Introduction to GAMS:  
Formulation of a simple problem

Dhazn Gillig  
&  
Bruce A. McCarl

Department of Agricultural Economics  
Texas A&M University
What is GAMS?

- **Generalized Algebraic Modeling System**
  - a language for setting up and solving mathematical programming optimization models
  - all-in-one package that allows one to
    - specify the structure of an optimization model
    - specify and calculate data that go into that model
    - solve that model
    - conduct report writing on a model
    - perform a comparative static analysis
Using GAMS

- Two approaches to using GAMS

1. Traditional method:
   use a text editor set up the model then use DOS (or UNIX) command line instructions to find errors in and run the model

2. GAMS IDE alternative:
   A graphical interface to create, debug, edit and run GAMS files.
Formulation of a Simple Problem

Maximize \[ 109X_1 + 90X_2 + 115X_3 \]

Subject to: \[ X_1 + X_2 + X_3 \leq 100 \]
\[ 6X_1 + 4X_2 + 8X_3 \leq 500 \]
\[ X_1, X_2, X_3 \geq 0 \text{ (non-negative)} \]

STEPS

1. Variable specifications
2. Equation specifications
   a. declaration
   b. algebraic structure specification
3. Model statement
4. Solve statement
VARIABLES
   Z    Variable Z ;

POSITIVE VARIABLES
   X1    Variable X1
   X2    Variable X2
   X3    Variable X3 ;

EQUATIONS
   Equation1   Equation 1
   Equation2   Equation 2
   Equation3   Equation 3 ;

Equation1..
   Z =E= 109*X1 + 90*X2 + 115*X3 ;

Equation2..
   X1 + X2 + X3 =L= 100 ;

Equation3..
   6*X1 + 4*X2 + 8*X3 =L= 500 ;

MODEL Example1 /ALL/;

SOLVE Example1 USING LP MAXIMIZING Z ;

Variable specifications
Equation declarations
Algebraic structure specification
Model statement
Solve statement
Dissecting GAMS – Variable naming

Variable Specification

GAMS requires variables in each problem to be identified. In the example, we have variables Z, X1, X2, X3

VARIABLES

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>Variable Z</td>
</tr>
</tbody>
</table>

POSITIVE VARIABLES

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>Variable X1</td>
</tr>
<tr>
<td>X2</td>
<td>Variable X2</td>
</tr>
<tr>
<td>X3</td>
<td>Variable X3</td>
</tr>
</tbody>
</table>

Why new Z? => Because GAMS requires all models to be of a special form Max CX becomes Max Z

s.t.    Z = CX

Variable Name can be up to 31 characters and can have explanatory text
Types of Variables:

- **VARIABLE**  
  unrestricted variables
- **POSITIVE VARIABLE**  
  restricted variables to be nonnegative
- **BINARY VARIABLE**  
  restricted variables to be 0 or 1
- **INTEGER VARIABLE**  
  restricted variables to be 0, 1, 2, ..., 9999
Equation Specifications consist of two parts.

1. Naming equations:
   GAMS requires the modeler name each equation, which is active in the model. In the example, the equations are named after the keyword EQUATIONS.

   Explanatory Text

   EQUATIONS
   Equation1
   Equation2
   Equation3

   Equation 1
   Equation 2
   Equation 3;

   The name for each equation can be anything up to 31 characters.
Dissecting GAMS – Equation naming

(2) Specifying algebraic structure:

After naming equations, the exact algebraic structure of equations must be specified by using .. notation.

\[
\begin{align*}
\text{Equation 1..} & \\
Z & = E = 109 \times X_1 + 90 \times X_2 + 115 \times X_3 \quad ; \\
\text{Maximize} & \\
& 109X_1 + 90X_2 + 115X_3 \\
\text{Equation 2..} & \\
X_1 + X_2 + X_3 & = L = 100 \\
\text{Subject to:} & \\
& X_3 + X_3 \leq 100 \\
& 6X_3 + 4X_3 \leq 500 \\
& X_1, X_2, X_3 \geq 0 \text{ (non-negative)} \\
\text{Equation 3..} & \\
6X_1 + 4X_2 + 8X_3 & = L = 500 \\
\end{align*}
\]

This algebraic form involves use of a special syntax to tell the exact form of the equation that may actually be an inequality.

- \( = E = \) indicates an equality constraint
- \( = L = \) indicates a less than or equal to constraint
- \( = G = \) indicates a greater than or equal to constraint
Dissecting GAMS – Model specification

- Model Specification

**MODEL** statement is used to identify models that will be solved. It involves 2 steps

**step 1:** give name of the model (e.g. Example1)

**step 2:** specify equations that will be included in the model in slashes / /

**MODEL Example1 /ALL/ ;** including all equations

**MODEL Example1 /Equation1, Equation2/ ;** Only Equation1 and Equation2
Dissecting GAMS – Solve specification

- **Solve Specification**

  `SOLVE` causes GAMS to apply a solver to the model named in the solve statement *(Example1)* using the data defined just before the solve statement.

  - `SOLVE Example1 USING LP MAXIMIZING Z ;`  
    - LP MAX
  
  - `SOLVE Example1 USING LP MINIMIZATION Z ;`  
    - LP MIN
  
  - `SOLVE Example1 USING MIP MAXIMIZING Z ;`  
    - Mixed integer programming
  
  - `SOLVE Example1 USING NLP MAXIMIZING Z ;`  
    - Non-linear programming
Dissecting GAMS – ; specification

GAMS requires one to terminate each statement with a ;.

```
VARIABLES
   Z        Variable Z
POSITIVE VARIABLES
   X1        Variable X1
   X2        Variable X2
   X3        Variable X3
EQUATIONS
   Equation1   Equation 1
   Equation2   Equation 2
   Equation3   Equation 3

Equation1..
   Z =E= 109*X1 + 90*X2 + 115*X3
Equation2..
   X1 + X2 + X3 =L= 100
Equation3..
   6*X1 + 4*X2 + 8*X3 =L= 500

MODEL Example1 /ALL:/
SOLVE Example1 USING LP MAXIMIZING Z:
```

; is a very important part of the syntax their omission => syntax errors.
Dissecting GAMS – Finding errors

**EQUATIONS**

Equation1    Equation 1
Equation2    Equation 2
Equation3    Equation 3
Equation1..  ??

\[ Z = E = 109 \times X_1 + 90 \times X_2 + 115 \times X_3 \]

**Error Message**

--- Starting compilation
--- EXAMPLE1.GMS(13) 1 Mb 1 Error
*** Error 096 in C:\TASANA\641CLASS_GAMS_SECTION_SPRING2003\PROGRAM\EXAMPLE1.GMS
   Blank needed between identifier and text
   (-or- illegal character in identifier)
   (-or- check for missing ';' on previous line)
--- EXAMPLE1.GMS(14) 1 Mb 3 Errors
*** Error 195 in C:\TASANA\641CLASS_GAMS_SECTION_SPRING2003\PROGRAM\EXAMPLE1.GMS
   Symbol redefined with a different type
*** Error 097 in C:\TASANA\641CLASS_GAMS_SECTION_SPRING2003\PROGRAM\EXAMPLE1.GMS
   Explanatory text can not start with '$$', '=', or '..'
   (-or- check for missing ';' on previous line)
Solution Reports

The report summary gives the total number of non-optimal, infeasible, unbounded cases encountered.

```
**** REPORT SUMMARY :  
0 NONOPT
0 INFEASIBLE
0 UNBOUNDED
```

Solution information can be output in several ways:

1. GAMS default standard solution output
2. Through added DISPLAY commands
3. Through added computed reports using values from solutions (we will discuss this later)
Solution Reports

1. Format of the default standard GAMS solution

<table>
<thead>
<tr>
<th>LOWER</th>
<th>LEVEL</th>
<th>UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>EQU</td>
<td>Equation1</td>
<td>.</td>
</tr>
<tr>
<td>----</td>
<td>EQU</td>
<td>Equation2</td>
<td>-INF</td>
</tr>
<tr>
<td>----</td>
<td>EQU</td>
<td>Equation3</td>
<td>-INF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOWER</th>
<th>LEVEL</th>
<th>UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>VAR</td>
<td>Z</td>
<td>-INF</td>
</tr>
<tr>
<td>----</td>
<td>VAR</td>
<td>X1</td>
<td>.</td>
</tr>
<tr>
<td>----</td>
<td>VAR</td>
<td>X2</td>
<td>.</td>
</tr>
<tr>
<td>----</td>
<td>VAR</td>
<td>X3</td>
<td>.</td>
</tr>
</tbody>
</table>

The single dot “.” represents a zero; INF represents infinity
Solution Reports

2. A DISPLAY command


VARIABLE Z.L = 9950.000 Variable Z
VARIABLE X1.L = 50.000 Variable X1
VARIABLE X2.L = 50.000 Variable X2
VARIABLE X3.L = 0.000 Variable X3

EQUATION Equation1.M = 1.000 Equation 1
EQUATION Equation2.M = 52.000 Equation 2

Shadow prices and reduced costs denoted using .M

syntax .L and .M is used to address optimal solution variable and shadow price values.

Variables solution values denoted with .L

Solution Reports

2. A DISPLAY command


VARIABLE Z.L = 9950.000 Variable Z
VARIABLE X1.L = 50.000 Variable X1
VARIABLE X2.L = 50.000 Variable X2
VARIABLE X3.L = 0.000 Variable X3

EQUATION Equation1.M = 1.000 Equation 1
EQUATION Equation2.M = 52.000 Equation 2

Shadow prices and reduced costs denoted using .M

syntax .L and .M is used to address optimal solution variable and shadow price values.

Variables solution values denoted with .L
GAMS Output

Echo Prints or a copy of an input file

2 VARIABLES
3 Z Variable Z ;
4 POSITIVE VARIABLES
5 X1 Variable X1
6 X2 Variable X2
7 X3 Variable X3 ;
8 EQUATIONS
9 Equation1 Equation 1
10 Equation2 Equation 2
11 Equation3 Equation 3 ;
12 Equation1.
13 Z =E= 109*X1 + 90*X2 + 115*X3 ;
14 Equation2.
15 X1 + X2 + X3 =L= 100 ;
16 Equation3.
17 6*X1 + 4*X2 + 8*X3 =L= 500 ;
18
19 MODEL Example1 /ALL/;
20 SOLVE Example1 USING LP MAXIMIZING Z ;
GAMS Output

Model Statistics

MODEL STATISTICS

<table>
<thead>
<tr>
<th>Counts Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocks of Equations</td>
<td>3</td>
</tr>
<tr>
<td>Blocks of Variables</td>
<td>4</td>
</tr>
<tr>
<td>Non Zero Elements</td>
<td>10</td>
</tr>
<tr>
<td>Single Equations</td>
<td>3</td>
</tr>
<tr>
<td>Single Variables</td>
<td>4</td>
</tr>
</tbody>
</table>

Model Statistics give information on model size. The BLOCK counts refer to the number of named GAMS equations and variables. The SINGLE counts refers to individual rows and columns. The NON ZERO refers to the number of non-zero coefficients.
GAMS Output

Status Reports

SOLVE SUMMARY

MODEL Example1
TYPE LP
SOLVER CPLEX

OBJECTIVE Z
DIRECTION MAXIMIZE
FROM LINE 20

**** SOLVER STATUS 1 NORMAL COMPLETION
**** MODEL STATUS 1 OPTIMAL
**** OBJECTIVE VALUE 9950.000

RESOURCE USAGE, LIMIT 0.000 1000.000
ITERATION COUNT, LIMIT 2 10000

SOLVE STATUS (a state of the program)
=> normal completion, iteration interrupted, resource interrupted,
terminated by solver, evaluation error limit, unknown.

MODEL STATUS (what solution looks like)
=> optimal, infeasible, unbounded, etc.
### GAMS Output

#### Solution Reports

<table>
<thead>
<tr>
<th></th>
<th>LOWER</th>
<th>LEVEL</th>
<th>UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQU Equation1</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>1.000</td>
</tr>
<tr>
<td>EQU Equation2</td>
<td>-INF</td>
<td>100.00</td>
<td>100.00</td>
<td>52.000</td>
</tr>
<tr>
<td>EQU Equation3</td>
<td>-INF</td>
<td>500.00</td>
<td>500.00</td>
<td>9.500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>LOWER</th>
<th>LEVEL</th>
<th>UPPER</th>
<th>MARGINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR Z</td>
<td>-INF</td>
<td>9950.00</td>
<td>+INF</td>
<td>.</td>
</tr>
<tr>
<td>VAR X1</td>
<td>.</td>
<td>50.000</td>
<td>+INF</td>
<td>.</td>
</tr>
<tr>
<td>VAR X2</td>
<td>.</td>
<td>50.000</td>
<td>+INF</td>
<td>.</td>
</tr>
<tr>
<td>VAR X3</td>
<td>.</td>
<td>.</td>
<td>+INF</td>
<td>-13.000</td>
</tr>
</tbody>
</table>

**Variables** => values under lower and upper refers to bounds

**Equations** => values under lower and upper are from RHS and relational type of equations

<table>
<thead>
<tr>
<th>Type</th>
<th>.LO</th>
<th>.UP</th>
<th>.L</th>
</tr>
</thead>
<tbody>
<tr>
<td>=E=</td>
<td>RHS</td>
<td>RHS</td>
<td>RHS</td>
</tr>
<tr>
<td>=L=</td>
<td>-INF</td>
<td>RHS</td>
<td>AX</td>
</tr>
<tr>
<td>=G=</td>
<td>RHS</td>
<td>INF</td>
<td>AX</td>
</tr>
</tbody>
</table>
This is a final section of the solution listing. It shows the counts rows or columns that have been marked NOPT, INFES, UNBND in the solution listing section.
GAMS Output

File Summary

**** FILE SUMMARY

INPUT       C:\TASANA\641CLASS_GAMS_SECTION_SPRING2003\PROGRAM\EXAMPLE1.GMS
OUTPUT      C:\TASANA\641CLASS_GAMS_SECTION_SPRING2003\PROGRAM\EXAMPLE1.LST

This is a final section of the output. It gives the names of the input and output disk files. If work files (save or restart) have been used, they will be named here as well.

**** FILE SUMMARY

RESTART     C:\TASANA\641CLASS_GAMS_SECTION_SPRING2003\PROGRAM\T\A1.G0?
INPUT       C:\TASANA\641CLASS_GAMS_SECTION_SPRING2003\PROGRAM\LOOP.GMS
OUTPUT      C:\TASANA\641CLASS_GAMS_SECTION_SPRING2003\PROGRAM\LOOP.LST
SAVE        C:\TASANA\641CLASS_GAMS_SECTION_SPRING2003\PROGRAM\T\A2.G0?
Download GAMS

You can download the current GAMS distribution by going to

http://www.gams.com/download/

Then

1. Fill out the form

2. After that user id and the password will automatically be e-mailed to the email address you provide within a few minutes.
GAMS documentation is accessible through the HELP menu under the choice GAMS.

GAMS User Guide: 2003

by

Bruce A. McCarl
Regents Professor of Agricultural Economics
Texas A&M University
Download McCarl GAMS User Guide 2003

You can download the current McCarl GAMS User Guide 2003 going to


Installing the McCarl's GAMS User Guide

1. Either: Copy the file mccdoc.zip (located in the Idocs directory on your GAMS CD) into your GAMS system directory. Then go to step 3.

2. Or: Download the file http://www.gams.com/docs/mccdoc.zip and place it into your GAMS system directory. Then go to step 3.

3. In the IDE go to

File > Options
Download McCarl GAMS User Guide 2003

Then, go to **File => Options**

Then, select the **Execute** tab. Click on the **Update button**, review the files in the list, and click continue to install all modules listed. The file **mccdoc.zip** is the McCarl's User Guide.
Please write an algebraic model (a feeding problem) into GAMS

Min \ 3X_1 \ + \ 4X_2 \ + \ 5X_3

s.t  \ 0.01X_1 + 0.07X_2 + 0.03X_3 \ \geq \ 0.05
     \ 0.02X_1 + 0.14X_2 + 0.1X_3 \ \geq \ 0.09
     \ X_1 + \ X_2 + \ X_3 \ = \ 1.00
     \ X_1, \ X_2, \ X_3 \ \geq \ 0

Please save your work as handson1.gms since it will be used again later.
McCarl, B. A. Basic GAMS class. (http://agecon.tamu.edu/faculty/mccarl/mccarl.htm).