From Model to App

Develop and Deploy your GAMS Models

Robin Schuchmann
GAMS Software GmbH

Software used in this talk:
GAMS: www.gams.com/download/
GAMS MIRO Desktop: www.gams.com/miro/download.html

GAMS at a Glance

Company

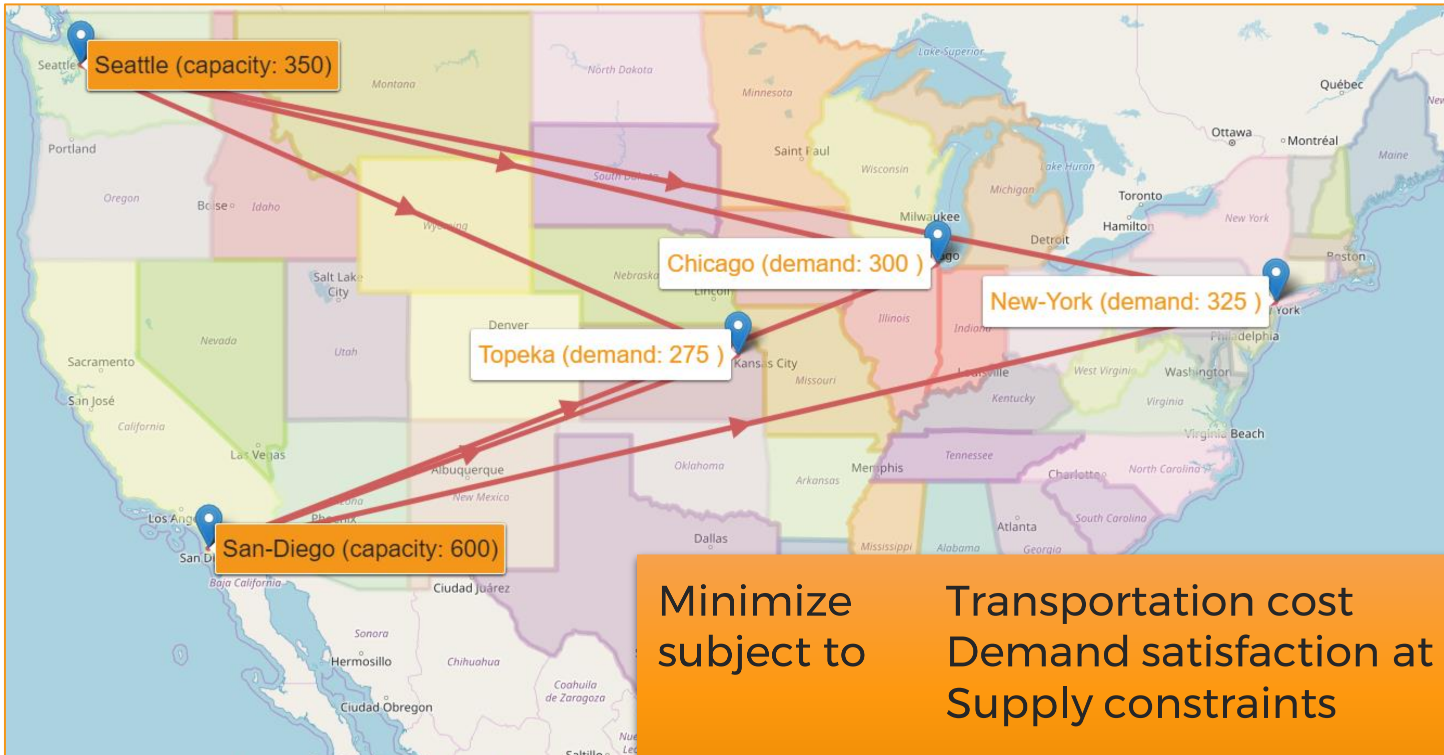
- Roots: World Bank, 1976
- Went commercial in 1987
- Locations
  - GAMS Development Corporation (USA)
  - GAMS Software GmbH (Germany)
- Product
  - The **G**eneral **A**lgebraic **M**odeling **S**ystem

Modeling Basics

Developing a GAMS Model

A Simple Transportation Problem

Canning Plants (supply) → shipments (Number of cases) → Markets (demand)



Minimize	Transportation cost
subject to	Demand satisfaction at markets
	Supply constraints

Freight: $90 case / thousand miles

Modeling Basics

Indices:

i = plants
j = markets

Given Data:

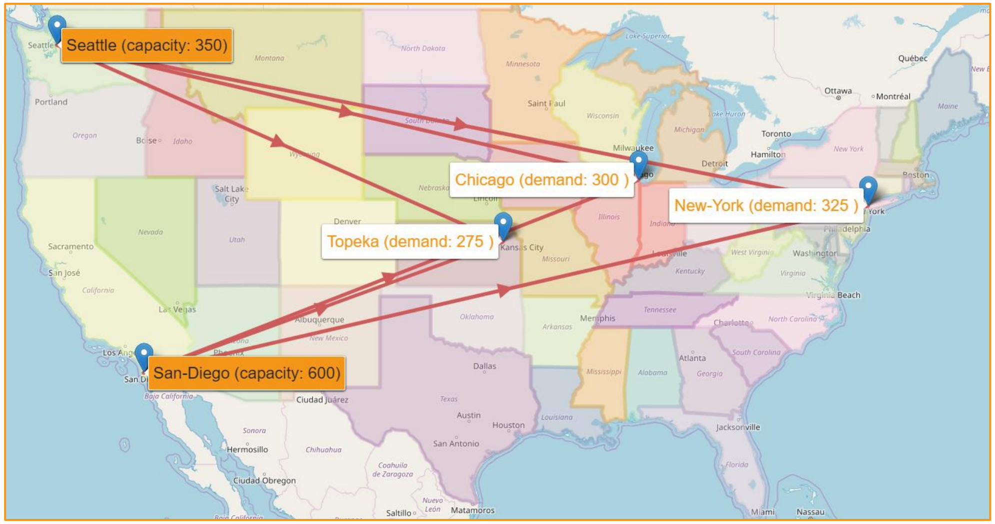
a_i = supply of commodity of plant i (in cases)
b_j = demand for commodity at market j
c_{ij} = cost per unit shipment between plant i and market j

Decision Variables:

x_{ij} = amount of commodity to ship from plant i to market j
where $x_{ij} \geq 0$, for all i, j

Constraints:

Observe supply limit at plant i: $\sum_j x_{ij} \leq a_i$ for all i (cases)
Satisfy demand at market j: $\sum_i x_{ij} \geq b_j$ for all j (cases)
Objective Function: Minimize $\sum_i \sum_j c_{ij} x_{ij}$ ($K)



Modeling Basics

Indices:

i = plants
j = markets

Given Data:

a_i = supply of commodity of plant i (in cases)
b_j = demand for commodity at market j
c_{ij} = cost per unit shipment between plant i and market j

Decision Variables:

x_{ij} = amount of commodity to ship from plant i to market j
where $x_{ij} \geq 0$, for all i, j

Constraints:

Observe supply limit at plant i: $\sum_j x_{ij} \leq a_i$ for all i (cases)
Satisfy demand at market j: $\sum_i x_{ij} \geq b_j$ for all j (cases)
Objective Function: Minimize $\sum_i \sum_j c_{ij} x_{ij}$ ($K)

```
Set
   i canning plants / seattle,  san-diego /
   j markets        / new-york, chicago, topeka /;

Parameter
   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 ;

Variable
   x(i,j) shipment quantities in cases
   z      total transportation costs in thousands of dollars ;

Equation
   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;
```

Modeling Basics

Indices:

i = plants
j = markets

Given Data:

a_i = supply of commodity of plant i (in cases)
b_j = demand for commodity at market j
c_{ij} = cost per unit shipment between plant i and market j

Decision Variables:

x_{ij} = amount of commodity to ship from plant i to market j
where $x_{ij} \geq 0$, for all i, j

Constraints:

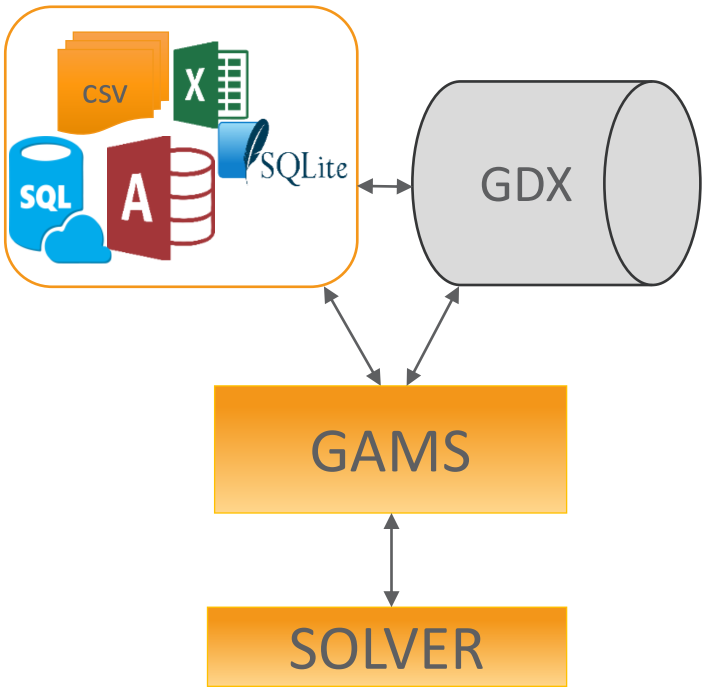
Observe supply limit at plant i: $\sum_j x_{ij} \leq a_i$ for all i (cases)
Satisfy demand at market j: $\sum_i x_{ij} \geq b_j$ for all j (cases)
Objective Function: Minimize $\sum_i \sum_j c_{ij} x_{ij}$ ($K)

```
Set
    i canning plants / seattle,  san-diego /
    j markets        / new-york, chicago, topeka /;

Parameter
    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 ;

Variable
    x(i,j) shipment quantities in cases
    z      total transportation costs in thousands of dollars ;

Equation
    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 /;
Option LP = CBC;
solve transport using lp minimizing z;
```

www.gams.com/products/solvers/

Independence of
- ✓ Model and Operating System
- ✓ Model and Solver

Independence of Model and Data




```
Set
    i canning plants / seattle,  san-diego /
    j markets        / new-york, chicago, topeka /;

Parameter
    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 ;

Variable
    x(i,j) shipment quantities in cases
    z      total transportation costs in thousands of dollars ;

Equation
    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;
```

Independence of

- ✓ Model and Operating System
- ✓ Model and Solver
- ✓ Model and Data

Independence of Model and User Interface

APIs

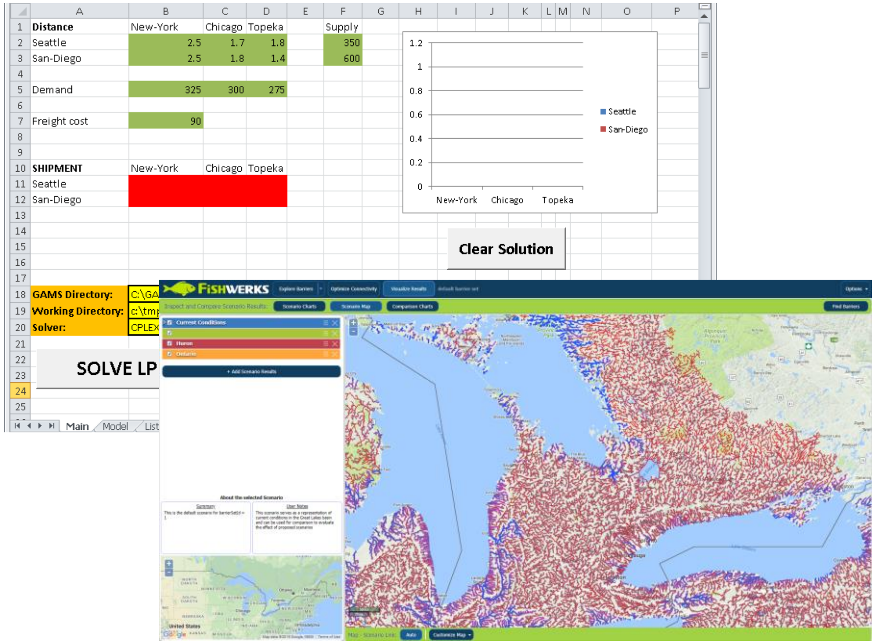
- Expert level APIs: high performance and flexibility
- Object Oriented APIs (Python, Java, C++, …) to develop applications

➞ Programming required to build applications

GAMS MIRO

- Deployment environment for GAMS models

➞ Configuration instead of programming

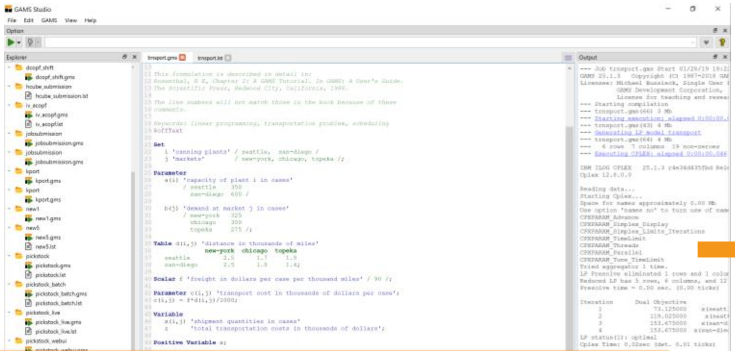


Independence of
- ✓ Model and Operating System
- ✓ Model and Solver
- ✓ Model and Data
- ✓ Model and User Interface

What is GAMS MIRO?

GAMS MIRO

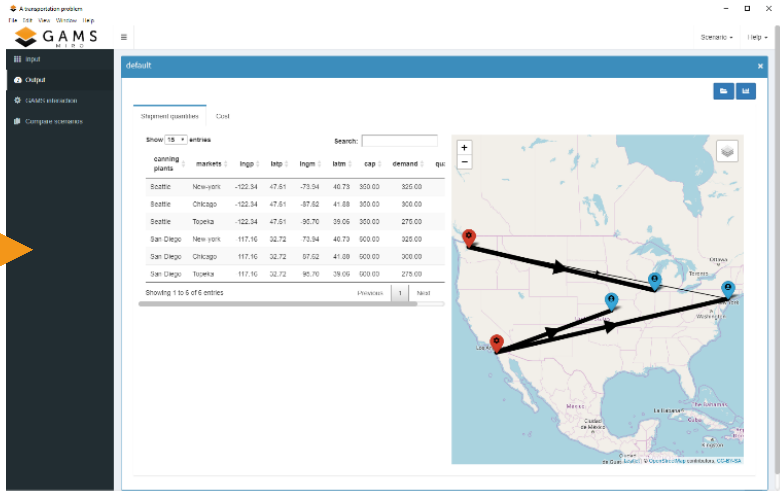
Model **I**nterface with **R**apid **O**rchestration



Observe supply limit at plant i: $\sum_j x_{ij} \leq a_i$ for all i (cases)

Satisfy demand at market j: $\sum_i x_{ij} \geq b_j$ for all j (cases)

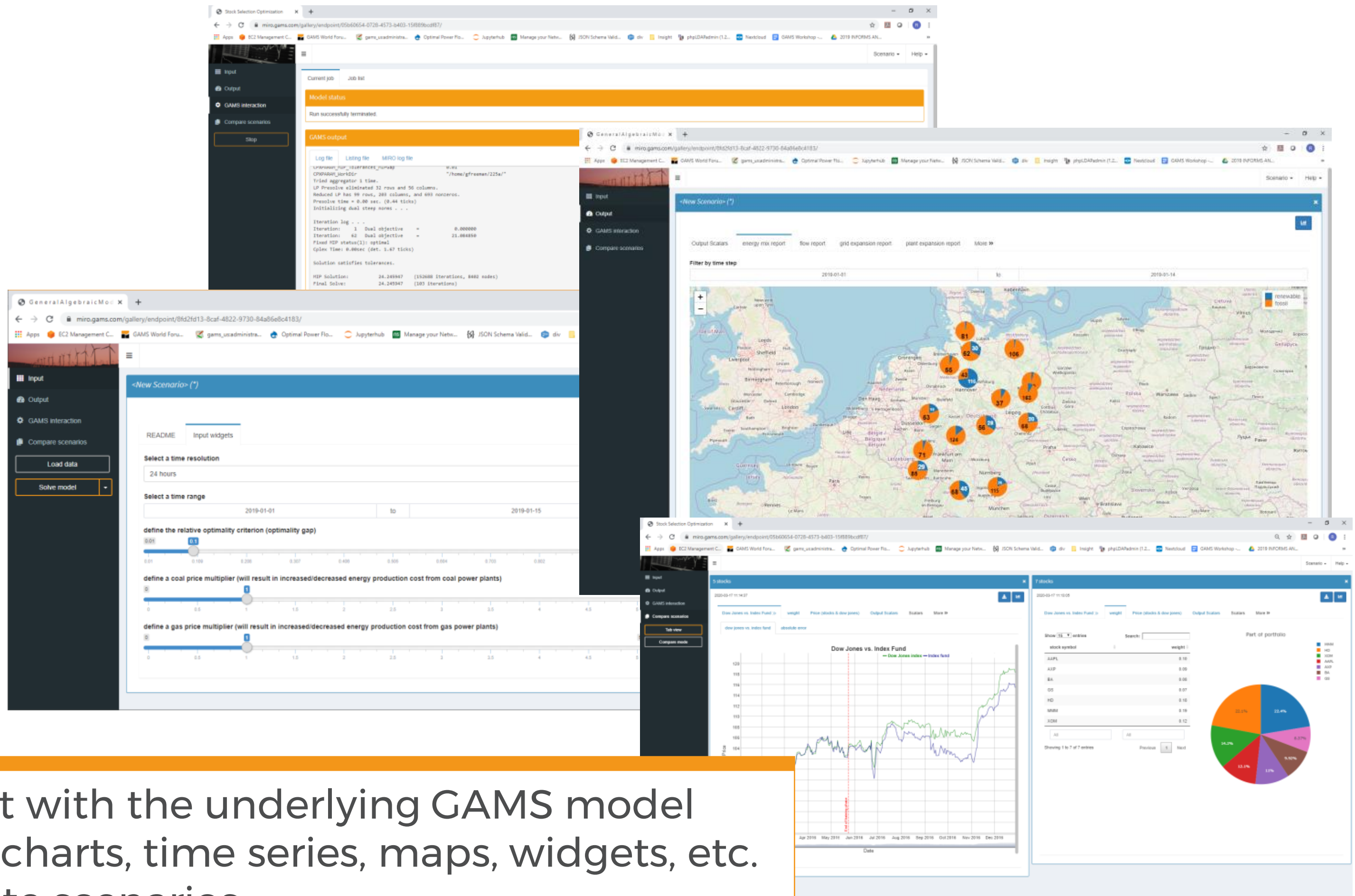
Objective Function: Minimize $\sum_i \sum_j c_{ij} x_{ij}$ ($K)



- ✓ Interactive interface for GAMS models
- ✓ Usage via web browser
- ✓ GAMS as a black box
- ✓ Automatic deployment

GAMS MIRO

Model **I**nterface with **R**apid **O**rchestration



- ✓ Interact with the underlying GAMS model
- ✓ Create charts, time series, maps, widgets, etc.
- ✓ Generate scenarios
- ✓ Compare results

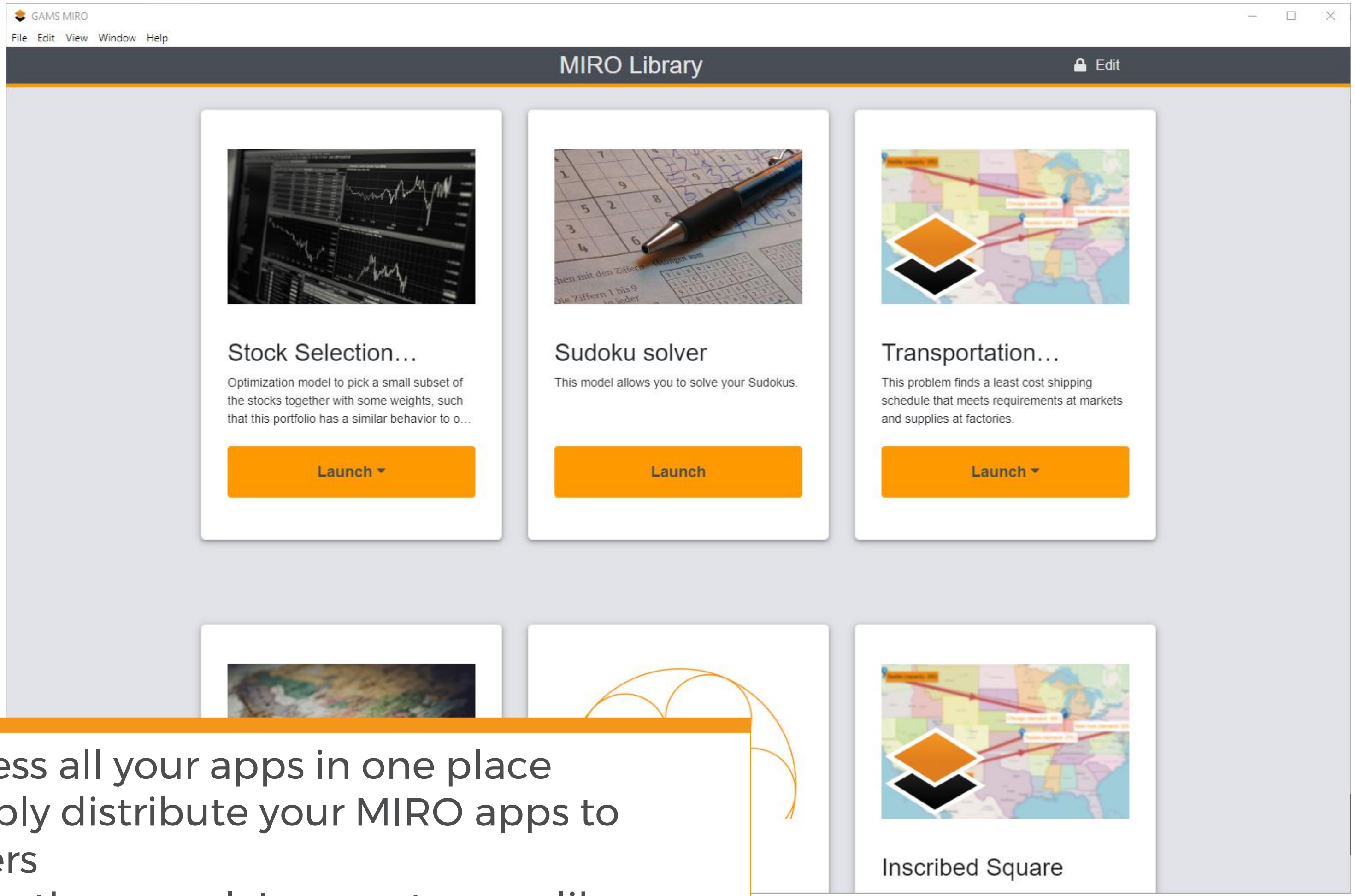
From model to app

Hands-on

Software used in this talk:

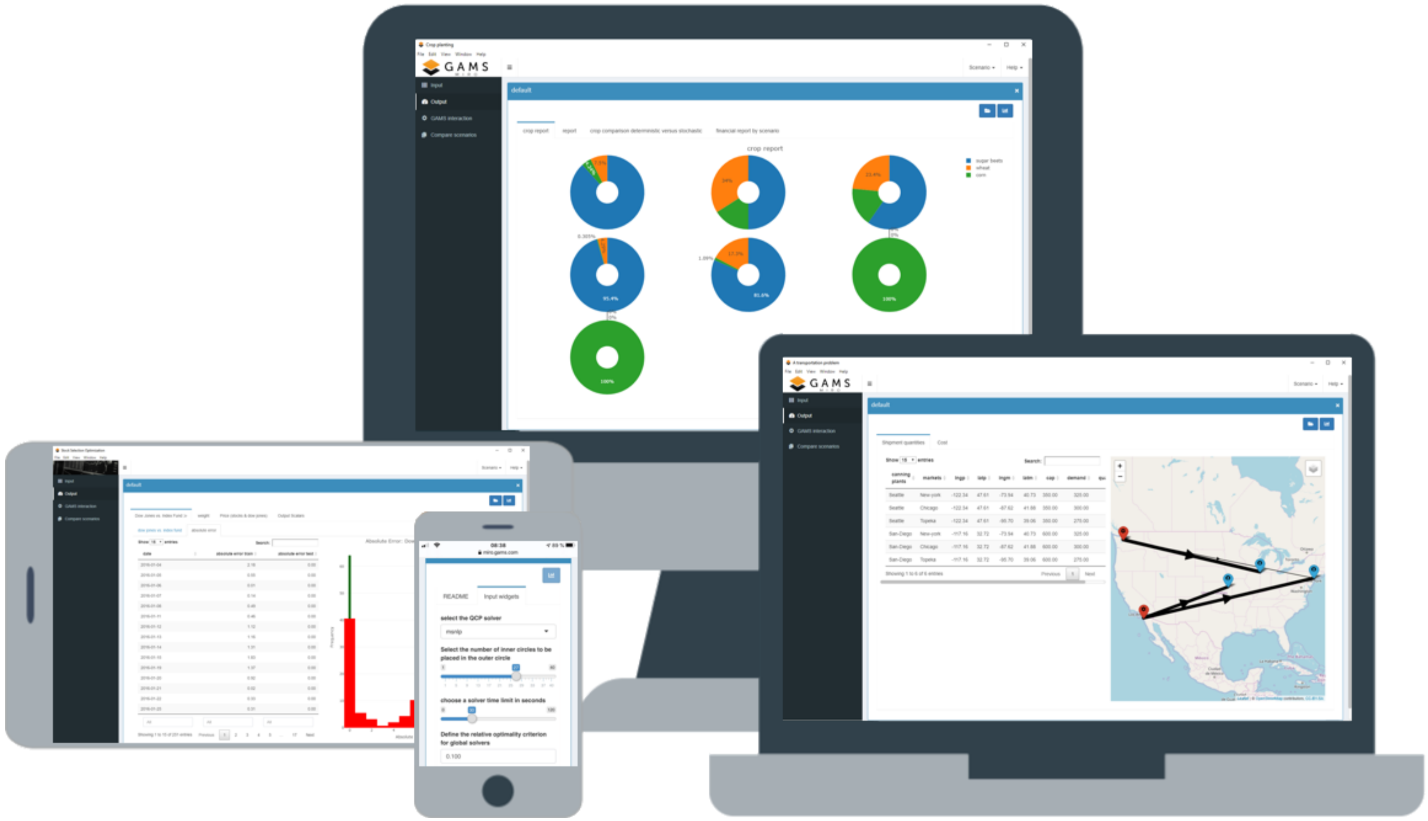
GAMS:	www.gams.com/download/
GAMS MIRO Desktop:	www.gams.com/miro/download.html

Deploying a MIRO app



- ✓ Access all your apps in one place
- ✓ Simply distribute your MIRO apps to others
- ✓ Add other people' s apps to your library

GAMS MIRO Engine: Optimizing in the Cloud



- ✓ Access via local apps or directly on a server
- ✓ High scalability
- ✓ Load balancing
- ✓ Rolling updates
- ✓ …

www.gams.com/miro/engine/

For more information visit:

www.gams.com

www.gams.com/miro

Additional material

What the Solve Statement is doing

At each Solve statement, the following happens:

1. The model (= list of (indexed) equations) is compiled into a model instance, that is, a scalar (= no indices) list of constraints and those variables, that appear in at least one constraint. (The Equation Listing and Column Listing shows in the listing file shows parts of this model.)
2. Instance statistics are written to the log:
```
--- Generating NLP model m
--- alkyl.gms(85) 5 Mb
---   8 rows 15 columns 32 non-zeroes
---   54 nl-code 19 nl-non-zeroes
```
3. Some error check is performed.
   The instance is passed on to a solver and processed there.
```
--- Executing IPOPT: elapsed 0:00:00.093
[solver log]
--- Restarting execution
```
4. Passed to a solver and processed there.
5. The result (model and solution status, solution, statistical information) is passed back to GAMS and reported in the listing file.

GAMS Log

The GAMS log can be written to the console, to standard output and/or to a file. This is controlled by the command line parameter logOption (or lo). The following items (and more) are part of the log:

- GAMS version

```
*** ************* BETA release
*** GAMS Base Module 24.5.0 r53642 BETA Released 25Aug15 WEI x86 64bit/MS Windows
*** ************* BETA release
```

- License

```
Licensee: Max Mustermann                                G141124/0001AW-GEN
GAMS Software GmbH                                                  DC8674
```

- Problem statistics

```
--   2 rows 3 columns 5 non-zeroes
--   2 discrete-columns
```

- Solver log

```
Cplex 12.8.0.0

Reading data...
Starting Cplex...
```

- Results

```
MIP Solution:    8.000000 (0 iterations, 0 nodes)
Final Solve:     8.000000 (0 iterations)
Best possible:   8.000000
Absolute gap:    0.000000
Relative gap:    0.000000
```

Listing File

Running a GAMS model generates a listing file (.lst file).

Compilation Errors:

- are indicated by ****
- contain a '$' directly below the point at which the compiler thinks the error occurred
- are explained near the end of the line-numbered listing part
- in the IDE, they are also indicated by red lines in the process (log) window (can be double-clicked)
- check carefully for the cause of the first error, fix it, and try again
- usual causes: undefined / undeclared symbols (parameters, variables, equations), unmatched brackets, missing semi-colons

Listing File: Equation and Column Listing

Equation Listing:

- listing of generated equations with sets unrolled, parameters removed, ...
- useful for model debugging: is the intended model generated?
- for nonlinear equations, a linearization in the starting point is shown

```
    AcidDef.. AcidDilut*AcidErr =e= 35.82-22.2*F4Perf;

->  AcidDef.. (1)*AcidDilut + 22.2*F4Perf + (3.6)*aciderr
              =E= 35.82 ; (LHS = 35.79, INFES = 0.03 ****)
```

- activity and violation of constraint in starting point also shown
- Column Listing:
- shows coefficients, bounds, starting values for generated variables

```
-- F4Perf F4 Performance Number

F4Perf

             (.LO, .L, .UP, .M = 1.45, 1.45, 1.62, 0)

      22.2   AcidDef

      (1)    F4Def
```

Listing File: Solve Summary

- generated for each solve command
- reporting status and result of solve

```
               S O L V E      S U M M A R Y

     MODEL   m                    OBJECTIVE F
     TYPE    NLP                  DIRECTION MINIMIZE
     SOLVER  CONOPT               FROM LINE 85

**** SOLVER STATUS     1 Normal Completion
**** MODEL STATUS      2 Locally Optimal
**** OBJECTIVE VALUE               -1.7650

 RESOURCE USAGE, LIMIT      0.006      1000.000
 ITERATION COUNT, LIMIT     16    2000000000
 EVALUATION ERRORS          0              0
```

Listing File: Solution Listing

- equation and variable primal and dual values and bounds
- marking of infeasibilities, “non-optimalities”, and unboundedness
- ‘.’ = zero

```
                        LOWER          LEVEL         UPPER        MARGINAL

-- EQU Objective          .              .             .           1.0000
-- EQU AlkylShrnk         .              .             .          -4.6116
-- EQU AcidBal            .            -0.0020         .          11.8406 INFES
-- EQU IsobutBal          .             0.0952         .           0.0563 INFES
-- EQU AlkylDef           .             0.0127         .          -1.0763 INFES
-- EQU OctDef           0.5743          0.5747       0.5743      -25.9326 INFES
-- EQU AcidDef         35.8200         35.8533      35.8200        0.2131 INFES
-- EQU F4Def           -1.3300         -1.3300      -1.3300       -4.1992

                        LOWER          LEVEL         UPPER         MARGINAL

-- VAR F               -INF            -1.4143      +INF .
-- VAR OlefinFeed         .             1.6198       2.0000       -0.1269 NOPT
-- VAR IsobutRec          .             1.3617       1.6000       -0.2133 NOPT
-- VAR AcidFeed           .             0.7185       1.2000       -0.0411 NOPT
-- VAR AlkylYld           .             2.8790       5.0000       -0.0076 NOPT
-- VAR IsobutMak          .             1.8926       2.0000       -0.4764 NOPT
-- VAR AcidStren        0.8500          0.8998       0.9300        0.5273 NOPT
```

MIRO – Model Annotations

```
Set
   i 'canning plants' / seattle,  san-diego /
   j 'markets'        / new-york, chicago, topeka /;

$onExternalInput
Parameter
   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 /;
$offExternalInput

Parameter c(i,j) 'transport cost in thousands of dollars per case';
c(i,j) = f*d(i,j)/1000;

$onExternalOutput
Variable
   x(i,j) 'shipment quantities in cases'
   z      'total transportation costs in thousands of dollars';
$offExternalOutput
```