Getting Started with GAMS

Thomas F. Rutherford

Department of Agricultural and Applied Economics
University of Wisconsin, Madison

WiNDC Short Course – 19 July 2021

Outline

- Brewing beer with GAMS
- Optimization hierarchy, solvers and modeling languages

Making Beer

The local brewery produces two varieties of beer (lagers and ales) which are marketed in taverns and grocery around town. At the moment, they are planning production for fall. Each beer requires malt, hops and yeast. The lagers return \$120 in profit per batch while ales earn only \$90 per batch. Lagers are made with German hops, while ales are made with Wisconsin hops. There are currently sufficient German hops in stock for 1000 batches of lager and Wisconsin hops for 1500 batches of ale. Lager requires 4 kg of malt per batch while ale uses only 2 kg. Both beers require one kg of yeast per batch. There are 1,750 kg of yeast and 4800 kg of malt on hand.

What quantities of lager and ale should be produced from these supplies to maximize total profit assuming that all that are made can be sold?

Beer Production Data

Recipe for brewing beer

	kg per batch				
	malt	yeast	German Hops	Wisconsin Hops	Profit ($)
Lager	4	1	1	0	12
Ales	2	1	0	1	9

Current inventory (kg)

	malt	yeast	German Hops	Wisconsin Hops
in stock	4800	1750	1000	1500

Elements of the profit maximization model

1. Decision variables

   x : number of batches of lager produced
   y : number of batches of ales produced

2. Constraints

   $4x + 2y \leq 4800$ (malt budget)
   $x + y \leq 1750$ (yeast budget)
   $x \leq 1000$ (German hops budget)
   $y \leq 1500$ (Wisconsin hops budget)
   $0 \leq x$ (non-negative lager production)
   $0 \leq y$ (non-negative ale production)

3. Objective function

   $\max 12x + 9y$ (profit)

   in which max means *maximize*.

Brewery Profit Model (optimization format)

$$\max_{x,y} 120x + 90y$$

subject to:

$$4x + 2y \leq 4800$$

$$x + y \leq 1750$$

$$0 \leq x \leq 1000$$

$$0 \leq y \leq 1500$$

- Note that this is an instance of a *linear program* (LP), which is a type of optimization model.

Decision Variables and Parameters

$$\max_{x,y} 120x + 90y$$

subject to:

$$4x + 2y \leq 4800$$

$$x + y \leq 1750$$

$$0 \leq x \leq 1000$$

$$0 \leq y \leq 1500$$

- Decision variables are the unknowns (endogenous), and parameters are data (exogenous)

Brewery Profit Model – Generic

$$\max_{x,y} r_x x + r_y y$$

subject to:

$$a_{1x} x + a_{1y} y \leq b_1$$

$$a_{2x} x + a_{2y} y \leq b_2$$

$$\ell_x \leq x \leq u_x$$

$$\ell_y \leq y \leq u_y$$

- By changing the parameters, we create a different *instance* of the same model.
- It is good practice to separate parameters (data) from the algebraic structure of the model.

Brewery Profit Model – GAMS Studio for Windows/Mac

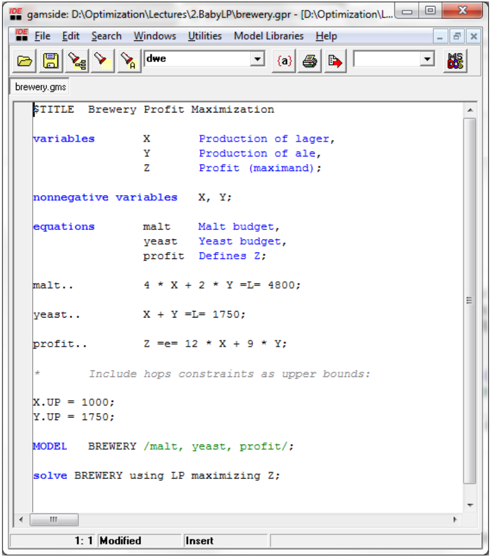
GAMS Studio

File Edit GAMS MIRO Tools View Help

brewery.gms

```
$TITLE  Brewery Profit Maximization

variables           X       Production of lager,
                    Y       Production of ale,
                    Z       Profit (maximand);

nonnegative variables   X, Y;

equations           malt    Malt budget,
                    yeast   Yeast budget,
                    profit  Defines Z;

malt..              4 * X + 2 * Y =L= 4800;

yeast..             X + Y =L= 1750;

profit..            Z =e= 12 * X + 9 * Y;

*        Include hops constraints as upper bounds:

X.UP = 1000;
Y.UP = 1750;

MODEL    BREWERY /malt, yeast, profit/;

solve BREWERY using LP maximizing Z;
```

Brewery Profit Model – IDE for Windows



Brewery Profit Model – GAMS Code

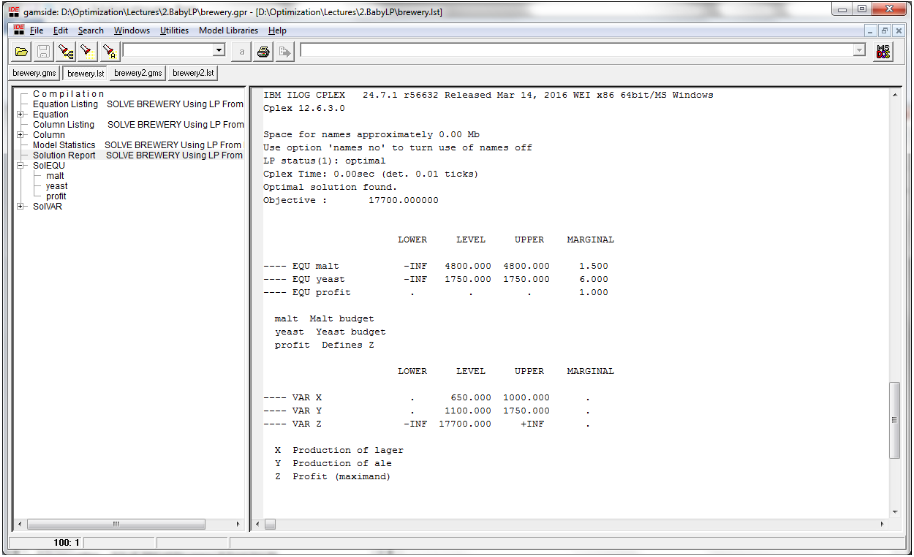
```
$TITLE  Brewery Profit Maximization

variables           X       Production of lager,
                    Y       Production of ale,
                    Z       Profit (maximand);

nonnegative variables   X, Y;

equations           malt    Malt budget,
                    yeast   Yeast budget,
                    profit  Defines Z;

malt..              4 * X + 2 * Y =L= 4800;

yeast..             X + Y =L= 1750;

profit..            Z =e= 12 * X + 9 * Y;

*       Include hops constraints as upper bounds:

X.UP = 1000;
Y.UP = 1750;

MODEL   BREWERY /malt, yeast, profit/;

solve BREWERY using LP maximizing Z;
```

Decision variables are written in upper case. GAMS is *case-insensitive*, yet this is a formatting technique which makes it easier to read GAMS models.

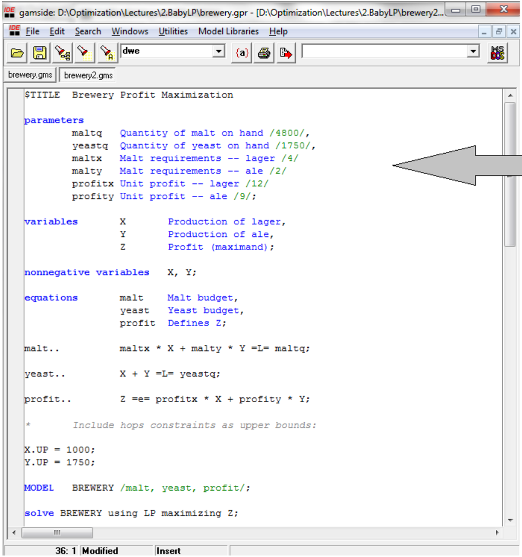
This model is written with explicit values which is not good programming practice. Better to read the data into *parameters* and then express the equations of the model in terms of the parameters.

A GAMS model is a collection of equations. The variables entering a GAMS model are those which enter the equations of the model. For linear programming models, one *free variable* must be used to define the objective function.

Brewery Profit Model – Solution Listing



Brewery Profit Model – GAMS Code with Parameters




```
$TITLE  Brewery Profit Maximization

parameters
        maltq   Quantity of malt on hand /4800/,
        yeastq  Quantity of yeast on hand /1750/,
        maltx   Malt requirements -- lager /4/
        malty   Malt requirements -- ale /2/
        profitx Unit profit -- lager /12/
        profity Unit profit -- ale /9/;

variables       X       Production of lager,
                Y       Production of ale,
                Z       Profit (maximand);

nonnegative variables   X, Y;

equations       malt    Malt budget,
                yeast   Yeast budget,
                profit  Defines Z;

malt..          maltx * X + malty * Y =L= maltq;

yeast..         X + Y =L= yeastq;

profit..        Z =e= profitx * X + profity * Y;

*       Include hops constraints as upper bounds:

X.UP = 1000;
Y.UP = 1750;

MODEL   BREWERY /malt, yeast, profit/;

solve BREWERY using LP maximizing Z;
```

Declare parameters (scalars) and provide initial values. We then use these in the model in place of explicit values. This programming technique permits us to solve the model in sequence with different parameter values.

*The most interesting answer to any question in economics is... "**It depends**".*

Brewery Profit Model – Reporting

brewery.gms | brewery2.gms

```
solve BREWERY using LP maximizing Z;


parameter        summary         Summary of solution values;
summary("Z","Ref")  = Z.L;
summary("X","Ref")  = X.L;
summary("Y","Ref")  = Y.L;
summary("malt","Ref")  = maltx*X.L + malty*Y.L;
summary("yeast","Ref") = X.L + Y.L;
```

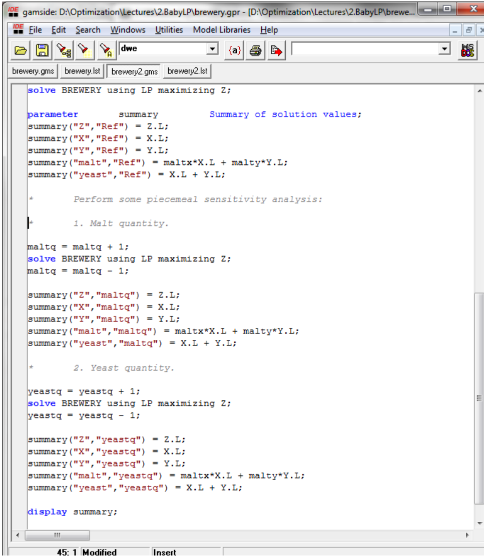
GAMS is a scripting language, and therefore it is quite forgiving. Parameters may be declared without a dimension or domain, leaving these to be determined by how they are used. In this case, GAMS ascertains that *summary* is a two-dimensional object when it is assigned.

<variable>.L returns the "level value" of a variable (or equation). This the numeric value of the variable, in this case, at the solution of the linear program.

Note that level values can be used in calculations. Here we calculate the quantity of malt and quantity of yeast employed in the LP solution.

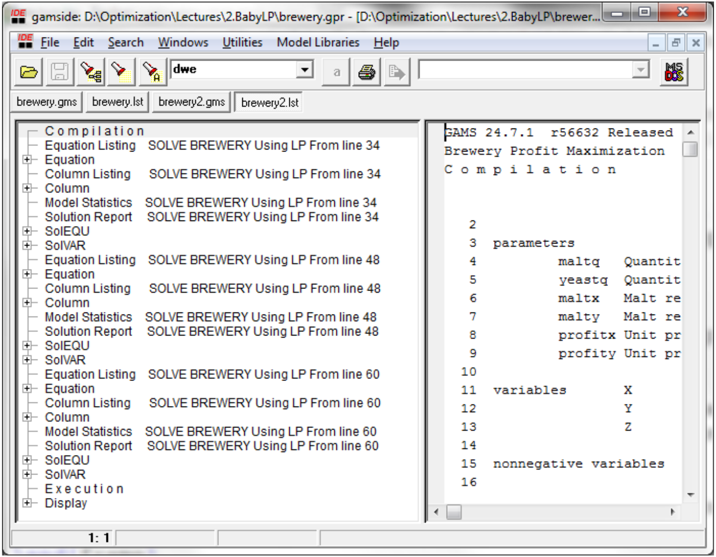
As a programming language, GAMS incorporates both *declarative* and *procedural* elements. The model equations are declarative. The assignment of values to the report parameter (summary) is procedural.

Brewery Profit Model – Sensitivity Analysis

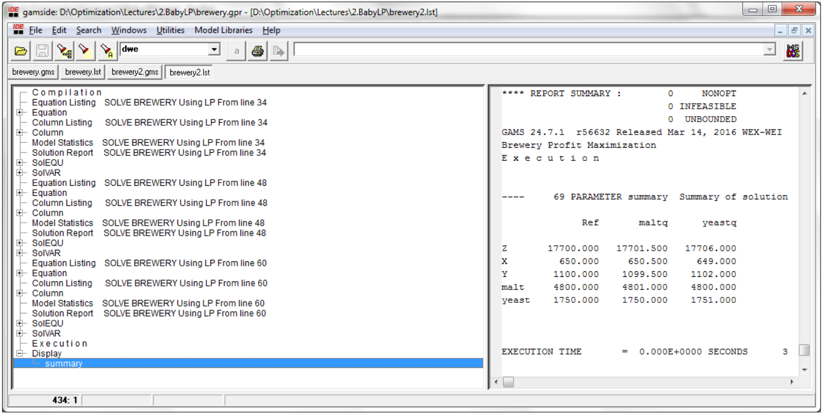



```
solve BREWERY using LP maximizing Z;

parameter        summary         Summary of solution values;
summary("Z","Ref")  = Z.L;
summary("X","Ref")  = X.L;
summary("Y","Ref")  = Y.L;
summary("malt","Ref")  = maltx*X.L + malty*Y.L;
summary("yeast","Ref")  = X.L + Y.L;

*        Perform some piecemeal sensitivity analysis:

*        1. Malt quantity.

maltq = maltq + 1;
solve BREWERY using LP maximizing Z;
maltq = maltq - 1;

summary("Z","maltq")  = Z.L;
summary("X","maltq")  = X.L;
summary("Y","maltq")  = Y.L;
summary("malt","maltq")  = maltx*X.L + malty*Y.L;
summary("yeast","maltq")  = X.L + Y.L;

*        2. Yeast quantity.

yeastq = yeastq + 1;
solve BREWERY using LP maximizing Z;
yeastq = yeastq - 1;

summary("Z","yeastq")  = Z.L;
summary("X","yeastq")  = X.L;
summary("Y","yeastq")  = Y.L;
summary("malt","yeastq")  = maltx*X.L + malty*Y.L;
summary("yeast","yeastq")  = X.L + Y.L;

display summary;
```

Brewery Profit Model – GAMS IDE Listing



Brewery Profit Model – Summary Listing



Sensitivity Analysis and Marginal Values

- Solution listing from the first solve (reference case):

```
                          LOWER      LEVEL      UPPER     MARGINAL
---- EQU malt              -INF    4800.000   4800.000      1.500
---- EQU yeast             -INF    1750.000   1750.000      6.000
---- EQU profit              .          .          .         1.000
```

- Results from the piecemeal sensitivity analysis:

```
----     69 PARAMETER summary  Summary of solution values

              Ref        maltq       yeastq
Z        17700.000    17701.500    17706.000
X          650.000      650.500      649.000
Y         1100.000     1099.500     1102.000
malt      4800.000     4801.000     4800.000
yeast     1750.000     1750.000     1751.000
```

- Notice how changes in the value of Z correspond to marginal values in the reference case solution.

Background Reading: Introduction to GAMS

https://www.gams.com/latest/docs/UG_Tutorial.html

Outline

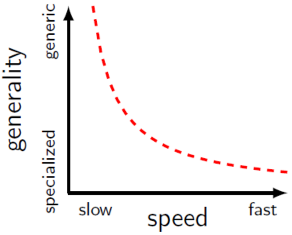
- Recap: brewing beer with GAMS
- Optimization hierarchy, solvers and modeling languages

Optimization Hierarch

- **Categories of models**: LP, QP, MIP, NLP, MCP
  - types of variables (continuous versus discrete)
  - types of constraints (equations, inequalities, linear, nonlinear)
  - types of cost functions (quadratic, convex nonlinear, conic, ...)
  - Example: every linear program (LP) has
    - continuous variables
    - linear constraints
    - a linear cost function
- **Algorithms**: gradient descent, simplex, interior point, quasi-Newton.
- **Solvers**: CPLEX, Mosek, Knitro, Minos, Conopt, Ipopt, Gurobi, Path

Optimization Methods

Numerical (usually iterative) procedures can solve instances of optimization problems. Typically, more specialized algorithms are faster. For example, quadratic programs are a particular type of nonlinear program. When a QP is convex, it can be solved using an interior point linear programming solver such as CPLEX or Mosek, and these solvers will be much more efficient than general purpose nonlinear optimization solvers such as Conopt or Minos.



Solvers

- Solvers are implementations of algorithms. Sometimes they can be quite clever!
- typically implemented in C/C++ or Fortran
- may use sophisticated error-checking, complex heuristics etc.
- Availability varies:
  - some are open-source
  - some are commercial
  - some have .edu versions

Modeling Languages

Modeling languages provide a way to interface with many different solvers using a common language.

- Can be a self-contained language (GAMS, AMPL)
- Some are implemented in other languages (JuMP in Julia, CVX in Matlab)
- Availability varies:
    - some are open-source
    - some are commercial
    - some have .edu versions

Solvers in GAMS

Options

Editor | Execute | Output | Solvers | Licenses | Colors | File Extensions | Charts/GDX | Execute2

Project Defaults | Reset | Legend

Solver	License	CNS	DNLP	EMP	LP	MCP	MINLP	MIP	MIQCP	MPEC	NLP	QCP	RMINLP	RMIP	RMIQCP	RMPEC
ALPHAECP	82 days						•		•							
AMPL	82 days	-	-		-	-	-	-		-	-		-	-		-
ANTIGONE	Demo	•	•				•		•		•	•	•		•	
BARON	82 days	•	•		•		•	•	•		•	•	•	•	•	
BDMLP	82 days				•			•						•		
BENCH	82 days	-	-		-	-	-	-	-	-	-	-	-	-	-	-
BONMIN	82 days						•		•							
BONMINH	82 days						•		•							
CBC	82 days				•			•						•		
CONOPT	82 days	•	•		•						•	•	•	•	•	
CONVERT	82 days	-	-		-	-	-	-	-	-	-	-	-	-	-	-
COUENNE	82 days	•	•				•		•		•	•	•		•	
CPLEX	82 days				•			•	•			•		•	•	
DE	82 days			-												
DECIS	82 days			-												
DECISC	82 days				-											
DECISM	82 days				-											
DICOPT	82 days						•		•							
EXAMINER	82 days		-		-	-	-	-	-	-	-	-	-	-	-	-
GAMSCHK	82 days		-		-	-	-	-	-		-	-	-	-	-	
GLOMIQO	82 days								•			•			•	
GUROBI	82 days				•			•	•			•		•	•	
IPOPT	82 days	•	•		•						•	•	•	•	•	
IPOPTH	82 days	•	•		•						•	•	•	•	•	
JAMS	82 days			•												
KESTREL	82 days	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
KNITRO	82 days	•	•		•		•		•	•	•	•	•	•	•	•
LGO	82 days		•		•						•	•	•	•	•	
LINDO	Demo		•	•	•		•	•	•		•	•	•	•	•	
LINDOGLOBAL	82 days		•		•		•	•	•		•	•	•	•	•	
LINGO	82 days		-		-		-	-			-		-	-		
LOCALSOLVER	Demo	•	•				•	•	•		•	•	•		•	
LOGMIP	82 days			•												

OK | Cancel

Using Solvers in GAMS

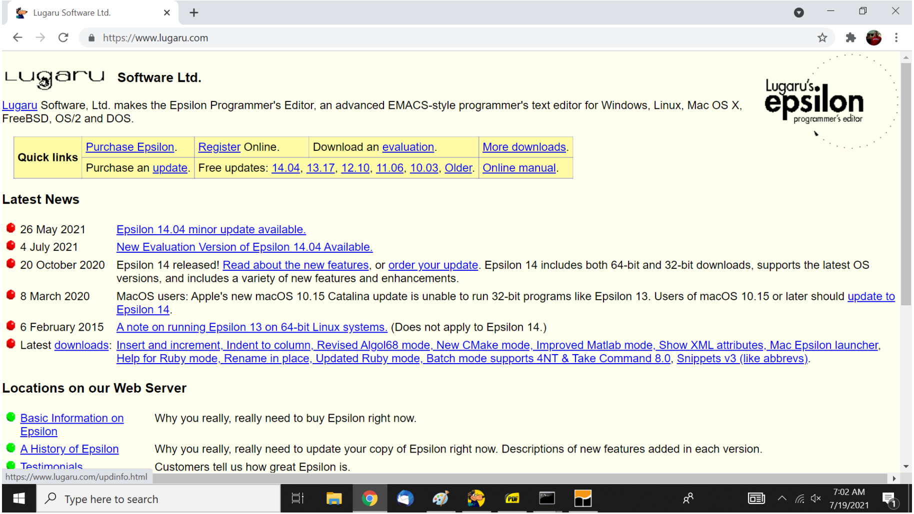
- Default solvers can be assigned for different problem types.
- A specific solver can be selected. For example, we can use the NLP solver to solve a linear program:

```
option LP=conopt;

solve BREWERY using LP maximizing Z;
```

Command Line Tools for GAMS (Text Editors)

- EMACS (GAMS mode from ShiroTakeda)
- Lugaru Epsilon (commercial version of EMACS)
- VSCode (gms mode by Laurent Drouet)
- ATOM (popular text editor with atom-language-gams by xhokir)

Evaluation License for Epsilon



Lugaru Software Ltd.

https://www.lugaru.com

Lugaru Software Ltd.

Lugaru Software, Ltd. makes the Epsilon Programmer's Editor, an advanced EMACS-style programmer's text editor for Windows, Linux, Mac OS X, FreeBSD, OS/2 and DOS.

Quick links	Purchase Epsilon.	Register Online.	Download an evaluation.	More downloads.
	Purchase an update.	Free updates: 14.04, 13.17, 12.10, 11.06, 10.03, Older.		Online manual.

Latest News

- 26 May 2021 — Epsilon 14.04 minor update available.
- 4 July 2021 — New Evaluation Version of Epsilon 14.04 Available.
- 20 October 2020 — Epsilon 14 released! Read about the new features, or order your update. Epsilon 14 includes both 64-bit and 32-bit downloads, supports the latest OS versions, and includes a variety of new features and enhancements.
- 8 March 2020 — MacOS users: Apple's new macOS 10.15 Catalina update is unable to run 32-bit programs like Epsilon 13. Users of macOS 10.15 or later should update to Epsilon 14.
- 6 February 2015 — A note on running Epsilon 13 on 64-bit Linux systems. (Does not apply to Epsilon 14.)
- Latest downloads: Insert and increment, Indent to column, Revised Algol68 mode, New CMake mode, Improved Matlab mode, Show XML attributes, Mac Epsilon launcher, Help for Ruby mode, Rename in place, Updated Ruby mode, Batch mode supports 4NT & Take Command 8.0, Snippets v3 (like abbrevs).

Locations on our Web Server

- Basic Information on Epsilon — Why you really, really need to buy Epsilon right now.
- A History of Epsilon — Why you really, really need to update your copy of Epsilon right now. Descriptions of new features added in each version.
- Testimonials — Customers tell us how great Epsilon is.

https://www.lugaru.com/updinfo.html

Type here to search

7:02 AM
7/19/2021