Speeding up GAMS Execution Time

Bruce A. McCarl

Specialist in Applied Optimization
Professor of Agricultural Economics, Texas A&M
Principal, McCarl and Associates

mccarl@tamu.edu
mccarl@bihs.net
agrinet.tamu.edu/mccarl

979-693-5694
979-845-1706

Speeding up GAMS Execution Time
Basics
(Fixmodel.pdf ch 10&12)

Sometimes GAMS model execution time and memory usage is a function of the GAMS implementation. Sometimes in slow models or memory hogs model execution time and memory usage can be reduced by altering the implementation without altering the results of the program

I have reduced execution time from 30 minutes to 15 seconds by rewriting a small amount of GAMS code without changing results

Here I cover

Diagnosis	whether and where there is a problem
Causality	features of GAMS which cause time problems to occur
Repair	manipulation of the GAMS code to repair the problem.

Coverage is aimed toward the reduction of the time to execute a problem within GAMS not within the GAMS solvers

Speeding up GAMS Execution Time
When do I look for excessive time use?

GAMS can take a lot of time or use a lot of space in computations and model setup. When confronted by program that takes a long time ask yourself some questions

Does the program take more time than you feel it should?

During execution does the screen show execution of one line number for a long time?

Is the procedure used often enough that efficiency is a concern?

If the answer to any of these questions is a yes then further investigation is in order to see whether there are poorly executing portions of the program.

Speeding up GAMS Execution Time

How do I find where excessive time is being used?

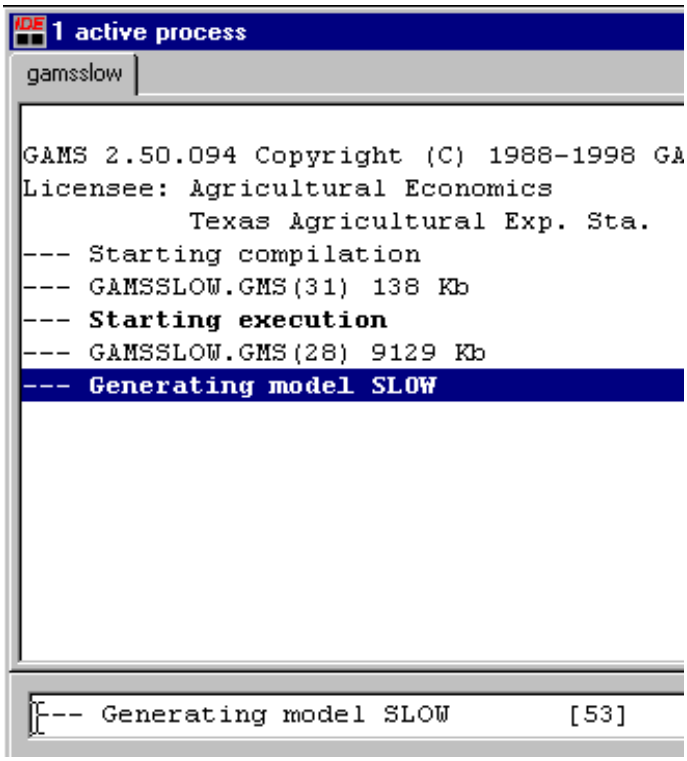
Tracking program execution

- Screen Watching
- Profile and Profiletol usage

Each is briefly covered below

Speeding up GAMS Execution Time
How do I find where excessive time is being used?
Screen watching

During execution GAMS reports the line numbers which it is executing although this is not always accurate in the latest version



If the program pauses on a line number for a moderately long time, then one would look at that line or perhaps 1 or 2 lines later as a cause of slow execution.

But screen watching as you may miss things and GAMS line reporting can be misleading when loops and if statements are being executed. Sometimes it is 2 lines behind in latest version

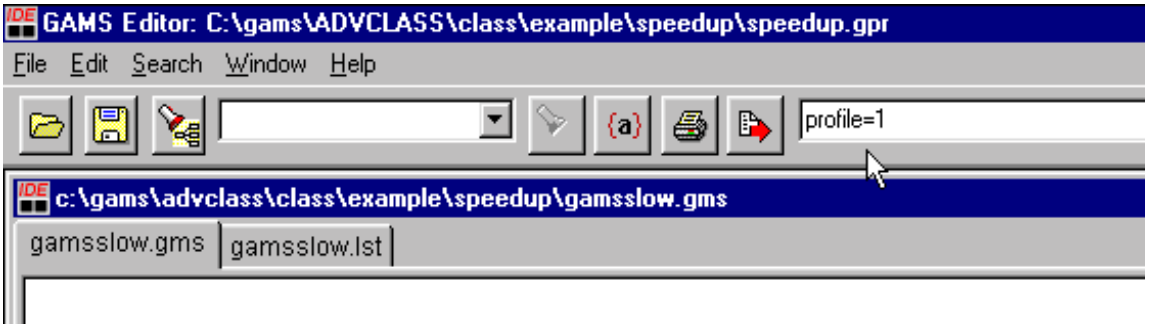
Speeding up GAMS Execution Time
How do I find where excessive time is being used?
Profile and Profiletol

GAMS can give information on statement execution time and associated memory usage by employing profile.

Invoking profile

GAMS MYMODEL PROFILE = 1

In IDE go through option dialogue



or place an option statement into the program as follows:

OPTION PROFILE=3;

Speeding up GAMS Execution Time
How do I find where excessive time is being used?
Profile

Profile report contents

a) ---- indicating this is a profile line (also contains SECS The better thing to search for)
b) GAMS statement number of the instruction being PROFILEd;
c) the symbol name of the item being worked on;
d) the execution time of each statement;
e) cumulative program execution time;
f) current memory use; and
g) the number of cases for which the statement is executed (if the cases exceed one)

```
----      2 OTHER                    0.000     0.000 SECS    0.1 Mb
----      3 OTHER                    0.000     0.000 SECS    0.1 Mb
----      4 OTHER                    0.000     0.000 SECS    0.1 Mb
----     12 ASSIGNMENT X             0.350     0.350 SECS    4.5 Mb  172800
----     14 ASSIGNMENT Z             1.200     1.550 SECS    8.9 Mb  172800
----     16 ASSIGNMENT Y             1.260     2.810 SECS    8.9 Mb
----     28 ASSIGNMENT SLOW          0.000     2.810 SECS    8.9 Mb
----     29 SOLVE INIT SLOW          0.010     2.840 SECS    8.9 Mb
----     23 EQUATION   OBJEQ          2.390     5.230 SECS    9.0 Mb       1
----     24 EQUATION   R              2.210     7.440 SECS   16.5 Mb    1200
----     25 EQUATION   Q              2.360     9.800 SECS   16.9 Mb    1440
----     29 SOLVE FINI SLOW          0.250    10.050 SECS   16.9 Mb
solve
----     29 GAMS FINI                0.230    10.280 SECS   16.9 Mb
----      1 EXEC-INIT                0.000     0.000 SECS    9.0 Mb
----     29 SOLVE READ SLOW          0.040     0.040 SECONDS
----     31 ASSIGNMENT SUMOFVAR      1.780     1.820 SECS    9.2 Mb
----     31 GAMS FINI                0.000     1.840 SECS    9.2 Mb
```

Speeding up GAMS Execution Time
How do I find where excessive time is being used?
Profile

Profile can generate non informative output. Reduce reporting by using PROFILETOL to impose a tolerance on the minimum time used using the syntax

```
OPTION PROFILETOL = 10;
```

This suppresses reporting on statements taking less than 10 seconds (You can use any number in place of the 10)

In invoking profile you set profile equal to a number

option profile=1 or option profile=3

Higher numbers cause profiling within control structures such as loop, if, or for. The number tells how deep to go. If profile is set to

1 GAMS reports timing at loop statement level without details on statements within loops

2 Output gives statement timing for statements not in ifs or loops plus first level of statements within a loop

3 GAMS includes statements timing reports for statements nested within a second loop

Speeding up GAMS Execution Time
Why might a Program be slow and How do I fix?

Set Addressing and References

Referencing out of order slows things down

Y(a,b,c)=X (a,b,c); is faster than Y(a,b,c)=X(b,c,a).

Arrange definitions, calculations, sums, equation references in consistent order

Considering Unnecessary Cases

Consider a calculation of a parameter which is defined over a large of number of sets such as the following can inadvertently cover a huge number of cases

X(A, B, C, D, E) = 5

If each set had 20 members then calculation would o cover 3.2 million cases and would take a long time

Speed can be gained by narrowing attention to good cases employing conditionals
X(A, B, C, D, E) $ GOODCASE (A, B, C, D, E) = 5

Speeding up GAMS Execution Time
Post Solution Report Writing Computations

Often modelers employ post solution report writing calculations. These calculations can involve retrieving and manipulating a lot of data then multiplying it by the optimal variable levels

```
Y=SUM((A,B,C,D,E,F,G),
      (DAT(A)+IT(B,C)+Y(D,E)+W(F,G))*X.L(A,B,C,D,E,F,G))
```

Such calculations will virtually always perform better if one enters a conditional which only causes the data retrieval and calculations to start if the solution variable value is nonzero

```
Y=SUM((A,B,C,D,E,F,G)$X.L(A,B,C,D,E,F,G),
            (DAT(A)+IT(B,C)+Y(D,E)+W(F,G))*X.L(A,B,C,D,E,F,G));.
```

This is a huge time saver for me. It relies on the fact that few variables will be nonzero in a programming model compared to the number of variables present.

Speeding up GAMS Execution Time
Searching

Sometimes code executes too slow to wait

Buffer Problem causes loss of end of profile information when aborting

To find speed problems in such models use

A Smaller model version

Code Isolation

Find last good statement by Screen watching then search for problem

Search strategies

Employ Save Restart to isolate a problematic part

Use Code Deactivation to suppress suspected slow parts until code becomes faster. Then investigate most recently suppressed part until problem is found. De activate by using

$Ontext based binary search

Make things into comments using *

Speeding up GAMS Execution Time
Other Speed ups

Trading Memory for Time

Avoiding repeated time intensive calculations

```
Z=SUM(( CROP, TILLAGE, LANDTREAT, ROTATION ),
    ACREPLANT( CROP, TILLAGE, LANDTREAT, ROTATION)*
    SUM(INPUT, USAGE(INPUT,  CROP)));
```

You can substitute memory for time by calculating numbers that would be used over and over again Here the code is revised by defining a parameter for the input usage sum and substituting i.e.;

```
INPUTUSE( CROP) = SUM(INPUT, USAGE (INPUT, CROP));

Z=SUM(( CROP, TILLAGE, LANDTREAT, ROTATION ),
    ACREPLANT( CROP, TILLAGE, LANDTREAT, ROTATION)*
        INPUTUSE (CROP));
```

Another huge time saver but to need to watch out for dynamic vs. static calculations making sure calculation is repeated when data entered into it changes

Speeding up GAMS Execution Time
Other Speed ups

You can also gain speed by

Increasing Solver Efficiency – Main methods

- Scaling
- Advanced Basis Usage and Starting Points
- Solver Choice
- Avoiding Degenerate cycling

Problem Reformulation

- Changing your model structure