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
\[
\sum_{c,p: (c,p) \in \mathcal{N}} \text{tcost} \cdot \text{dist}(c, p) \cdot x^c_p \rightarrow \min
\]

\[
\sum_{c,p: (c,p) \in \mathcal{N}} x^c_p \leq \sup(c) \quad \forall c
\]

\[
\sum_{c,p: (c,p) \in \mathcal{N}} x^c_p \geq \text{dem}(p) \quad \forall p
\]

\[
x^c_p \geq 0 \quad \forall c, p : (c, p) \in \mathcal{N}
\]
GAMS Algebra

Variables

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

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/ ;
A few Word about GAMS Syntax

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

  **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

  **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;
Demo! Transportation Model
Modifications to the transport model
Types of Variables

• Continuous Variables
  – Free/Positive/Negative
  – Lower and/or upper bound

• Binary Variables
  – Either 0 or 1

• Integer Variables
  – Any integer number

• 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 of LPs)
  - Theoretically difficult problem class
  - Practical:
    - mixed bag
    - *Art of Modeling*

- Example: Minimum Shipment
  - Ship at least 100 tons or don’t ship
Demo! Binary Vars: Minimum Shipment

- Continuous Variable $x$ (shipment)

- Binary Variable $ship$ (decision whether to ship or not):
  - $ship = 1$ if $x \geq 100$
  - $ship = 0$ if $x = 0$

- Coupling Constraints:
  - $x \geq 100 * ship$
  - $x \leq bigM * ship$

- 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 mins / 100 /
    bigm / 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= mins*ship(i,j);
maxship(i,j).. x(i,j) =l= bigm*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;
Demo! NL-Model: Economy of Scales

\[ \text{Cost} = \text{const} \cdot \text{Volume}^{\text{factor}} \]
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;
* 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=lindoglobal;
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<repship;

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:' repcost(help) / /;
   loop((i,j)$repship(i,j,help),
      put 'Shipment from 'i.te(i):10' to 'j.te(j):10' is: 'repship(i,j,help) /;
   );
   put / /;
);
putclose;
Demo! GDX and GDXXRW

- execute_unload 'all.gdx';
- gdx=all2
  \[ \rightarrow \text{gdxdiff} \]

- execute_unload 'reports.gdx' repcost, repship;
  execute 'gdxxrw reports.gdx par=repcost cdim=1 rdim=0 rng=Report!c1';
  execute 'gdxxrw reports.gdx par=repship cdim=1 rdim=2 rng=Report!a4';
Demo! Capture an Instance

- 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`
→ translation into format required by other tools
  - mps
  - mpi
  - oml
  - …
**Demo! Starting a model from a spreadsheet**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Distance</td>
<td>New-York</td>
<td>Chicago</td>
<td>Topeka</td>
<td>Supply</td>
</tr>
<tr>
<td>2</td>
<td>Seattle</td>
<td>2.5</td>
<td>1.7</td>
<td>1.8</td>
<td>350</td>
</tr>
<tr>
<td>3</td>
<td>San-Diego</td>
<td>2.5</td>
<td>1.8</td>
<td>1.4</td>
<td>600</td>
</tr>
<tr>
<td>4</td>
<td>Demand</td>
<td></td>
<td>325</td>
<td>300</td>
<td>275</td>
</tr>
<tr>
<td>7</td>
<td>Freight cost</td>
<td></td>
<td>90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>SHIPMENT</td>
<td>New-York</td>
<td>Chicago</td>
<td>Topeka</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Seattle</td>
<td>50</td>
<td>300</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>San-Diego</td>
<td>275</td>
<td>0</td>
<td>275</td>
<td></td>
</tr>
</tbody>
</table>

**GAMS Directory:** `c:\program files\GAMS23.2\`

**Working Directory:** `c:\tmp\`

**Solver:** CPLEX

Solver: CPLEX  
Equations: 6  Variables: 7  
Model Status: 1 Optimal  
Solver Status: 1 Normal Completion  
Iterations: 4  Solve Time: 0.00  
Objective Value: 153.675
## Contacting GAMS

<table>
<thead>
<tr>
<th>Region</th>
<th>Address</th>
<th>Phone</th>
<th>Fax</th>
<th>Website</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>USA</strong></td>
<td>GAMS Development Corp.</td>
<td>+1 202 342 0180</td>
<td>+1 202 342 0181</td>
<td><a href="http://www.gams.com">http://www.gams.com</a></td>
<td><a href="mailto:sales@gams.com">sales@gams.com</a>, <a href="mailto:support@gams.com">support@gams.com</a></td>
</tr>
<tr>
<td></td>
<td>1217 Potomac Street, NW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20007</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Europe</strong></td>
<td>GAMS Software GmbH</td>
<td>+49 221 949 9170</td>
<td>+49 221 949 9171</td>
<td><a href="http://www.gams.de">http://www.gams.de</a></td>
<td><a href="mailto:info@gams.de">info@gams.de</a>, <a href="mailto:support@gams-software.com">support@gams-software.com</a></td>
</tr>
<tr>
<td></td>
<td>Eupener Str. 135-137</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50933 Cologne</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>