GAMS
Transportation Model

Michael Bussieck  mbussieck@gams.com
Jan-Hendrik Jagla   jhjagla@gams.com

GAMS Software GmbH
www.gams.de
GAMS Development Corporation
www.gams.com

EURO 2009 Bonn
What is a Model?

- Mathematical Programming (MP) Model
  - List of Equations

- Collection of several intertwined MP Models
  - Data Preparation
  - Data Calibration
  - “Solution” Module (e.g. sequential, parallel, loop)
  - Report Module
A Transportation Model

Minimize Transportation cost
subject to Demand satisfaction at markets
Supply constraints
\[
\begin{align*}
\sum_{(c,p) \in \mathcal{N}} & \quad t_{\text{cost}} \cdot \text{dist}(c, p) \cdot x^c_p \quad \rightarrow \quad \min \\
\sum_{(c,p) \in \mathcal{N}} x^c_p & \quad \leq \quad \text{sup}(c) \quad \forall c \\
\sum_{(c,p) \in \mathcal{N}} x^c_p & \quad \geq \quad \text{dem}(p) \quad \forall p \\
x^c_p & \quad \geq \quad 0 \quad \forall c, p : (c, p) \in \mathcal{N}
\end{align*}
\]
GAMS Algebra

Variables
\[ x(i,j) \] shipment quantities in cases
\[ z \] total transportation costs in thousands of dollars ;

Positive Variable \( x \) ;

Equations
\[ \text{cost} \] define objective function
\[ \text{supply}(i) \] observe supply limit at plant \( i \)
\[ \text{demand}(j) \] satisfy demand at market \( j \) ;

\[ \text{cost} .. \quad z =e= \sum (i,j), c(i,j) \times x(i,j) ; \]

\[ \text{supply}(i) .. \quad \sum (j, x(i,j)) =l= a(i) ; \]

\[ \text{demand}(j) .. \quad \sum (i, x(i,j)) =g= b(j) ; \]

Model transport /all/ ;
A few Word about GAMS Syntax

• Symbols:
  – Sets
  – Parameters
  – Variables
  – Equations
  – Models
  – ASCII Output Files

```plaintext
Sets
  i  canning plants / seattle, san-diego /
Parameters
  a(i) capacity of plant i in cases
     / seattle 350
     san-diego 600 /
Variables
  x(i,j) shipment quantities in cases;
Equations
  supply(i) observe supply limit at plant i;
Model
  transport /all/ ;
File
  fx some file /‘c:\t\text.txt’ /
```

• Statements
  – Declarations
  – Data Assignments
  – Equation Definition
  – Programming Flow Control
  – Option statement

```plaintext
Parameter c(i,j);
c(i,j) = f * d(i,j) / 1000 ;
supply(i) .. sum(j, x(i,j)) =l= a(i);
loop(i, put fx i.t1);
option reslim=10;
```
Hands-on! Transportation Model

![Transportation Model Diagram]

**Parameters**

\[ a_{ij} \] capacity of plant \( i \) in cases

- seattle 350
- san-diego 600

\[ b_{ij} \] demand at market \( j \) in cases

- new-york 325
- chicago 300
- tokyo 275

**Table**

\[ d_{ij} \] distance in thousands of miles

<table>
<thead>
<tr>
<th></th>
<th>new-york</th>
<th>chicago</th>
<th>tokyo</th>
</tr>
</thead>
<tbody>
<tr>
<td>seattle</td>
<td>1.9</td>
<td>1.7</td>
<td>1.0</td>
</tr>
<tr>
<td>san-diego</td>
<td>1.5</td>
<td>1.8</td>
<td>1.4</td>
</tr>
</tbody>
</table>

**Scaler**

\[ f \] freight in dollars per case per thousand miles \( \times 100 \)

**Parameter**

\[ c_{ij} \] transport cost in thousands of dollars per case

---

--- Job transport.gms Start 07/03/07 10:25:45
GAMS Rev 146 Copyright (C) 1987-2007 GAMS Development. All rights reserved. Licensee: San-Madrik Tolga
GAMS Software GmbH 0570416-00350
---

--- Starting compilation
--- transport.gms [45] 5 Mb
--- Starting execution
--- transport.gms [45] 4 Mb
--- Executing LP model transport
--- transport.gms [66] 4 Mb
--- 6 cores, 7 columns, 18 non-zeros
--- Executing CPLEX

GAMS/Cplex Jun 1, 2007 WIN.CP.CP 22.5 034.037-041.VIS For Cplex 10 Cplex 10.2.0, GAMS Link 54
Modifications to the transport model

LP → MIP → MINLP → Put Utilities
LP → NLP → MINLP → GDX Utilities
LP → NLP → MINLP → Convert + Dumpopt
Types of Variables

- Continuous Variables
  - Free/Positive/Negative
  - Lower and/or upper bound

- Binary Variables
  - Either 0 or 1

- Integer Variables
  - by default ranging between 0 and 100

- Semicont/Semiint Variables
  - 0 or above a given minimum

- Special Ordered Set Variables (SOS1, SOS2)
Binary Variables

- Powerful Tool to model yes/no decisions

- Models with discrete variables (MIP)
  - Solved using Branch-and-Cut algorithms (lots or LPs)
  - Theoretically difficult problem class
  - Practical:
    - mixed bag
    - *Art of Modeling*

- Example: Minimum Shipment
  - Ship at least 100 tons or don’t ship
Minimum Shipment

• Continuous Variable $x$ (shipment)

• Binary Variable $ship$ (indicator for minimum shipment):
  – $ship=1$ if $x \geq 100$
  – $ship=0$ if $x = 0$

• Coupling Constraints:
  – $x \geq 100 \times ship$
  – $x \leq bigM \times ship$

• How big do we have to make $bigM$?
Implement Min/Max Shipments (MIP)

Parameter rep1(i,j,*) Shipments between plants and markets
    rep2(*) Objective value;

rep1(i,j,'lp') = x.l(i,j);
rep2('lp') = z.l;

scalars xmin / 100 /
    xmax / 325 /;

binary variables ship(i,j) decision variable to ship
equations minship(i,j) minimum shipments
    maxship(i,j) maximum shipments;

minship(i,j).. x(i,j) =g= xmin*ship(i,j);
maxship(i,j).. x(i,j) =l= xmax*ship(i,j);

model m2 min shipments / all /;
solve m2 using mip minimizing z;
rep1(i,j,'mip') = x.l(i,j);
rep2('mip') = z.l;

option mip=coincbc
solve m2 using mip minimizing z;
rep1(i,j,'mip-coincbc') = x.l(i,j);
rep2('mip-coincbc') = z.l;
display rep1,rep2;
Economy of Scales: Cost = Volume^{factor}

- factor > 1
- factor = 1
- factor < 1
Implement Nonlinear Cost (NLP)

* nonlinear cost
equation nlcost nonlinear cost function;
scalar beta;

nlcost.. z =e= sum((i,j), c(i,j)*x(i,j)**beta);

model m3 / nlcost,supply,demand /;

beta = 1.5;
solve m3 using nlp minimizing z;
rep1(i,j,'nlp-convex') = x.l(i,j);
rep2('nlp-convex') = z.l;

beta = 0.6;
solve m3 using nlp minimizing z;
rep1(i,j,'nlp-concave') = x.l(i,j);
rep2('nlp-concave') = z.l;

option nlp=baron;
solve m3 using nlp minimizing z;
rep1(i,j,'nlp-baron') = x.l(i,j);
rep2('nlp-baron') = z.l;

display rep1,rep2;
Implement Min/Max and Nonlinear (MINLP)

* min/max and nonlinear objective

model m4 / nlcost, supply, demand, minship, maxship /;

option minlp=baron;
solve m4 using minlp minimizing z;
rep1(i,j,'minlp-bar') = x.l(i,j);
rep2('minlp-bar') = z.l;

option minlp=lindo_global;
solve m4 using minlp minimizing z;
rep1(i,j,'minlp-lin') = x.l(i,j);
rep2('minlp-lin') = z.l;

display rep1, rep2;
Data Connectivity

• Data Import/Export from *Standard Applications*
  - Text files
  - Gams Data eXchange (GDX)
  - MS Office, Databases, …

• Capture an *Instance*
  - Reproducibility of Model/System Bugs
  - Problems: Life Database/different Platforms
  - convert
  - dumpopt
set help(*)
option help<rep1;

file fx /results.txt/;

put fx 'Results of different models created on ' system.date /
put '-----------------------------------------------' / /;
loop(help,  
  put 'Model:' help.te(help) /;
  put '--------------' / /;
  put 'Objective value:' rep2(help) / /
  loop((i,j)$rep1(i,j,help),
    put 'Shipment from 'i.te(i):10' to 'j.te(j):10' is: 'rep1(i,j,help) /;
  );
  put / /
);
putclose;
GDX and GDXXRW

- execute_unload 'all.gdx';
- gdx=all2
→ gdxdiff

- execute_unload 'reports.gdx' rep1, rep2;
  execute 'gdxxrw reports.gdx par=rep2 cdim=1 rdim=0 rng=Report!c1';
  execute 'gdxxrw reports.gdx par=rep1 cdim=1 rdim=2 rng=Report!a4';
Capture an Instance with convert /dumpopt

• GAMS “solver”: convert
  – gams mymodel modeltype=convert
  or
  – option minlp=convert;
    solve m4 using minlp minimizing z;

  → anonymized scalar model gams.gms and dictionary dict.txt

• GAMS option: dumpopt
  – gams mymodel dumpopt=11
  → mymodel.dmp with all source and data