# *Software Notes*

# An Integrated Package for Linear Programming

David J. Pannell*

## 1. Introduction

Batterham (1987) described a computer program which eases entry of mathematical programming data. Users of the program first use a text editor to enter data in a relatively efficient format, which the program then converts into MPS format as required by most mathematical programming solution programs. Alternatively the program can read and convert data from a microcomputer spreadsheet data file.

This note is a description of a microcomputer program which has advantages over each of these procedures. The program, GULP (General, User friendly Linear Programming), is an integrated package of procedures for data entry, matrix printing and model solution. GULP has particular advantages for education and has been used in a number of courses in Western Australia.

## 2. Program Features

### 2.1 Data Editor

One of Batterham's motivations was to reduce the difficulties students have using text editors. This was achieved with a more efficient data format, but entering this data still requires use of a text editor. In GULP, data is entered and edited using a spreadsheet styled editor which has been specifically tailored for linear programming (LP) matrices. Table 1 shows a screen of the editor as it looks after the entry of the example problem given by Batterham. It is laid out like an LP matrix with activity names as column headings, constraint names as row headings, the objective function as the first row and constraint limits (right hand side terms) as the first column of data. Constraint types are indicated next to constraint names. Entering data simply entails using arrow keys to move the cursor to the desired cell and typing in the entry. Names and constraint types can also be entered by positioning the cursor and typing them in.

The program has a number of advantages for data entry. Typing is minimised since each row and column name need only be entered once. (Batterham's format requires the row name to be entered for each coefficient). Typing errors are reduced by the editor ignoring any invalid key strokes. For example it is not possible to enter a non-numeric character in a numeric cell. Users need only ever think in terms of a matrix; they need not convert data from the matrix format in which their problem is specified to another format before data entry. Working with a matrix makes it much more likely that coefficients in the wrong cell will be noticed due to their departure from the visual pattern of entries. The editor includes facilities for deleting, inserting and copying rows, columns and blocks of data. A multiperiod model with a block diagonal structure can be quickly created by marking the block and copying it repeatedly.

### 2.2 Matrix Printer

An option to print all or any part of the matrix is available from a menu within the

* Western Australian Department of Agriculture and School of Agriculture, University of Western Australia. GULP is written in Turbo Pascal. A copy of the shareware version can be obtained by sending $5 ($A10 outside Australia) to Optimal Software, P.O. Box 610, Victoria Park, Western Australia 6100.

editor. A copy of the matrix can also be saved in a file. Column widths and the number of decimal places displayed can be set in the editor and both apply to the screen and to printouts.

*Table 1: The Data Editor*

| HeadyCandl | Limit | Corn | Oats | Soybean | Wheat | Col5 | Col6 |
|---|---|---|---|---|---|---|---|
| Objective | N | 30.00 | 10.00 | 40.00 | 12.00 | | |
| Land | L 100.00 | 1.00 | 1.00 | 1.00 | 1.00 | | |
| JulyLab | L 80.00 | 1.00 | | 2.00 | | | |
| MarchLab | L 100.00 | | 1.00 | | 0.50 | | |
| Row 4 | L | | | | | | |
| Row 5 | L | | | | | | |
| Row 6 | L | | | | | | |
| Row 7 | L | | | | | | |
| Row 8 | L | | | | | | |
| Row 9 | L | | | | | | |
| Row 10 | L | | | | | | |
| Row 11 | L | | | | | | |
| Row 12 | L | | | | | | |
| Row 13 | L | | | | | | |
| Row 14 | L | | | | | | |
| Row 15 | L | | | | | | |
| Row 16 | L | | | | | | |
| Row 17 | L | | | | | | |
| Row 18 | L | | | | | | |
| Row 19 | L | | | | | | |
| Row 20 | L | | | | | | |

Row = 0 Col = 0 Aim = MAX File = DEFAULT.GLP

Press </> for commands

## 2.3 Solution Algorithm

Solution proceeds via a menu item in the editor. The main solution algorithm used in GULP is a "highest step" procedure based on Heesterman (1983). A "steepest ascent" algorithm can also be selected; this option is intended to be useful in courses on the simplex method, particularly when combined with the facility to drop back into the data editor after each iteration to observe changes to the matrix. Students can develop a good understanding of the simplex method without having to spend frustrating hours making mistakes on a hand calculator.

The main algorithm is particularly efficient for problems with many equals or greater than constraints, but is prone to rounding errors in large, poorly scaled problems with highly degenerate initial tableaus (*i.e.* many transfer rows). The problem is reduced by using a maths coprocessor version (for 8087 or 80287) which works to more significant figures (20 compared with 11), but the program is not a competitor with programs such as AESOP for very large problems.

Data is stored in MPS format so, if desired, GULP could be used to enter data for another program. However, the integration of the editor with the solution algorithm has advantages for interactively conducting sensitivity analysis. After printing the solution, GULP re-loads the original matrix and returns to the previous position in the data editor, so that the "what if" questions suggested by the output can be immediately tested.

## 2.4 Limitations

The standard version of the program has a maximum matrix size of 200 columns by 150 rows. Although the limitations of the algorithm make solution of problems much larger than this impractical, the program's editing facilities would still be useful for entering and editing larger matrices for solution by more powerful algorithms such as AESOP. For this reason, a version which allows editing of very large matrices (up to 1500 by 1000) by taking advantage of their sparseness has been developed.

# 3. Availability

GULP is primarily used on IBM compatible microcomputers running PC-DOS or MS-DOS, but versions are also available for less compatible MS-DOS computers (*e.g.* NEC APC III) and Kaypro compatible CP/M computers. Anyone interested in implementing the program on an Apple Macintosh should contact the author.

GULP is not in the public domain but a version with a reduced matrix size is distributed under the related "shareware" system. The shareware version may be freely copied and distributed for evalution and users pay only if they end up using it. This system facilitates very low prices ($150 for a site licence for the full sized version).

# References

BATTERHAM, R. L. (1987), "Data entry programs for mathematical programming models", *Review of Marketing and Agricultural Economics* 55 (2), 178–80.

HEESTERMAN, A. R. G. (1983), *Matrices and Simplex Algorithms*, Reidel, Boston.