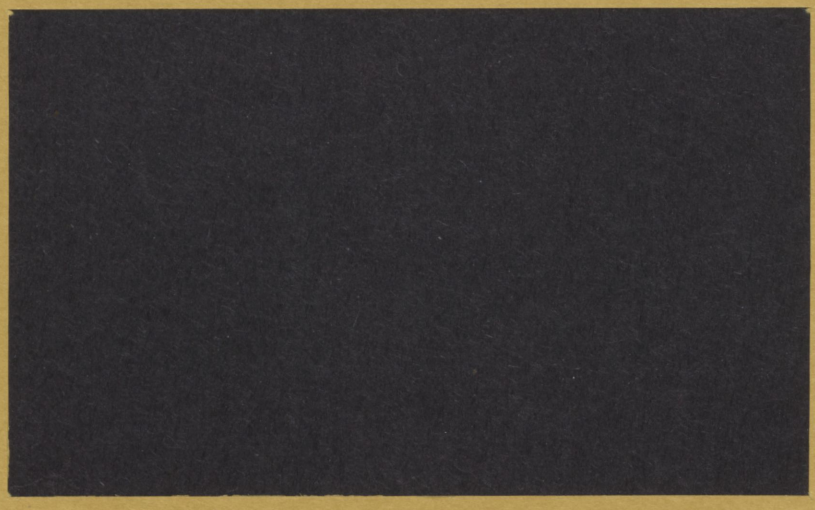
Business

75/10

# SCHOOL OF AGRICULTURAL ECONOMICS AND EXTENSION EDUCATION



ONTARIO AGRICULTURAL COLLEGE
**UNIVERSITY OF GUELPH**
Guelph, Ontario, Canada



# A MINI-COMPUTER LINEAR PROGRAMMING SYSTEM

# FOR SMALL-BUSINESS APPLICATIONS

W.C. Pfeiffer

School of Agricultural Economics and
Extension Education
University of Guelph
August 1976

AEEE/75/10

ISSN 0318-1812

## Introduction

The Desktop Automatic Budgeting System (D.A.B.S.) is a package of computer programs designed to support linear mathematical programming with mini-computers. Because small computers can be obtained for a fraction of their previous cost, it may now be possible for more business firms to afford a computer than ever before. This new machinery makes it possible to further extend scientific tools of management to medium-sized businesses which are often in a position to profit from the information they provide.

Of course, in order for mini-computers to aid business managers in a meaningful way, software (programs) must exist which are tailored to solve problems. Because the mini-computer-advanced programmable calculator offerings are still new, the stock of available software is small relative to that designed for larger equipment. This paper introduces a system of programs for building linear models for management planning which operates entirely on a portable computer without any link-up with a large-scale machine. It is a large system with many features similar to those found in mathematical programming systems that operate on large-scale computers.

The system is written in the BASIC computer programming language. It was designed to be interactive with the user through use of a visual display and keyboard for data entry on a Hewlett-Packard 9830 calculator. Other mini-computers (such as an IBM 5100 or a Wang 2200) can be used to support the system. The machine chosen must allow for storage of

data on tape or disk and have BASIC as its programming language. The system is designed to support linear mathematical programming in a straightforward manner from the viewpoint of two types of users, namely: 1) those who design and build a model, and 2) those who apply it under actual business conditions.

The system presents designers of mathematical programming models with a flexible means of communicating their model's features to the machine. The system provides the opportunity to design concise data entry forms which relate to their model but are written for a non-technical audience. It also allows the user to design a detailed report of results which will be printed automatically at the end of each run. The report feature is included so that results can be output in a form which can be readily understood by a non-technical reader.

Users of LP models supported by the system have a straightforward and uncomplicated method of entering their data into the machine, calling for a solution and obtaining results. Because it ensures that each model it supports has a built-in, user - oriented data input form along with a non-technical printed report, use of the system requires very little or no prior computer training or experience.

The remainder of this paper describes the D.A.B. system in a non-technical manner. Subsequent papers will present technical details of D.A.B.S. extensions and improvements, and example management planning models built using the system.

Readers with prior experience in constructing linear programming models for business application may refer directly to Appendix I for a user's guide to operating the D.A.B. system.

# Chapter I

## The Nature and Objectives of the Desktop Automatic Budgeting System

The use of mathematical programming, linear and non-linear, for modelling the operations of agricultural firms has been made practical through computers. Computers have given management specialists the calculation speed necessary to solve large and complex problems.

Management models, if they are to be practical, must meet several requirements. First, of course, they must clearly address a problem which management faces but cannot deal with by manual means. Secondly, such models must be based mainly on numerical information obtainable in the course of business events. Thirdly, they must be flexible in accomodating changes to their basic data. Fourth, they must operate quickly and be low in cost from two standpoints, namely, 1) labour requirements and 2) computer time (and hence money costs) each time they're used. Fifth, any model designed for management purposes must deliver its answers to management such that relevant details are clearly presented.

Admittedly, these are not the only requirements, but they do serve as general guidelines to model construction for business application. While all five requirements impinge upon model design, the first two are unaffected by machine considerations. The computer equipment employed can, however, greatly effect the realization of the last three guidelines.

Because small computers can now be purchased at prices within the budget of many small to medium-sized business firms, and because they typically require little technical knowledge to operate, they present new

opportunities for management planners. Even though the new mini-computers are categorized as "small" they afford the business-oriented model builder sufficient electronic machine capacity to design realistic simulations. Models based on linear programming are often compared re: the size of their table of coefficients (called a matrix). "The larger this matrix the more complex (and hopefully more realistic) the model" is a rule of thumb which usually applies within reasonable limits. Many business-oriented models of agricultural firms exist which have matrix sizes in the range from 20 rows and 20 columns all the way up to over 200 rows and 200 columns.

It is becoming commonplace to find mathematical models being used in everyday business settings. In agriculture, feed companies use them to blend basic ingredients into finished products in a fashion which minimizes their input costs. Machinery companies have adopted them for assembly line scheduling and parts distribution to dealers. Large farms use them to plan a year's cropping activities and to formulate livestock feeding rations. In agriculture, farm management specialists have become aware of many new possibilities models offer for solving farm problems. Many agricultural research institutions now stress close extension co-operation with farm operators themselves. They agree on the desireability of placing these new tools into the hands of people who can benefit from their use.

The Desktop Automatic Budgeting System has features which cater to the business manager who understands enough of the LP technique to rely on it for planning; but who also wants to minimize the problems of computerizing his particular LP model.

In this system, the matrix of technical coefficients which comprise the model is regarded as a "black box" which only the technical

developer of a particular model deals with under ordinary circumstances. This matrix is filled originally with figures which represent the manager's best idea of each coefficient. These figures are retained by the system. A user need change only those figures with which he disagrees in order to initiate a new solution to the model each time the effect of changing conditions on the simulation is desired.

Before operationalizing a new model, a "data booklet" should be prepared to contain all of the basic numerical information about the business needed for the model. This booklet can be written and organized such that the presentation of information to the system seems logical to the user. The system has a matrix generator program built into it which automatically relates figures in a data booklet to locations in the LP matrix of technical coefficients.

The second major feature of the system is that a report writer program is included which can be instructed to produce results in a form chosen by the user of the model. Again this feature caters to business rather than research applications for the system. The report writing options in the system allow the developer of any model to organize the presentation of results along lines which seem logical and straightforward to the person who will later apply the model to actual business operations.

This approach whereby an LP model is built "from the ground up" to include input and output forms which are intended for the user audience (the business manager) has been used extensively in centres of farm management research. At Purdue University in the United States this generation of management models has been introduced to farm managers through extension workshops. Based on their experience in these workshops Purdue economists

believe that this approach to "harnessing" computer models will gain wide acceptance.[1/]

The system described herein consists of three groupings of computer programs: 1) the "set-up" group, 2) the "utility" group, and 3) the "operating" group. The first two groups of programs set up man-machine dialogues with the developer of a particular LP model. During these dialogues the system obtains specifications for 1) the model's technical coefficient matrix, 2) the input form, 3) matrix generation arithmetic, 4) the final report specifications and 5) the English-language literal notations to be included in the report.

The programs in the third group in turn use these specifications to establish a man-machine dialogue with the person wishing to apply the model. This "operating" group of programs obtains the necessary information from the end-user to initiate a solution. This information consists of figures from the input data booklet which are entered on request by the machine. When the user is satisfied that enough information has been entered, a single command (SOLVE) causes the machine to proceed automatically through the model's mathematical solution with the current figures and generate an updated report. The end-user does not need to know anything about the machine or the system (technologically) to make it work with a pre-developed simulation model. He is merely required to read the data booklet to discover what information the model requires and the type of decisions it is capable of making.

---

[1/] Candler, W. quoted in "A New Hired Hand" motion picture presenting the Purdue Automatic Cropping Budget.

## Chapter II

## Guidelines for Constructing LP Models Using the Desktop Automatic Budgeting System

The system is designed to support linear programming in a manner understandable by persons with little or no prior training in computer usage. It is flexible because it can be driven through matrix construction, LP calculations and report writing by instructions which are external to the computer program itself. These external instructions are provided by the model builder and in essence provide the "guidelines" or "rules" for the system to follow toward achieving a computer solution and producing a "tailor-made" report of the results. At no point in this process is either the model builder or the user required to perform traditional computer programming using any special language such as FORTRAN, BASIC, etc.

There are three sets of "rules" which must be fed into the system by the builder of an LP model. One of these is oriented toward the technical aspects of LP and therefore is entirely provided by the person who designs the LP model. The other two provide the system with information about the format and order of the input data and the form of output presentation, each of which are oriented toward the end-user.

To build an LP model using the system the following steps should be followed before starting a dialogue with the mini-computer:

1) Design the LP matrix on paper according to standard mathematical programming procedures.

2) Identify the coefficients in the matrix which must be obtained from the business enterprise under study.

3) Create a format on paper for an input booklet consisting of "data groups" (with multiple "members") which can be related to these numbers.

4) Create a format on paper for a printed report which will be easily understood by the person applying the LP model (the end-user).

The system is programmed to automatically generate a Basic Feasible Solution to any Linear Programming problem within the size limit determined by the memory capacity of the mini-computer. This means that the canonical form for LP which requires all inequalities to be resolved prior to the operation of the Simplex[2/] method need not be entered from the keyboard. Rather, only those matrix elements corresponding to the LP problem expressed as:

MAX: C'X

SUBJECT TO: $AX \leq b$

WHERE ALL x in $X \geq 0$

need to be entered through the mini-computer keyboard. This feature reduces the amount of information required from the LP model builder. In common LP terminology this means that the system will automatically generate the necessary slack, surplus or disposal activities as required depending on the characteristics of the model.

The following example demonstrates this feature. Consider a small matrix with only three activities and four constraints as:

---

2/ Dantzig, G.B., Computational Algorithm of Revised Simplex Method, Rand Report, RM-1266, October 26, 1953.

| | Grow 1A Corn | Grow 1A Soybeans | Grow 1A Barley | Constraint Type | RHS |
|---|---|---|---|---|---|
| Returns (OBJ) | $60 | $49 | $51 | | |
| Land | 1 | 1 | 1 | ≤ | 300 |
| Labor | 2.1 | 2.0 | 1.9 | ≤ | 1000 |
| Capital | 180 | 160 | 140 | ≤ | 10,000 |
| Field hrs. | 1.8 | 1.9 | 1.6 | ≤ | 800 |

The canonical (mathematically complete) form would require a tableau with four additional columns to handle the slack variables on the four constraints. This would appear as:

| | Grow 1A Corn | Grow 1A Soyb. | Grow 1A Barley | Slack Land | Slack Labour | Slack Capital | Slack Field hrs. | RHS |
|---|---|---|---|---|---|---|---|---|
| Returns (OBJ) | 60 | 59 | 51 | 0 | 0 | 0 | 0 | |
| Land | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 300 |
| Labor | 2.1 | 2.0 | 1.9 | 0 | 1 | 0 | 0 | 1000 |
| Capital | 180 | 160 | 140 | 0 | 0 | 1 | 0 | 10,000 |
| Field hrs. | 1.8 | 1.9 | 1.6 | 0 | 0 | 0 | 1 | 800 |

where all inequalities have been converted to equalities to establish the complete objective function and a set of four simultaneous linear constraint equations.

The system requires machine memory space for the completed tableau of five rows and eight columns. The system always considers the first row as the objective function and the last column as the Right Hand Side ("b") vector of resource levels. Because the slack, surplus and artificial activities will always be automatically generated by the system, it frees the LP model builder from the tedium of entering these vectors explicitly (a significant task in large models).

Because of the automatic tableau generation features of the system the following matrix layout format is suggested:

| | Col. 1 | Col. 2 | Col. 3. . . . . Col. n | Row Type | RHS |
|---|---|---|---|---|---|
| Row 1 | | | | | |
| Row 2 | | | | | |
| Row 3 | | | | | |
| . | | | | | |
| . | | | | | |
| . | | | | | |

By following this format each row and column in the matrix is assigned a number. Row 1 will be considered by the system as the objective function (hence no RHS element). The last column will be considered by the system as the RHS to the LP equation set.

The "Row Type" column is used here for the L.P. model builder to keep a record of the direction of the inequality on each constraint in the basic design. The system can accomodate L.P. problems involving mixed inequalities. It expects to see all maximum restrictions ($\leq$ type rows) grouped starting with row 2. Equality restrictions (= type rows) for L.P. transfer rows are expected to be grouped together after the maximums. Minimum restrictions ($\geq$ type rows) should be grouped at the bottom of the tableau. The system will ask the model designer for the number of rows in each group when the model is set up for the first time. It will then automatically augment the basic matrix with columns for the appropriate numbers of slack, artificial and surplus variables. During this process the RHS vector is moved to the right in the tableau so that it remains the last column in the internal canonical form tableau.

Once the model has been laid out in matrix form, the next step is to design a paper form to contain all of the model's variable input information. An input form in this context could be anything from a worksheet to a multi-page booklet. In any case its purpose is to contain the numbers which go into the LP matrix which are subject to change from time to time. Any information in the model (such as technical coefficients in the $A_{i,j}$ matrix, elements of the objective function or right hand side) which the designer wishes to leave up to the user's discretion to change should be assigned a position in the input form.

Additionally, the coefficients actually used in the original matrix should also be included in the input form as "example entries", "built-in figures", or some other designation alongside the blanks to be filled in by the user of the model. This makes the input form somewhat "self-prompting" because it suggests a possible entry to the user in every position. Oftentimes business planning models contain technical information outside of the normal record keeping system for a firm. By having the "built-in" figure appear in the input form, the form itself becomes a record keeping device for the model. Input forms should contain generous footnoting and explanation of the sources for technical data as a further aid to the person who will use the model for planning purposes.

Because the system retains these built-in figures in its magnetic data file for any model, the end-user does not have to enter a complete set of figures each time the model is run. The user is required to enter only the figures specific to his current business situation which do not coincide with the suggested figures.

Suppose a small LP model was constructed using this system to simulate the production of feed grains on Southwestern Ontario farms. The first page of the input form might appear as:

Data Form for Feed Grain Production Model

page 1

| Information Group I | Example Situation | Your Situation |
|---|---|---|
| Total arable land area (acres) | 500 | (1)_____ |
| Total rentable land area (acres) | 300 | (2)_____ |
| Number of full time men | 2 | (3)_____ |
| Market price for grain corn ($/bu.) | 3.00 | (4)_____ |
| Market price for soybeans ($/bu.) | 4.65 | (5)_____ |

In all likelihood a farmer entering figures into this form would not change everything each time to generate a current solution from the model. For example, "market prices" would probably change frequently whereas "total land area" would probably remain constant for many runs of a production planning model for any given cash crop farm.

This approach to designing an input form helps relieve the burden of data entry for the user. In order to preserve this feature for users of LP models supported by this system, designers should be careful to limit the scope of their basic figures. For example, the structure of an LP matrix to simulate cash grain production on a 300 acre farm may be exactly the same as for a 600 acre farm. However, the figures in each model may differ greatly. In such cases different versions of the basic model could be stored in the system each containing the appropriate "built-in" figures. Likewise, separate input forms could be constructed for the various versions (eg. 300, 600 and 1,000 acre cash grain farms) to make the built-in figures in each representative of the situation being simulated. Even though the matrix might be logically identical, each version of the model would be implemented separately on the mini-computer using this system.

One specific restriction relates to the grouping of figures in an input form for a model supported by the system. Because of technical reasons, this system expects to receive input in groups of 5 entries or less. Therefore the input form for a model requiring 60 entries, for example, would contain 12 data groups. The system is programmed to ask for data to be entered according to its "group #" and its "member number". For example to change the market price of soybeans in the above example to $5.00 per bushel, the user would enter the figures "1" (group 1) and

"5" (member 5) and then the updated figure (hence 3 numbers typed as: 1, 5, 5.00).

Having accomplished the above steps, namely: 1) designing the LP matrix on paper and 2) designing the input form and filling it with the requisite "built-in" information, two more subordinate tasks follow. These are:

1) List the primary matrix setup rules on paper which relate the numbers in the input booklet (by group and member) to cells in the LP matrix (by row and column), and
2) List the secondary matrix setup rules on paper which instruct the system to make whatever additional entries (constants, etc.) that are necessary to complete the logical structure of the non-canonical LP table.

Primary Matrix Setup Rules

Each figure in the input form must have a "rule" governing its placement by the system into the internal LP matrix. A model with 60 numbers in its input form would require 60 "primary rules" to set up its internal LP matrix. Each rule merely consists of four integers which relate in order to 1) Data group, 2) Member within group, 3) Matrix row, and 4) Matrix column. Thus the rule 1,3,4,7 would instruct the system to relate the third number in the first data group of the input form to the cell in the matrix where row 4 intersects column 7. Once these rules are entered during the Setup Phase for any model they remain on tape. These rules are coded by the designer and end-users of LP models supported by this system need not know they exist and are not made aware of them by the system.

Secondary Matrix Setup Rules

Most LP models require extra elements in the matrix over and above those contained in the input form. For example transfer rows are often done with a technical coefficient at one end and a constant (usually 1) at the other. For example the transfer row between a 1 acre corn growing activity and a 1 bushel corn selling activity might appear in an LP matrix as:

| | Grow 1 Acre of Corn | Sell 1 Bushel of Corn | Row Type | RHS |
|---|---|---|---|---|
| Corn Transfer | -90 | 1 | = | 0 |

The -90 coefficient represents the corn growing activity's per acre yield (in bushels) which may change from time to time. This figure would probably be included in the input form and be identified to the system with a Primary Matrix Setup Rule as discussed previously. The "1", however, merely completes the link (and resolves the units) between the growing and selling activities. This element of the matrix is not subject to change and therefore, would probably not appear in the input form. Its existence would be made known to the system through a Secondary Matrix Setup Rule.

These rules have the general form of seven integer numbers meaning: (1) Matrix Row, (2) Matrix Column as a function of (3) Matrix Row, (4) Matrix Column via an (5) Operator Code with (6) Matrix Row, (7) Matrix Column. This means that one element in the matrix can be related to another element via an operation code. There are five

operation codes recognized by the system:

(1) "1" = add

(2) "2" = subtract

(3) "3" = multiply

(4) "4" = divide

(5) "5" = equals

Suppose element (1,3) in the matrix were to be set equal to the difference between element (9,11) and element (6,7). The Secondary Rule to accomplish this would read: 1,3,9,11,2,6,7. For another example, suppose element (4,5) were to be duplicated in cell (7,8) (the same number entered into the matrix in two places), the Secondary Rule for this would read: 7,8,7,8, 5,4,5. That is to say "Element (7,8) is a function of itself being set equal to element (4,5)". This notation seems a bit redundant in this case but the general seven code rule format is preserved. The cell duplicating rules (using the equals operator) usually exhibit this characteristic.

If the last (7th) integer of the rule is missing, the system assumes that the 6th element refers to a constant rather than a matrix row designation. Therefore constants (such as the "1" on the transfer row above) can be placed in the matrix using this abbreviated form of Secondary Rule. Suppose element (6,8) were to be a constant namely: "1"; the Secondary Rule for this would appear as 6,8,6,8,5,1. Notice that only 6 integer numbers are used in this case. If element (6,8) were to contain the constant 45 instead, the Secondary Rule would have been 6,8,6,8,5,45.

The format of the Secondary Matrix Setup Rules is sufficient to

do arithmetic on matrix elements (and enter constants). The rules instruct the system to perform any operations to cells of the basic LP matrix over and above the task of transferring numbers from the input form.

When the end-user of a model initiates a solution on the machine, the creation of the basic LP matrix in the mini-computer's memory is a three step process that proceeds automatically. First the system distributes the current set of coefficients from the input form (actually the built-in figures plus any updated items entered by the end-user during each run into the appropriate cells in the matrix. This phase is controlled by the Primary Matrix Setup Rules. Second, the system performs any ancillary arithmetic on these coefficients and distributes any constants into their appropriate matrix cells. This phase is controlled by the Secondary Matrix Setup Rules. Last, the system augments the basic LP matrix with the necessary columns to resolve all inequalities. This phase completes the matrix for solution by the Simplex algorithm. Automatically, the system will proceed to search for an optimal solution.

Basic Data Entry

As mentioned previously, the system retains all of the coefficients included in the original LP matrix established during the setup phase. The best way to proceed on this is to merely list all backup data on paper which will go into the system and become the "built-in" or "default" data to be permanently "memorized". Later when this information is to be actually entered into the machine during the setup phase it will ask for coefficients in the same order in which the Primary Matrix Setup Rules were put into the system. This task of entering basic data only needs to be done once

by the person responsible for designing a particular model. The method of entering individual elements from the keyboard is as straightforward as putting numbers into any desk calculator.

Designing the Output Report

The system includes features to support the systematic presentation of results in a manner tailor-made to each model it is used to support. This means that the designer of an LP model can put together a printed report format that includes literal explanations for each number printed. The system creates a matrix of numbers consisting of two rows and as many columns as included in the matrix for any model. The elements of the first row are the levels of all activities in the model at the optimal solution. The second row contains all relevant shadow prices pertaining to constraints which are operating in the optimal solution.

The designer of a model should proceed through two final steps to complete the specifications for an LP model to be supported by this system, namely:

1) Design the desired Output Report on paper and give each line of this report (including lines of blanks for spacing) a number starting with 1 at the top of the page, and
2) List the reporting rules which relate each number desired from the final LP table (by row and column) to the line (by line number and horizontal spacing) of the printed report where it should appear.

When designing a printed report, a layout sheet should be constructed on which all lines are numbered (starting at 1) and a column is reserved on the right margin labelled "# of rules". The literal

notations for the report are then printed or typewritten onto this worksheet. A portion of the Report Layout Sheet for the example cash-grain model mentioned above might appear as:

Report Layout Sheet

| Line | | # of Rules |
|---|---|---|
| 1. Cash Crop Model - Corn vs Soybeans | | 0 |
| 2. Line of blanks for spacing | | 0 |
| 3. Total Acres of Grain Corn | xxxx | 1 |
| 4. Line of blanks for spacing | | 0 |
| 5. Total Acres of Soybeans | xxxx | 1 |
| 6. Line of blanks for spacing | | 0 |
| 7. Total Acreage Cropped | xxxx | 1 |

Report Generation Rules

Each line of the report may or may not contain a number from the computer solution. In the example above, line 1 is merely the title for the report and contains no other information. Therefore, note that it does not have any rule for printing numbers associated with it (0 is entered under " of rules" for line 1). The same holds true for lines 2, 4 & 6 which are blank lines used for vertical spacing. Lines 3 & 5, however, are to contain a number taken from the computer solution. They are denoted by "xxxx" on the Layout Sheet. Suppose in our example they are taken directly from the solution vectors in memory and printed. Thus one "Report Generation Rule" is required in each case as indicated.

These rules, like the Secondary Matrix Setup rules described above, consist of seven integers referring to (1) line, (2) horizontal spaces left of print field, as a function of (3) line, (4) horizontal spaces left of print field via an (5) operation code with matrix element given by (6) row and (7) column. That is to say, the number printed on any line can be expressed to the system as a function of a number printed on another line, a matrix element, or both. In the sample case of printing an activity level from the optimal solution (eg. "Total acreage of grain corn") the rule would make use of operation 5 (equals). Suppose that this activity were in column 3 in the model. Then its level at optimal would be held in element (1,3) by the system. To print this number on line 3 with 5 spaces between it and the literal field, therefore, would require the rule: 3,5,3,5,5,1,3. Or "print a number on line 3, five spaces from the end of the literal field, which is defined as itself equals matrix element 1,3." In this case (using the "equals" operator) the rule appears redundant. However, like the Secondary Matrix Setup rules, the Report Generation Rules allow the same flexibility in performing arithmetic on elements from the solution matrix before they are printed.

This feature can be used to advantage when more complex post-optimal arithmetic is desired before a resulting number is printed. If, for example, one wished to add a matrix element to a number already printed (as on line 7 of the example above), the rule might appear as 7,5,5,5,1,1,3. That is to say; "print a number on line 7, five spaces from the literal field, which is defined as the number on the printout line 5 (which was also five spaces over from the literal field) plus matrix element (1,3). Matrix element (1,3) you will remember was the

origin of the printout number on line 3 described above.

By stacking many rules together for one printed number, the system can be made to perform chains of arithmetic on many intermediate values prior to actually printing the resulting number in the report. Both sets of report generation information (the Literal Notations and the Report Generation Rules) are entered during the Setup Phase by the designer. They are kept on file by the system and are used automatically each time the model is run. The user of the model is not made aware that they exist. The output report is the only information which appears after the mini-computer has finished a run.

After the design steps described above have been completed on paper, the system's set-up routines are used to transfer the five information groups onto a magnetic tape cassette for long term storage in machine readable form. Feeding information into the system is much like using a typewriter with the added feature that the machine prompts for the next entry throughout the session. The set-up routines specifically ask for each item to be entered from the keyboard.

When all five pieces of information, namely:

1) Primary Matrix Setup Rules
2) Secondary Matrix Setup Rules
3) Backup Data
4) Report Literals
5) Report Generation Rules

have been successfully entered, the set-up routines proceed to further process the information and store it on the cassette tape which contains the system's LP analysis programs (the operations group). It is this

tape which the end-user (i.e. the business manager) uses repeatedly together with the input booklet when putting the model to work under his control and with his specifically updated data elements.

Over time as a firm acquires a number of separate LP models which simulate various aspects of the business, the system's group of utility programs can conveniently store each one permanently on an archival cassette tape. Like the setup and operational routines, these programs also prompt the operator for various types of information (such as the name of the model being stored or retrieved). Therefore, no prior computer knowledge is required. The utility programs can automatically maintain a permanent record of all necessary information about many models on a single back-up tape. The advantage of this is that only one copy of the system's operations tape is needed to support a number of separate LP planning models.

Because of its nature, the system insulates end-users of LP models from many technical details which are necessary to solving them with a computer. Furthermore, it ensures that LP models will be designed with both an input form and an output report which are tailor-made to communicate with end-users.

The system can support varied types of LP models without re-programming. It is designed to be controlled by external instructions. In the past many LP models which were designed for business application needed to be embedded in a specific computer program to generate the matrix from special input forms and to print the report according to a special format. This mini-computer based system combines programs which are flexible. They do not change from one LP model to another. The sets of external

instructions change instead. It is hoped that these instructions (or rules) are easier to code than a computer program. For this reason and the fact that mini-computers are often low in cost relative to accessing large systems for small business applications, the Desktop Automatic Budgeting System should make it possible for more businesses to afford computer modelling.

Detailed, step-by-step instructions for using the system are illustrated in Appendix I.

# APPENDIX I

## D.A.B. System Operating Instructions

The following operating instructions pertain to the version of the D.A.B. System which runs on the Hewlett-Packard - Model 9830 mini-computer.

## D.A.B. System Operating Instructions

### Section I - Setup Phase

Step 1 : Turn machine and printer ON and insert the D.A.B. System's Setup Tape into the cassette tape carrier in the machine frontpanel (closing the carrier door after tape cassette is inserted).

Step 2 : Type LOAD2 and press the EXECUTE key.

Step 3 : Press the RUN key and the EXECUTE keys in sequence. The machine will display "D.A.B. SYSTEM - SETUP PHASE" and halt. The operator should take a moment to make sure all paper forms containing specifications for the model are handy. When ready type GO and press the EXECUTE key. The machine will then display "WHAT IS THE NAME OF THIS MODEL?." The operator should reply by typing the name of his choice ($\leq$ 10 letters) and press the EXECUTE key.

The machine will then display "ARE THE RULES FOR DATA ON TAPE?." If the operator has previously stored the Primary Matrix Setup Rules on the cassette tape in another session, he may reply YES and press the EXECUTE key. The machine will then retrieve these rules from the tape and proceed to the corrections part of Step 5.

Otherwise the machine proceeds to the next step.

Step 4 : The machine will begin by asking three questions of the user. Each should be answered by typing the appropriate number and pressing the EXECUTE key. The three questions appear as follows:

1) HOW MANY DATA GROUPS TO BE ENTERED?

2) HOW MANY MEMBERS/GROUP?

3) WHICH GROUP BEGINS THIS SESSION? (When starting from the beginning the response should be 1. However, the system includes a feature which allows the operator to enter Primary Matrix Setup Rules in Batches if desired).

The machine will then display "ENTER RULE n FOR GROUP m." Answer this by typing the 4 integers representing a Primary Matrix Setup Rule (e.g. 1,3,4,7) and press EXECUTE. This repeats until n * m rules have been entered.

Step 5 : When all Primary Rules have been entered the machine will display "ONE MOMENT PLEASE" and then store the rules on the cassette tape. It will then display "ANY CORRECTIONS?." If the operator answers by typing YES and pressing the EXECUTE key, the machine will display "GROUP AND MEMBER" (if the answer is NO, the machine jumps to Step 6). Answer this with 2 integer numbers (e.g. 1,3 representing Group 1 - Member 3) separated by a comma and press EXECUTE.

The machine will then display "RULE FOR THAT LOCATION." Answer this by typing the 4 integers representing a Primary Matrix Setup Rule (e.g. 1,3,4,7) and press EXECUTE.

The machine will then display "ANY MORE CORRECTIONS?." If answered YES the machine will query for another rule. If

NO, the machine will update the Primary Rules on the casette tape.

Step 6 : The machine will then ask "IS THE BASE DATA ON TAPE?". If the operator answers by typing YES and pressing the EXECUTE key, the machine will load data already stored on the tape from a previous session and jump to Step 7. If the operator answers by typing NO and pressing the EXECUTE key, the machine will display "ENTER n MEMBERS OF GROUP m." The operator then types the n numbers (separated by commas) and presses the EXECUTE key.

Step 7 : After the base (default) data has been loaded in Step 6 the machine will display "ANY CORRECTIONS?." If the user answers NO and presses the EXECUTE key, the machine proceeds to Step 8. If YES the machine will display "ENTER GROUP # AND MEMBER #." The operator then answers with the appropriate 2 integers (separated by a comma) and presses EXECUTE. The machine then displays "ENTER VALUE" to which the user responds with the appropriate number and presses EXECUTE.

This process repeats until the operator answers NO to the call for corrections (above). At this time the machine stores the updated data on the cassette tape.

Step 8 : The machine displays "ARE SECONDARY RULES ON TAPE?" if the operator answers YES and presses the EXECUTE key, the machine proceeds to Step 9. Otherwise the machine displays "RULE?" to which the user responds with a 7 integer Secondary Matrix Setup Rule (integers separated by commas) and presses the EXECUTE key.

This sequence repeats until the operator answers with FINISH instead of a 7 integer rule. The machine then stores the rules which were entered on tape, informs the operator how many rules were stored and proceeds to the next step.

Step 9 : After the Secondary rules have been loaded in Step 8 (above) the machine displays "ANY CORRECTIONS?". If the operator answers with NO and presses the EXECUTE key, the machine proceeds to Step 10. Otherwise (i.e. answer YES) the machine displays "RULE # ?" to which the user responds with a 7 integer Secondary Matrix Setup Rule (integers separated by commas) and presses the EXECUTE key. The machine then stores the corrected rule on tape and repeats the sequence.

The operator can terminate this step by entering FINISH instead of a 7 integer rule.

Step 10: The machine then asks for the number of greater-than-or-equal type of constraints ($\geq$) and the number of equality type of constraints (=) which the model contains by displaying "# >s & # = 's". The operator should respond with the appropriate 2 numbers (separated by commas) and press EXECUTE (note the system deduces the number of $\leq$ type constraints automatically).

The machine then generates a printed report of all the information entered thus-far and prepares to transcribe the information onto the D.A.B. System's operations cassette tape.

The machine displays "MOUNT THE OPERATIONS TAPE" and halts. The operator must then 1) remove the Setup cassette from the front-panel tape carrier and replace it with the Operations cassette closing the door after it is seated in the carrier and

2) type OK and press the EXECUTE key. The machine then stores information on the Operations cassette tape and displays "MOUNT THE SETUP TAPE." The operator then reverses the above tape exchange and types OK and presses the EXECUTE key.

The machine repeats this cycle three times and then proceeds to Step 11.

Step 11: The machine displays "ARE REPORT RULES ON TAPE?". If the operator replies YES the machine proceeds to Step 12. Otherwise the machine displays "HOW MANY LINES IN THE REPORT?". The operator should type the total number of report lines (including lines of blanks for vertical spacing) and press the EXECUTE key.

The machine then begins a sequence of questions for each report line as follows:

1) It displays "LITERAL FOR LINE n?" to which the operator replies with the string of characters (or blanks $\leq$ 32) for printing on report line n and presses the EXECUTE key.

2) The machine immediately requests that the operator verify the line by displaying "ANY CORRECTIONS?." If the line is okay and requires no last-minute changes the response would be NO and press EXECUTE. If YES is replied the machine asks for the line to be entered again (NOTE: This is the only place that corrections can be made to literal strings intended for the printed report).

3) The machine then displays "HOW MANY RULES FOR LINE n?" to which the operator replies with the appropriate integer and presses the EXECUTE key.

4) The machine then displays "RULE m FOR LINE n?". The

operator then enters a 7 integer Report Generation Rule (integers separated by commas) and presses the EXECUTE key.

When all report literal lines and all their Report Generation Rules have been entered the machine displays a verification count on the total number of rules entered as "REPORT HAS m RULES." It then proceeds to store the report writing information (literals and rules) on the cassette tape.

Step 12: The machine displays "ANY CORRECTION TO REPORT RULES?." If the answer is NO, the machine proceeds to the Final Setup Step. Otherwise it goes on to display "RULE # ?." The operator should reply with the sequence number of the rule to be changed and press the EXECUTE key. The machine then fetches the old rule from the cassette tape and displays "RULE?." The operator should reply with the corrected 7 integer rule (integers separated by commas) and press the EXECUTE key. The machine then asks for immediate final verification of the re-entered rule by asking "OK TO RELEASE." If NO, the machine allows the operator to re-enter the rule again. If YES, the machine stores the new rule on cassette tape.

The machine then displays "MORE CORRECTIONS?." If YES, this step is cycled again. If NO, the machine proceeds to the last step in the Setup.

Final Setup Step: The machine produces a printed report of the report information (literals and rules) and then displays "MOUNT THE OPERATIONS TAPE" and halts.

The operator should then remove the D.A.B. System's Setup Tape from the frontpanel cassette carrier and replace it with the Operations Tape. After closing the frontpanel tape carrier door, the operator should reply OK to the machine and press the EXECUTE key.

After the machine has finished transferring a data file to the Operations Tape it will display "REMOUNT THE SETUP TAPE." The procedure here is the reverse of that above and the operator should type OK and press the EXECUTE key after changing the tapes. This sequence may repeat several times depending on the length and complexity of the printed report which the D.A.B. System is being instructed to produce.

At the end of this sequence of transferring information to the Operations Tape, the machine will display "END OF SETUP SESSION." At this time the entire set of specifications for the LP model has been received, edited and transferred to the Operations Tape. The operator should shut-down the mini-computer as follows:

1) Press REWIND and wait for the Setup Tape to be rewound,
2) Remove the Setup Tape from the frontpanel carrier and
3) Turn both the machine and printer OFF.

Section II - Operation Phase

There are only four machine steps necessary to operate an LP model which has been stored on the Operations Tape.

Step 1 : Turn both the H-P 9830 and its printer ON and load the Operations Tape into the frontpanel cassette tape carrier.

Step 2 : Type LOAD 1 and press the EXECUTE key. When the tape has stopped, press the RUN and EXECUTE keys in sequence. The machine will pause to initialize itself and then display "DESKTOP AUTOMATIC BUDGET SYSTEM" and begin a fast forward tape search. After about a 30 second time interval the machine will display "WHAT IS THE NAME FOR THIS RUN?." The operator should reply with the name that he wishes to see printed on the final report (20 characters or less) and press the EXECUTE key.

The machine will then cycle through a series of displays designed to remind the operator to have the data input booklet for the model handy, etc. The machine will then display "ARE YOU READY TO PROCEED?." If the operator replies NO, the machine will repeat the cycle of displays.

Step 3 : As soon as the operator replies YES to the question above and presses the EXECUTE key, the machine will display "GROUP # AND MEMBER # ?." The operator should begin to enter information from the input booklet by typing the appropriate 2 integers (separated by commas) which represent the location of the first data item to be changed from the value included in the booklet as built-in (or default).

When the EXECUTE key is now pressed the machine will display

"WHAT IS THE ENTRY VALUE?" and halt. The actual updated figure should be entered and the EXECUTE key pressed to resume. The machine then repeats this step for another entry.

To terminate the entry of updated figures, cause the machine to proceed to solve the LP problems and print the final report, the reply should be SOLVE instead of the 2 integers representing the data booklet location of another figure.

Step 4: After a several minute interval (during which the machine searches for an optimal LP solution) the machine will begin printing the final results. During this operation the cassette tape drive becomes quite active. When the report is finished the machine will display "ANOTHER RUN?." If the operator answers YES the machine re-initializes and begins again at Step 2 (above). If NO, the D.A.B. System terminates. The operator should then do the following to shut-down the Operations Phase:

1) Press the REWIND and wait for the cassette tape to be rewound at which time the cassette drive stops;
2) Remove the Operations Tape from the frontpanel cassette carrier and
3) Turn both the H-P 9830 and the printer OFF.

## BIBLIOGRAPHY


Ackoff, R.L., "The Development of Operations Research as a Science," Operations Research, June 1956, pp. 265-95.

Allendoerfer, C.B., and Oakley, C.O., Principles of Mathematics, McGraw-Hill Book Company, New York, 1955.

Arrow, K.J., Hurwicz, L., and Uzawa, H., Studies in Linear and Nonlinear Programming, Stanford University Press, Stanford, 1958.

Arrow, K.J., Karlin, Samuel, and Scarf, Herbert, Studies in the Mathematical Theory of Inventory and Production, Stanford Mathematical Studies in Social Science 1, Stanford University Press, 1958.

Baumol, W.J., "Activity Analysis in One Lesson," American Economic Review, December 1958, pp. 837-73.

Beneke, R.R., and Winterboer, R., Linear Programming Applications in Agriculture, Iowa State University Press, Ames, Iowa, 1973.

Boulding, Kenneth, and Spivey, W. Allen, Linear Programming and the Theory of the Firm, New York: The MacMillan Company, 1960.

Bright, J.R., "Does Automation Raise Skill Requirements," Harvard Business Review, July-August 1958, pp. 85-98.

Browne, E.T., Introduction to Determinants and Matrices, the University of North Carolina Press, Chapel Hill, 1958.

Carroll, T.H., "Toward a Liberal Education for Business," California Management Review, Spring 1959, pp. 73-8.

Castle, E.N., Becker, M.H., and Smith, F.J., Farm Business Management, 2nd ed., Collier-MacMillan Canada Ltd., Toronto, Ontario, 1972.

Churchman, C.W., et al., "Economics and Operations Research: A Symposium,' Review of Economics and Statistics, August, 1958, pp. 195-229.

Dantzig, George., Linear Programming and Extensions, Princeton, N.J.,: Princeton University Press, 1963.

Dantzig, G.B., "Application of the Simplex Method to a Transport Problem," Activity Analysis of Production and Allocation, T.C. Koopmans (ed.), John Wiley and Sons, New York, 1951, pp. 359-73.

Dantzig, G.B., Computational Algorithm of Revised Simplex Method, Rand Report, RM-1266, October, 1953.

Davis, Chandler, "Linear Programming and Computers," Computers and Automation, July and August 1955, pp. 10-17 and pp. 10-16.

Dorfman, Robert, "Mathematical, or 'Linear,' Programming: A Non-Mathematical Exposition," American Economic Review, December, 1953, pp. 797-825.

Dorfman, Robert, Application of Linear Programming to the Theory of the Firm, University of California Press, Berkeley, 1951.

Dorfman, Robert, Samuelson, P.A., and Solow, R.M., Linear Programming and Economic Analysis, McGraw-Hill Book Company, New York, 1958.

Gass, S.I., Linear Programming: Methods and Applications, McGraw-Hill Book Company, New York, 1958.

Gass, S.I., Linear Programming, 2nd ed., New York: McGraw-Hill Book Company, Inc., 1964.

Hadley, G., Linear Programming, Reading, Mass.: Addison-Wesley Publishing Company, Inc., 1962.

Halsbury, The Earl of, "From Plato to the Linear Program," Operations Research, August 1955, pp. 239-54.

Heady, E.O. and Wilfred Candler, "Linear Programming Methods," Iowa State University Press, 1958.

Hillier, Frederick, S., and Lieberman, Gerald, J., Introduction to Operations Research, San Francisco: Holden-Day Inc., 1967.

Koopmans, T.C., Activity Analysis of Production and Allocation, Cowles Commission Monograph No. 13, John Wiley & Sons, New York, 1951.

Kuhn, H.W., and Tucker, A.W. (eds.), Linear Inequalities and Related Systems, Princeton University Press, Princeton, 1956.

Kuhn, H.W., Scheduling of Petroleum Refinery Operations, Harvard Economic Studies No. 48., Harvard University Press, Cambridge, 1956.

Newman, P., "Some Calculations on Least Cost Diets, using the Simplex Method," Bulletin of the Oxford Institute of Statistics, September 1955, pp. 303-20.

Paull, A.E., and Walter, J.R., "The Trim Problem An Application of Linear Programming to the Manufacture of Newsprint Paper," Econometrica, July 1955, abstract, p. 336.

Riley, Vera and Gass, S.I., Linear Programming and Associated Techniques, A Comprehensive Bibliography of Linear, Nonlinear, and Dynamic Programming, Operations Research Office, The Johns Hopkins University, Chevy Chase, Md., 1958.

Spivey, W. Allen, Linear Programming, New York: The MacMillan Company, 1963.

Spivey, W. Allen, and Thrall, Robert, Linear Optimization, New York: Holt, Rinehart and Winston, Inc., 1970.

Teichroew, Daniel, An Introduction to Management Science, New York: John Wiley & Sons, Inc., 1964.

Vazsonyi, Andrew, Scientific Programming in Business and Industry, John Wiley & Sons, New York, 1958.

Wiener, Norbert, Cybernetics, John Wiley & Sons, New York, 1948.