curriculum

COMPARISON OF GAMS,

has the values of parameters with their definitions, and AMPL has the values of parameters in a data file. These programs are easy

and a variable willleave the

zation, the values of the objective function at the optimum were the same for GAMS and MINOS ($1,154.43/day), which was the same as Grossmann's['] result. But AMPL gave a slightly better optimal value ($1,161.34/day). This optimal solution had been reported by the original author of

the problem, Liebman, et al.[12]Grossmann claimed the dif-
ference between the optimal results from his GAMS solution and Liebman's solution was likely due to different default tolerances in MINOS. Also, we have shown that this prob-lem has multiple optimal solutions, and several local maxima have been found by giving different starting points. In the absence of a specified starting point, MINOS executed the problem by setting the variables to zero or to a bound (if it was specified) that was closest to zero and exited when an optimum was located. Without good starting points for most of the variables, MINOS was unable to reach the final maximum objective value. But GAMS found the optimal solution with only one variable initialized, and AMPL was able to reach the final optimal solution with-out the initialization of any variable.

The multi-spindle automatic lathe problem minimized a nonlinear objective function subject to ten nonlinear con-straints. For this optimization problem, GAMS successfully located the global optimal solution from different starting points, or even without specifying a starting point. MINOS and AMPL could locate the correct global optimal solution only when a starting point close to the global optimal solu-tion was given. Otherwise, some sub-optimal solutions were found. Also, when this problem was solved using GAMS with the CONOPT solver, re-scaling of variables and con-straints was required-otherwise the problem could not be solved. When a starting point close to the global optimal solution was specified for the three methods, GAMS and MINOS found the same optimal value (-4,430.088), but AMPL located a slightly higher value (-4,430.005). This illustrates the need for starting points close to the optimum and scaling of variables and constraint equations.

In Table 5, measures of the computation efficiency are given by the total number of iterations, superbasic variables left, and function calls for the eleven problems. MINOS took fewer iterations and function calls than GAMS and AMPL in total and for most problems. This may be significant for large, complicated problems. But creating the MPS me and FORTRAN subroutine for MINOS isit\me consuming and prone to errors. These drawbacks for MINOS may supplant its advantage. For example, some of these optimization prob-lems were assigned to students for homework in an optimi-zation course. A few students solved the problems using MINOS in the time allotted, while all found optimal solu-tions by AMPL and GAMS. Also, they reported that GAMS and AMPL were easier to use than MINOS when starting with no experience with these methods.

All of the problems required well-scaled variables and

224

TABLE 5

Comparison of the Computation Efficiency for Eleven Optimization Problems with MINOS, GAMS, and AMPL

Total of major Totalof minor Total ofsuperbaslc Total
iterations iterations variablesleft functioncalls

MINOS
GAMS
AMPL

317
610
377

32
27
31

1011
1593
1255

62
75
81

constraint equations. 'Scaling is performed by multiplying factors to have the variables and constraints close to a

magnitude of one.[l] Scaling is key to obtaining optimal solutions for problems with widely varying values of the variables and constraint equations. The users manuals describe procedures for scaling.

SUMMARY

Programming and solving standard optimization problems showed that GAMS, AMPL, and MINOS are all effective, and they release modelers from programming optimization algorithms. The comparisons showed that optimization prob-lems are relatively easy to program in GAMS and AMPL, and they offer a choice of solvers and have a presolve phase to reduce model size. In addition, AMPL has features of separate model and data mes, flexible output, and options to run batch operations. GAMS provides a comprehensive out-put summary that is very helpful in detecting model errors, and it is interfaced with more solvers than AMPL now. MINOS could be more robust than GAMS and AMPL, but programming is more difficult. In addition, this is an active area for developments; Floudas describes MINOPT,I13Jan automated mixed-integer nonliner optimizer. Also, GAMS has been extended to use the APROS technique to connect the NLP and MILP in the decomposition of MINLP (Paules and Floudas in [5]).

REFERENCES
1. Brooke, A., D. Kendrick, and A. Meeraus, GAMS: A User's Guide, Release 2.25, The Scientific Press, San Francisco,
CA (1992)
2. Fourer, R., D.M.in

(1981)
9. Schittkowski, K, More Test Exampks for Nonlinear Pro-gramming Codes, Springer-Verlag, New York, NY (1987)
10. Floudas, CA, and I.E. Grossmann,

P{'] :=genl 30 g.n2 20; X:=genJ gas 19.8857 genloil 10.1143
gen2 gas 16.4388 gen20il 3.56123;
Z{'J:= gas 10 oil

Problem Number 11
Mjnimize
Rows 20
Columns 30
Elements 50

MPSfiJe 10

G MINOS MPS FUe for Fuel AUocation Optimization

I

NAME FUELOIL
ROWS
LOlL_AMY
L GAS_AMY

E GENTI
EGENf2

G TPOWER
N PUR_OIL
COLUMNS
XII GENTI 1.0
XI2 GENTI 1.0
X21 GENT 2 1.0
X22 GENf2 1.0
ZI PUR_OIL 1.0
Z2
PI GENTI -1.0
P2 GENTZ -1.0
RES
DEMAND TPOWER 50.0
UP BOUNDOI Z2 10.0
UP BOUNDOI PI 30.0
LO BOUNDOI PI 18.0
UP BOUNDOI P2 25.0
LO BOUNDOI P2 14.0
FR 1NIT1AL PI 24.0
FR INffiAL P2 19.5
ENDATA

TPOWER 1.0
TPOWER 1.0

@ MINOS SPC (Speclfications) FUe for Fuel Allocation
Optimizution

BEGIN FUEL OIL (NLP problem)
,
"
To Minimize the Consumption of Fuel Oil for Fuel Oil Allocation
,

Printlevel I 'OK for smallproblems
Printfreqnency I
Snmmary frequency I

Nonlinearconstraints 2
Nonlinear JacobianVar 6
Nonlinear Objective Var 0

TABLE A3

ScaleOption 2
ENDFUELOILPROBLEM

1 G

Funcon Subroutines for Fuel Allocation Optimization
I

PROGRAM MINOS

lMPUCIT OOUBLEPRECISION(A-H, O-Z)
PARAMETER (NWCORE=30000)
DOUBLEPRECISION Z(30000)
CALL MINOSI(Z,NWCORE)
END