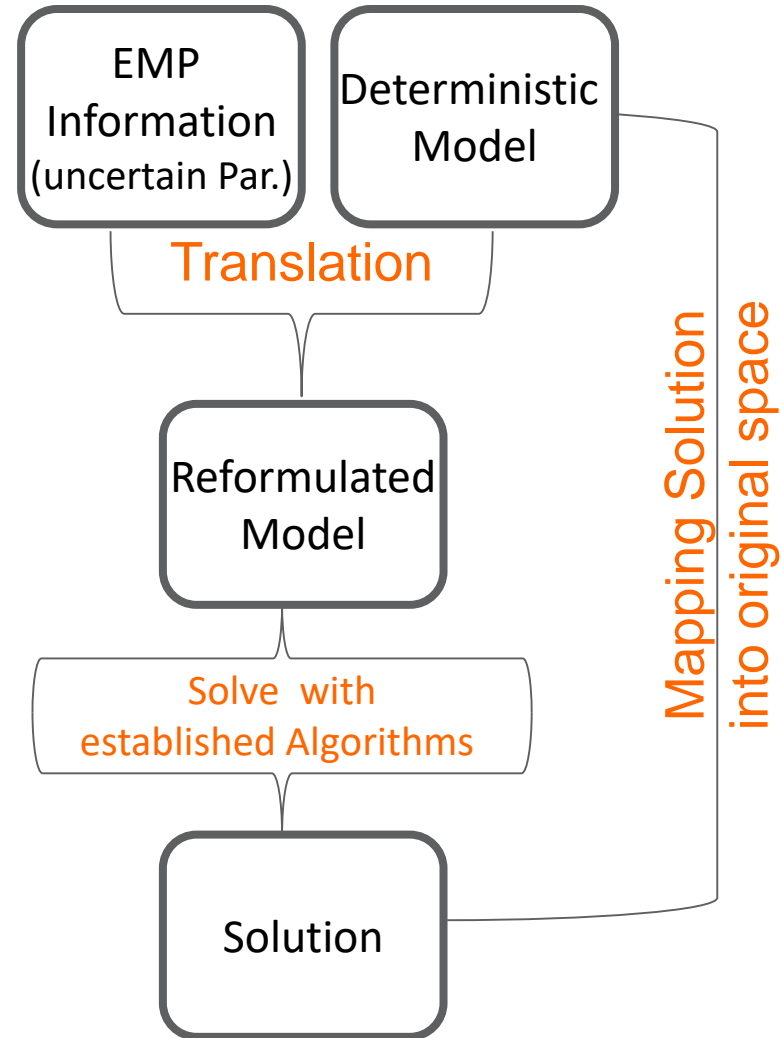
- **trnsgrid**: [https://www.gams.com/latest/gamslib\\_ml/libhtml/gamslib\\_trnsgrid.html](https://www.gams.com/latest/gamslib_ml/libhtml/gamslib_trnsgrid.html)
  - Simple asynchronous solves in a loop, separate collection loop
- **tgridmix**: [https://www.gams.com/latest/gamslib\\_ml/libhtml/gamslib\\_tgridmix.html](https://www.gams.com/latest/gamslib_ml/libhtml/gamslib_tgridmix.html)
  - Asynchronous solves in combined submission & collection loop. Keep number of submitted models  $\leq$  #threads
- **gussgrid**: [https://www.gams.com/latest/gamslib\\_ml/libhtml/gamslib\\_gussgrid.html](https://www.gams.com/latest/gamslib_ml/libhtml/gamslib_gussgrid.html)
  - Asynchronous GUSS-solves in combined submission & collection loop. Keep number of submitted models  $\leq$  #threads



# Stochastic Programming in GAMS

## EMP/SP

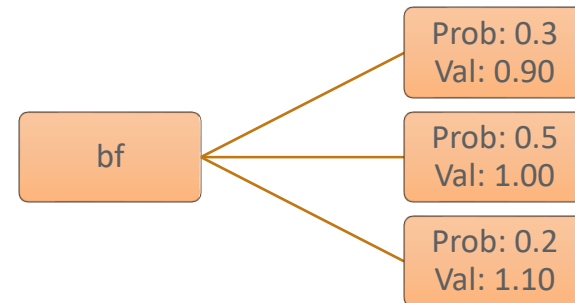
- Simple interface to add uncertainty to existing deterministic models
- (EMP) Keywords to describe uncertainty include: discrete and parametric random variables, stages, chance constraints, Value at Risk, ...
- Available solution methods:
  - Automatic generation of **D**eterministic **E**quivalent (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  $bf$

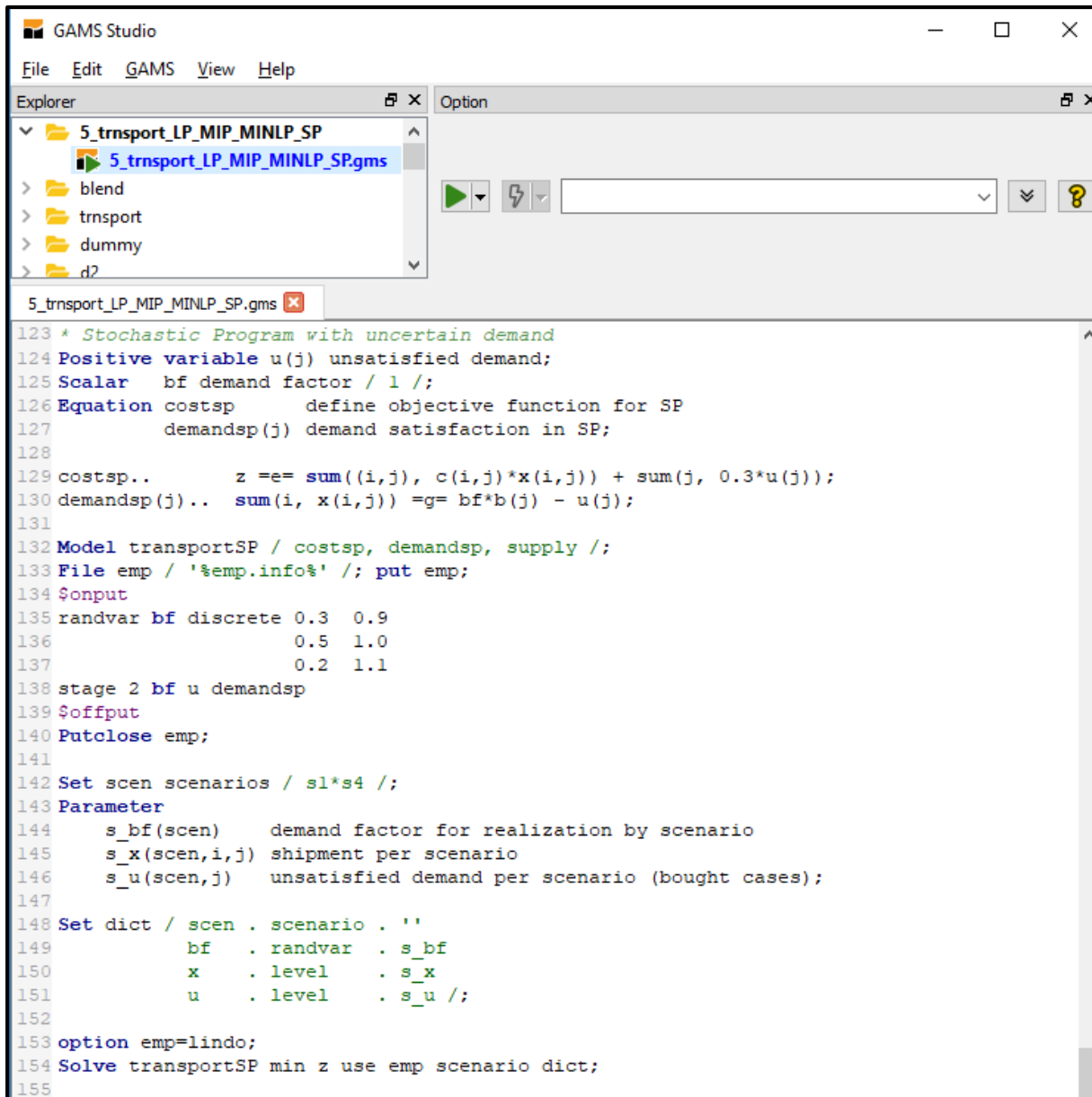


## Decisions to make

- 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



The screenshot shows the GAMS Studio application. The Explorer panel on the left lists a project named '5\_transport\_LP\_MIP\_MINLP\_SP' with a sub-file '5\_transport\_LP\_MIP\_MINLP\_SP.gms'. The Options panel on the right is empty. The main code editor displays the following GAMS code:

```
123 * Stochastic Program with uncertain demand
124 Positive variable u(j) unsatisfied demand;
125 Scalar bf demand factor / 1 /;
126 Equation costsp define objective function for SP
127 demandsp(j) demand satisfaction in SP;
128
129 costsp.. z =e= sum((i,j), c(i,j)*x(i,j)) + sum(j, 0.3*u(j));
130 demandsp(j).. sum(i, x(i,j)) =g= bf*b(j) - u(j);
131
132 Model transportSP / costsp, demandsp, supply /;
133 File emp / '%emp.info%' /; put emp;
134 $onput
135 randvar bf discrete 0.3 0.9
136 0.5 1.0
137 0.2 1.1
138 stage 2 bf u demandsp
139 $offput
140 Putclose emp;
141
142 Set scen scenarios / s1*s4 /;
143 Parameter
144 s_bf(scen) demand factor for realization by scenario
145 s_x(scen,i,j) shipment per scenario
146 s_u(scen,j) unsatisfied demand per scenario (bought cases);
147
148 Set dict / scen . scenario . ''
149 bf . randvar . s_bf
150 x . level . s_x
151 u . level . s_u /;
152
153 option emp=lindo;
154 Solve transportSP min z use emp scenario dict;
155
```

Hands-On

# Uncertain Demand: Results

GAMS Studio

File Edit GAMS View Help

Explorer

- 5\_transport\_LP\_MIP\_MINLP\_SP
  - 5\_transport\_LP\_MIP\_MINLP\_SP.gdx
  - 5\_transport\_LP\_MIP\_MINLP\_SP.lst
  - 5\_transport\_LP\_MIP\_MINLP\_SP.gms
- blend
- transport

Option

5\_transport\_LP\_MIP\_MINLP\_SP.gms 5\_transport\_LP\_MIP\_MINLP\_SP.lst 5\_transport\_LP\_MIP\_MINLP\_SP.gdx

Column

- x
- z
- ship
- Model Statistics SOLVE transportMIP ...
- Solution Report SOLVE transportMIP...
- Solve EQU
- Solve VAR
- Execution
- Display
- Equation Listing SOLVE transportMIN...
- Equation
- Column Listing SOLVE transportML...
- Column
- Model Statistics SOLVE transportMIN...
- Solution Report SOLVE transportMIN...
- Solve EQU
- Solve VAR
- Execution
- Display
- Equation Listing SOLVE transportSP ...
- Equation
- Column Listing SOLVE transportSP ...
- Column
- Model Statistics SOLVE transportSP U...
- Solution Report SOLVE transportSP ...
- Solve EQU
- Solve VAR
- Execution
- Display
- s\_bf

1031 ---- 156 PARAMETER s\_bf demand factor for realization by scenario

1032

1033 s1 0.900, s2 1.000, s3 1.100

1034

1035

1036 ---- 156 PARAMETER s\_b demand per scenario

1037

1038 new-york chicago topeka

1039

1040 s1 292.500 270.000 247.500

1041 s2 325.000 300.000 275.000

1042 s3 357.500 330.000 302.500

1043

1044

1045 ---- 156 PARAMETER s\_x shipment per scenario

1046

1047 new-york chicago topeka

1048

1049 s1.seattle 50.000 300.000

1050 s1.san-diego 242.500 275.000

1051 s2.seattle 50.000 300.000

1052 s2.san-diego 242.500 275.000

1053 s3.seattle 50.000 300.000

1054 s3.san-diego 242.500 275.000

1055

1056

1057 ---- 156 PARAMETER s\_u unsatisfied demand per scenario (bought cases)

1058

1059 new-york chicago topeka

1060

1061 s2 32.500

1062 s3 65.000 30.000 27.500

C:\WC\source\share\Lutz\Presentations\2018\_OR\_Brussels\Workshop\examples\5\_transport\_LP\_MIP\_MINLP\_SP.lst 1086 lines 1031 / 17 RO UTF-8

# Stochastic Programming in GAMS

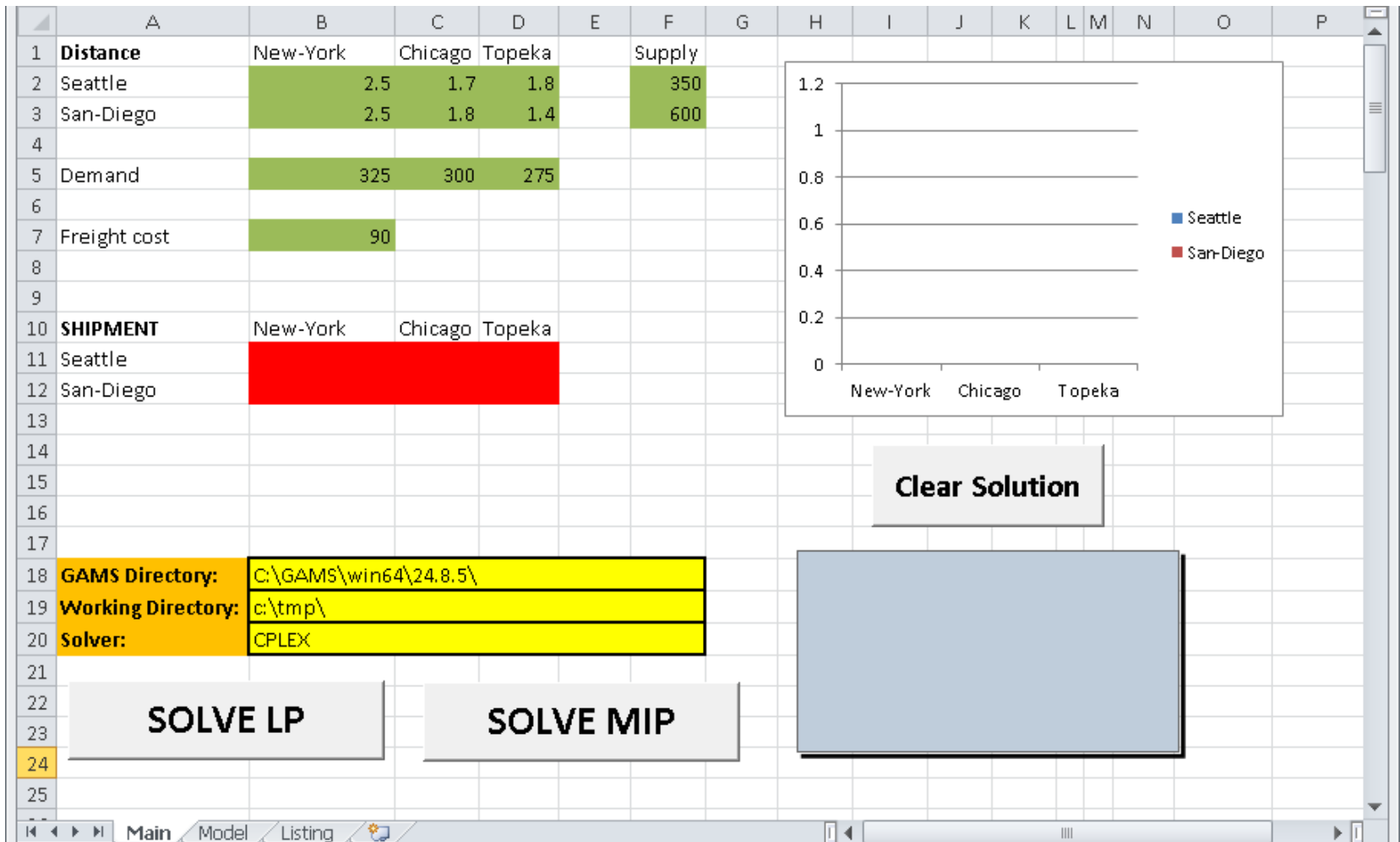
- The Extended Mathematical Programming (EMP) framework is used to replace parameters in the model by random variables
- Support for Multi-stage recourse problems and chance constraint models
- Easy to add uncertainty to existing deterministic models, to either use specialized algorithms or create Deterministic Equivalent (new free solver DE)
- More information:  
[https://www.gams.com/latest/docs/UG\\_EMP\\_SP.html](https://www.gams.com/latest/docs/UG_EMP_SP.html)



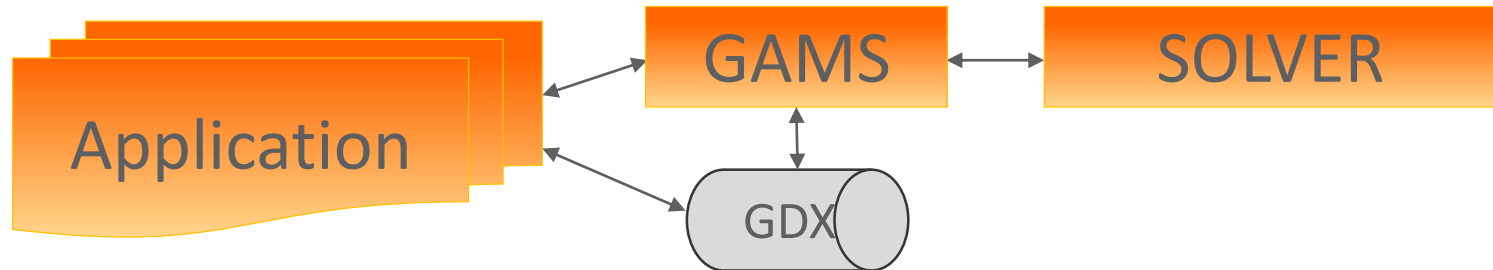
# Outlook: Deployment of GAMS Models

- APIs – Application Programming Interfaces to GAMS
  - Low Level APIs
  - Object Oriented APIs
- Using R/Shiny to deploy and visualize GAMS models in a Web Interface

# Excel and GAMS



# Calling GAMS from your Application



## Creating Input for GAMS Model

→ Data handling using **GDX** API

## Callout to GAMS

→ GAMS option settings using **Option** API

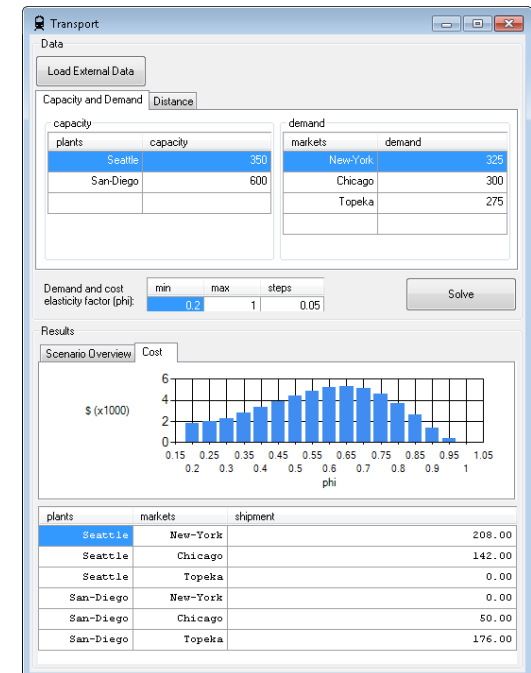
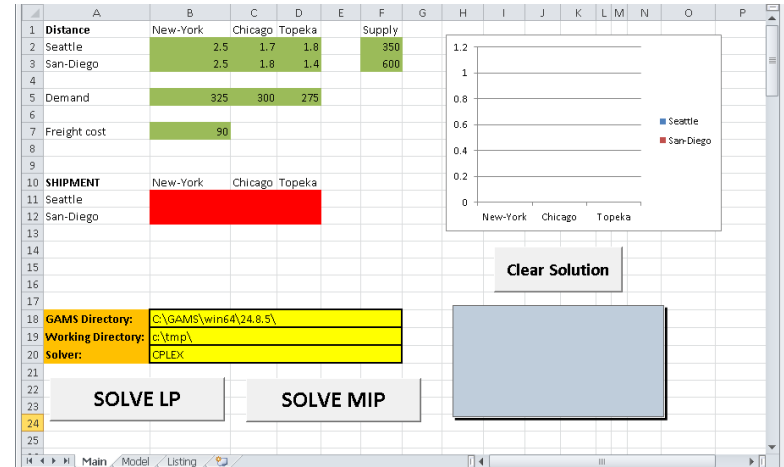
→ Starting GAMS using **GAMS** API

## Reading Solution from GAMS Model

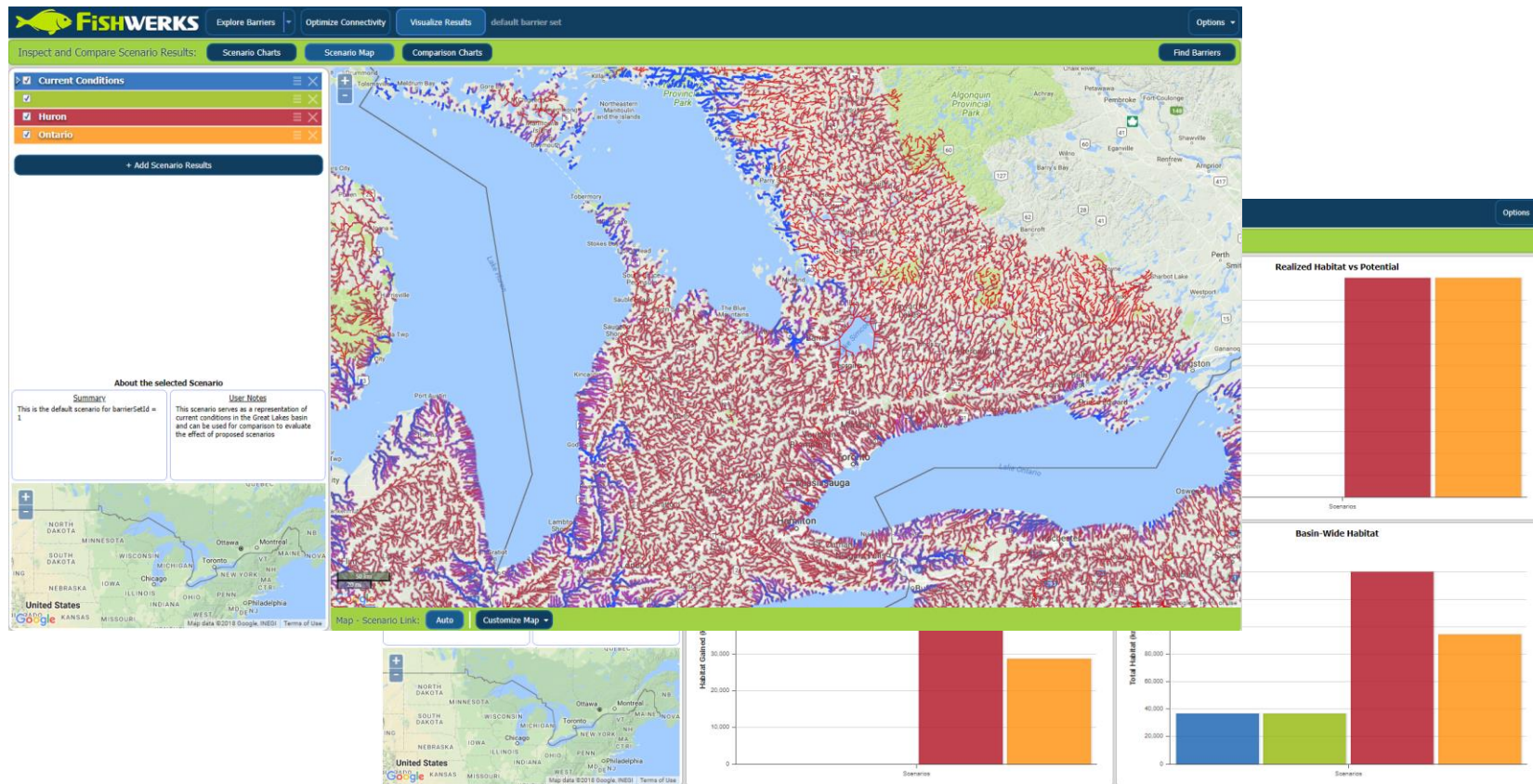
→ Data handling using **GDX** API

# Low level APIs → Object Oriented API

- Low level APIs
  - GDX, OPT, GAMSX, GMO, ...
  - High performance and flexibility
  - Automatically generated imperative APIs for several languages:  
C, C++, C#, Delphi, Java, Python, VBA, ...
- Object Oriented GAMS API
  - Additional layer on top of the low level APIs
  - Object Oriented
  - Written by hand to meet the specific requirements of different Object Oriented languages



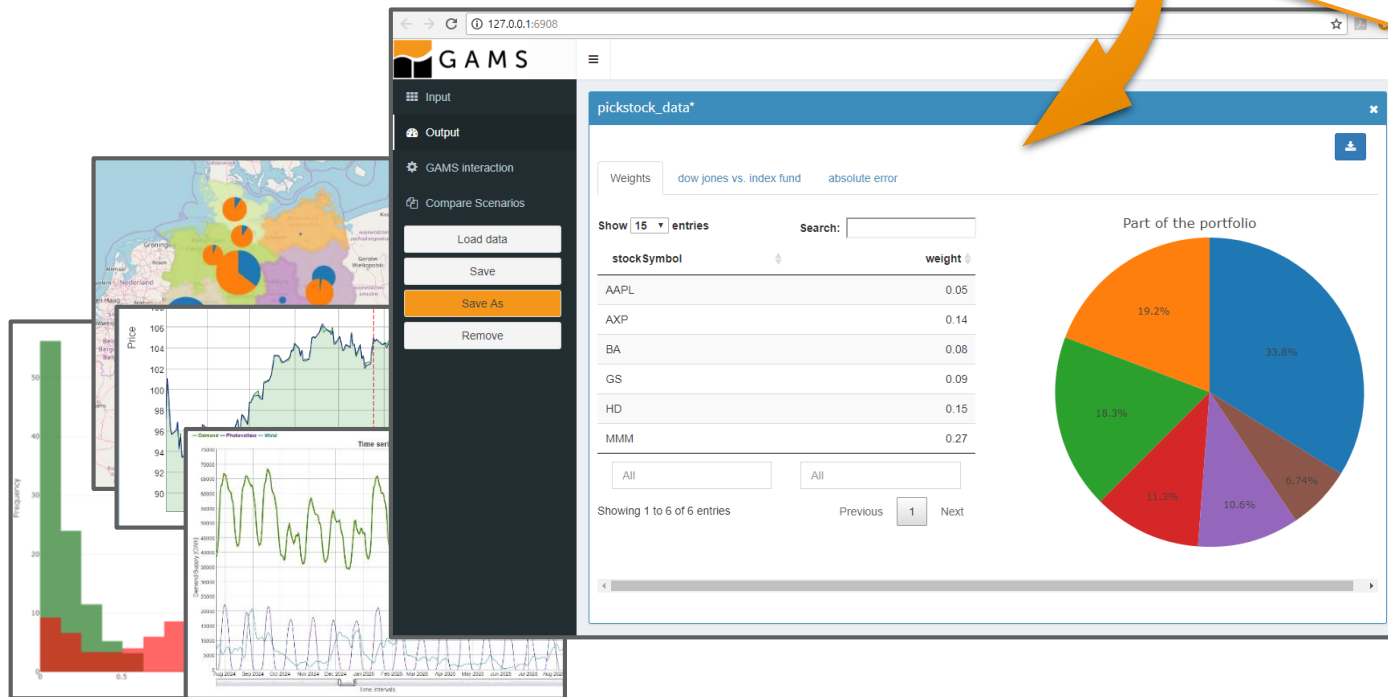
- GAMS comes with several OO APIs (Python, Java, C++, C#, ...) to develop applications
  - Programming required to build your applications



# From GAMS Model to Visual Web User Interface

```
54 Equation
55   deffit(date)      'fit to Dow Jones index'
56   defpick(symbol)   'can only use stock if picked'
57   defnumstock       'few stocks allowed'
58   defobj            'absolute violation (L1 norm) from index';
59
60 deffit(ds)..  sum(s, price(ds,s)*w(s)) =e= index(ds) + slpos(ds) - slneg(ds);
61
62 defpick(s)..  w(s) =l= p(s);
63
64 defnumstock.. sum(s, p(s)) =l= maxstock;
65
66 defobj..      obj =e= sum(ds, slpos(ds) + slneg(ds));
```

Currently under  
Development



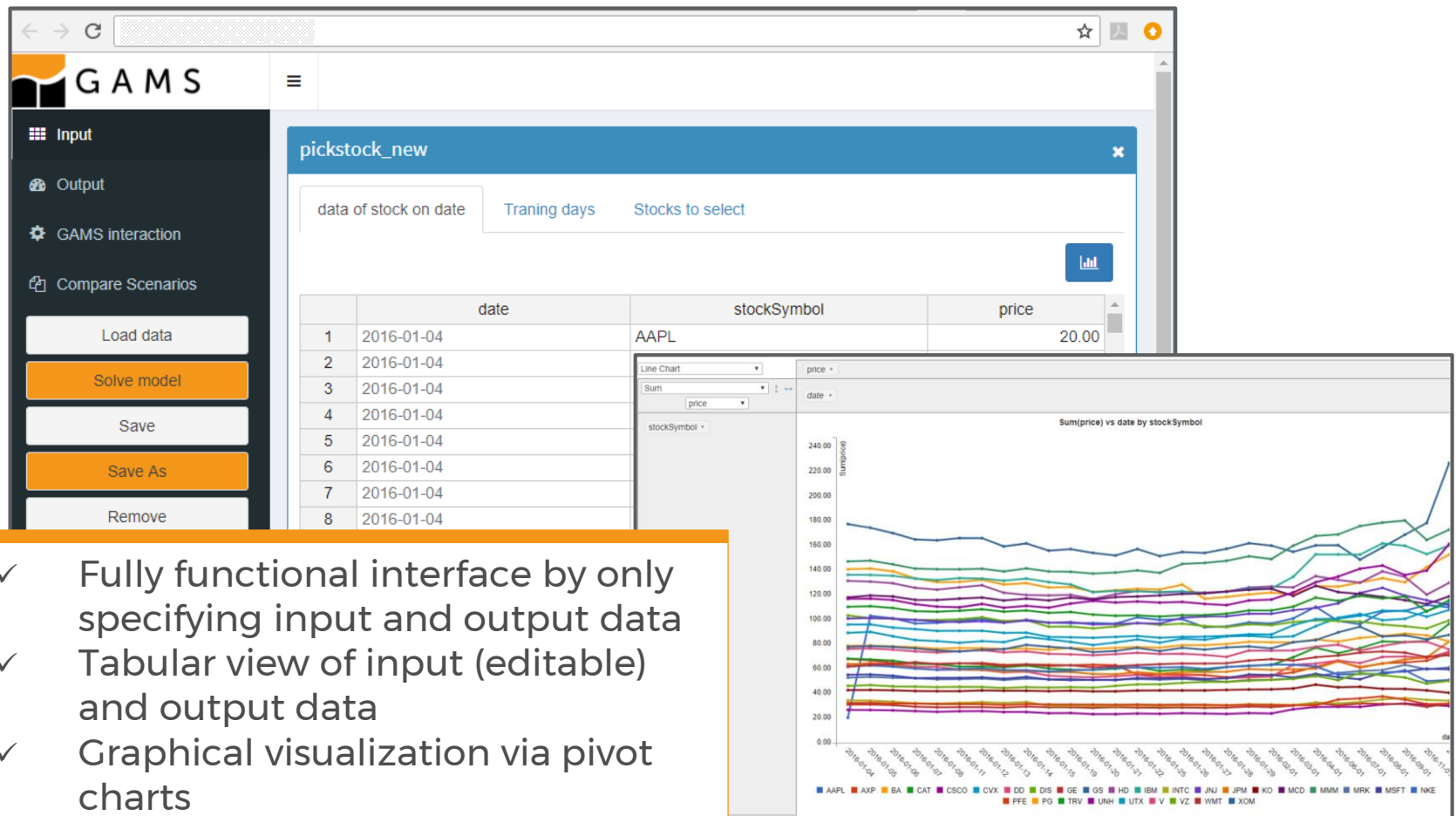
# From GAMS Model to Visual Web User Interface

1

## Basic setup:

- ✓ Annotating GAMS model (defining the input and output data to be displayed in the WebUI)

# Basic Setup – GAMS Model Annotations





# Basic Setup – GAMS Model Annotations

```
7 $onExternalInput
8 Set date      'date'
9   symbol 'stockSymbol';
10
11 Parameter
12   stockData(date,symbol,hrd)  'data of stock on date ### { "headers":{"date":{"readonly":true}} }';
13
14 Scalar
15   maxstock      'maximum number of stocks to select ### { "slider":{"min":1, "max":"card(stockdata$
16   trainingdays  'number of days for training          ### { "slider":{"min":1, "max":"card(stockdata$
17 $offExternalInput
18
19
20
21
22
23
24
25
26
27
28 $onExternalOutput
29 Set wHdr      'w header'           / 'weight' /
30   fHdr      'fund header'         / 'dow jones','index fund' /
31   errHdr     'stock symbol header' / 'absolute error train', 'absolute error test' /;
32 Parameter
33   partOfPortfolio(symbol,wHdr)      'what part of the portfolio'
34   dowVSIndex(date,fHdr)             'dow jones vs. index fund'
35   abserror(date,errHdr)             'absolute error'
36 Singleton Set lastDayTraining(date) 'last date of training period ### vertical marker in chart' ;
37 $offExternalOutput
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106 $if not exist webui.gms
107 $if set GSMWEBUI $abort Asked to do webui but can't find webui.gms. Set idir=path/to/webui
108 $batinclude webui
```

# Basic Setup – GAMS Model Annotations

## 7 \$onExternalInput

```
8 Set date 'date'
9     symbol 'stockSymbol';
10
11 Parameter
12     stockData(date,symbol,hrd) 'data of stock on date ### { "headers":{"date":{"readonly":true}} }';
13
14 Scalar
15     maxstock 'maximum number of stocks to select ### { "slider":{"min":1, "max":"card(stockdata$
16     numDays 'number of days for training ### { "slider":{"min":1, "max":"card(stockdata$
```

## 17 \$offExternalInput

## 88 \$onExternalOutput

```
89 Set whdr 'header' / 'weight' /
90     fHdr 'fund header' / 'dow jones','index fund' /
91     errHdr 'stock symbol header' / 'absolute error train', 'absolute error test' /;
92 Parameter
93     partOfPortfolio(symbol,whdr) 'what part of the portfolio'
94     dowVSIndex(date,fHdr) 'dow jones vs. index fund'
95     abserror(date,errHdr) 'absolute error'
96 Singleton Set lastDayTraining(date) 'last date of training period ### vertical marker in chart' ;
```

## 97 \$offExternalOutput

```
106 $if not exist webui.gms
107 $if set GMSWEBUI $abort Asked to do webui but can't find webui.gms. Set idir=path/to/webui
108 $batinclude webui
```

# Basic Setup – GAMS Model Annotations

```
7 $onExternalInput
8 Set date      'date'
9   symbol 'stockSymbol';
10
11 Parameter
12   stockData(date,symbol,hrd)  'data of stock on date ### { "headers":{"date":{"readonly":true}} }';
13
14 Scalar
15   maxstock      'maximum number of stocks to select ### { "slider":{"min":1, "max":"card(stockdata$
16   trainingdays  'number of days for training          ### { "slider":{"min":1, "max":"card(stockdata$
17 $offExternalInput
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88 $onExternalOutput
89 Set wHdr      'w header'          / 'weight' /
90   fHdr        'fund header'       / 'dow jones','index fund' /
91   errHdr      'stock symbol header' / 'absolute error train', 'absolute error test' /;
92 Parameter
93   partOfPortfolio(symbol,wHdr)    'what part of the portfolio'
94   dowVSIndex(date,fHdr)           'dow jones vs. index fund'
95   abserror(date,errHdr)           'absolute error'
96 Singleton Set lastDayTraining(date) 'last date of training period ### vertical marker in chart' ;
97 $offExternalOutput
98
99
100
101
102
103
104
105
106 $if not exist webui.gms
107 $if not exist webui not Asked to do webui but can't find webui.gms. Set idir=path/to/webui
108 $batinclude webui
```

# From GAMS Model to Visual Web User Interface

1

## Basic setup:

- ✓ Annotating GAMS model (defining the input and output data to be displayed in the WebUI)



2

## Advanced setup:

- ✓ Configuration of standard graphics and UI
- ✓ Sophisticated (custom) graphics (R API)
- ✓ Scenario management with internal database

# Advanced Setup – Configuration

```
"dowvsindex" : {  
  "outType" : "graph",  
  "graph" : {  
    "tool" : "dygraph",  
    "title" : "Dow Jones vs. Index Fund",  
    "xdata" : "date",  
    "ydata" : {  
      "dow jones" : {  
        "label" : "Dow Jones index",  
        "fillGraph" : true,  
      },  
      "index fund" : {  
        "label" : "Index fund",  
      },  
    },  
  },  
}
```

Introduction  
Configuration Generator  
Custom Renderers  
Language Files

## Configuration Generator

In order for you to be able to customize your WebUI to its full potential, we have developed this Configuration Manager. Here you can customize your WebUI and graphics. At the end you can download the JSON-file that needs to be placed in the folder <modelid>recozy/conf.

### General settings

Basic Information

### Graphs

Configure graphs

### Tables

Customize the table layout

Format in which input or output data is to be displayed in the WebUI.

datatable

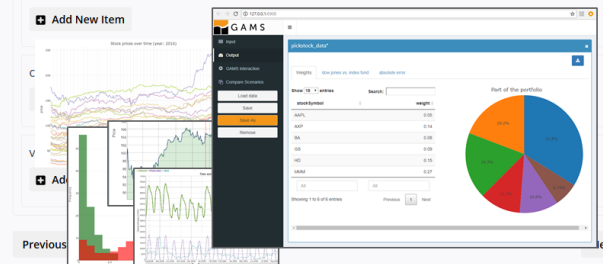
[Add New Item](#)

pivottable

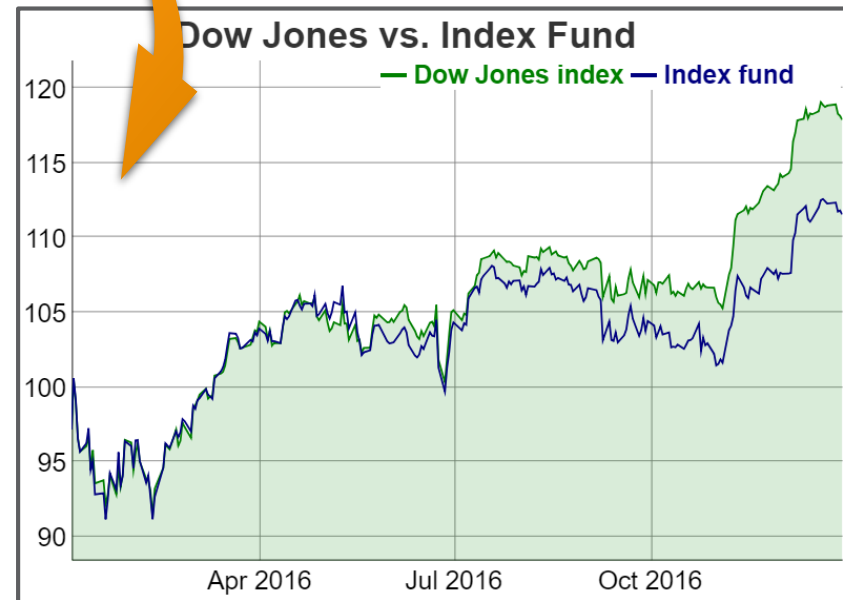
[Add New Item](#)

graph

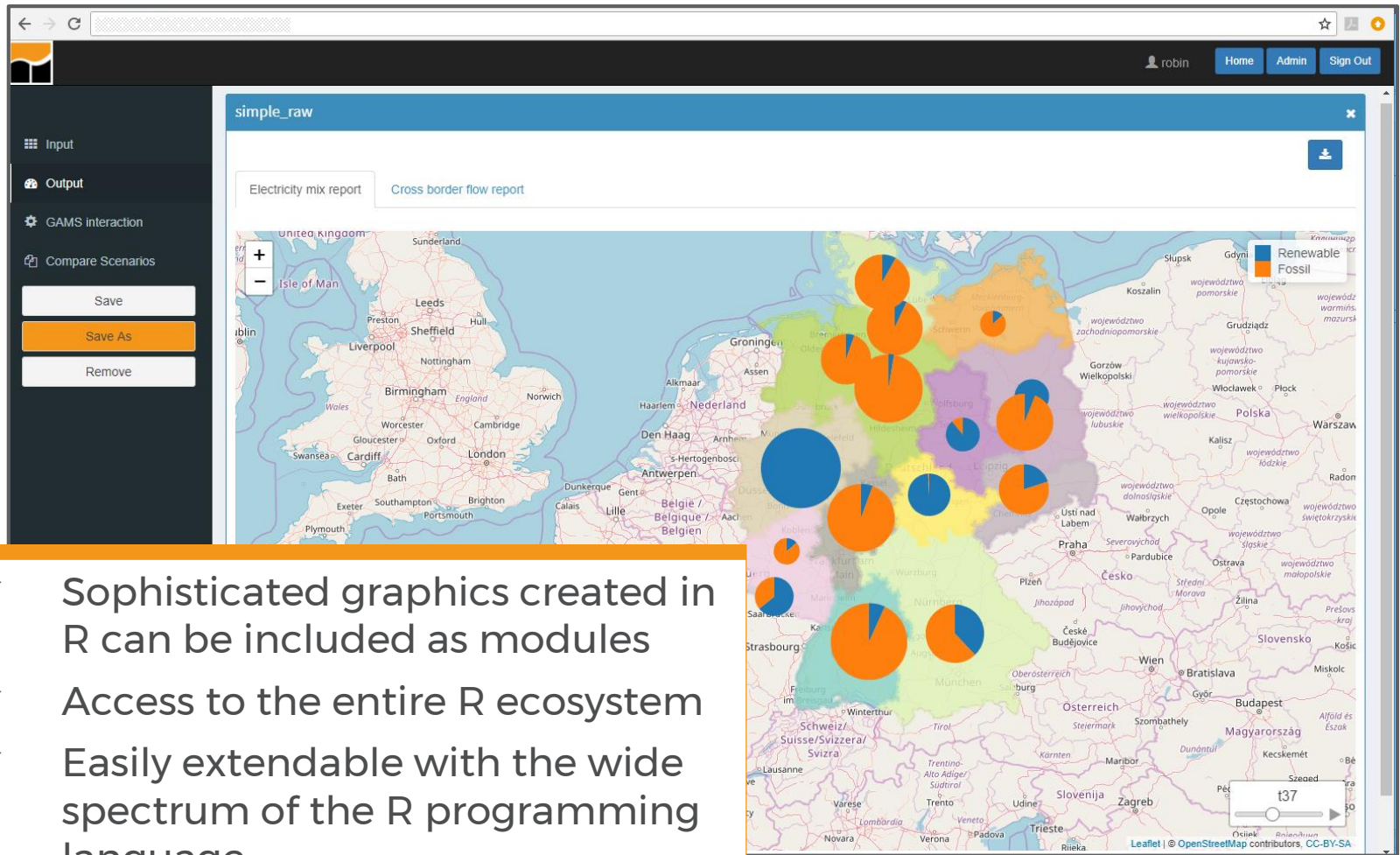
[Add New Item](#)



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



# 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

# From GAMS Model to Visual Web User Interface

1

## Initialization:

- ✓ Annotating GAMS model (defining the input and output data to be displayed in the WebUI)



2

## Advanced setup:

- ✓ Configuration of standard graphics and UI
- ✓ Sophisticated (custom) graphics (R API)
- ✓ Scenario management with internal database



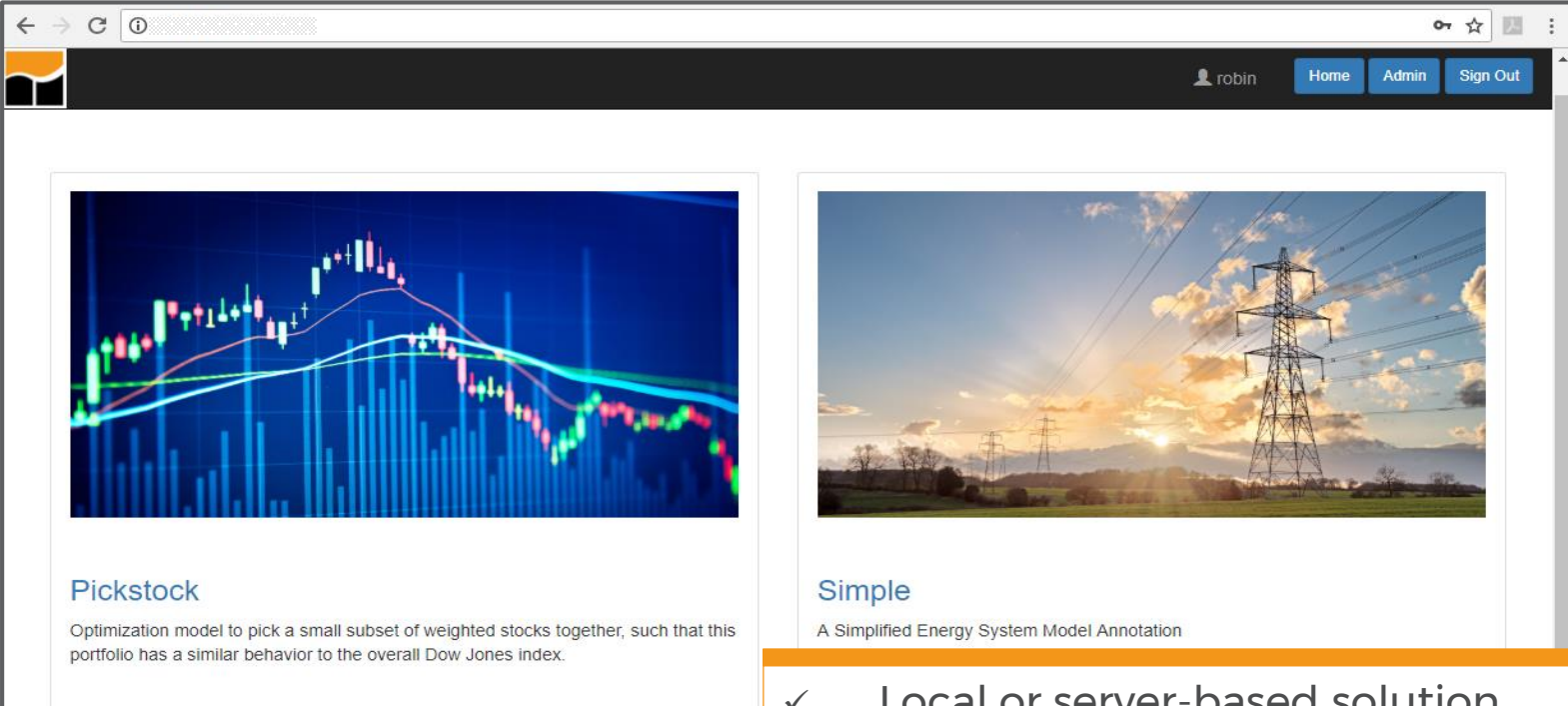
3

## Enterprise setup:

- ✓ User- and Application management



# Enterprise Setup – User and Application Management



The screenshot shows a web application interface with a dark header bar. On the right side of the header, there is a user profile icon labeled 'robin' and three buttons: 'Home', 'Admin', and 'Sign Out'. The main content area displays two application cards. The first card, titled 'Pickstock', features a candlestick chart with blue and red bars and a green moving average line. The second card, titled 'Simple', features a landscape image of a power line tower at sunset. Below the 'Simple' card, there is a text description: 'A Simplified Energy System Model Annotation'.

**Yaml config file**

```
- name: simple
  display-name: Simple
  logo-url: file://localhost/home/ec2-user/simple.png
  description: A Simplified Energy System Model Annotation
  docker-cmd: ["R", "-e", "shiny::runApp('/root/GMSWebUI')"]
  docker-image: gams/app
  docker-network: "my-network"
  docker-env:
    SHINYPROXY_MODELNAME: simple
  groups: GAMS_team, guest
```

- ✓ Local or server-based solution
- ✓ User authentication (e.g. LDAP, OAuth 2.0, Google, GitHub, Facebook)
- ✓ Multi-Application support with docker-based technology



# Enhanced Model Deployment in GAMS using R/Shiny

- 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 (configuration vs. programming)
- This “product” is currently **under development**. If you are interested in getting involved, please contact [support@gams.com](mailto:support@gams.com)

# Thank You