HONEYWELL EDP

SOFTWARE MANUAL

SERIES 200

LINEAR PROGRAMMING SYSTEM H (BASIC)

GENERAL SYSTEM: SERIES 200/OPERATING SYSTEM - MOD 1

SUBJECT: Programming and Operating Procedures for Linear Programming System H (Basic), the Revised Simplex Algorithm.

SPECIAL INSTRUCTIONS: The reader is assumed to be familiar with the software manual Fortran Compiler D, Order No. 027.

DATE: May 27, 1966

FILE NO.: 123.8305.001H.0-377*

8945
5566
Printed in U. S. A.

*When ordering this publication please specify Title and Underscored portion of File Number.

Copyright 1966
Honeywell Inc.
Electronic Data Processing Division
Wellesley Hills, Massachusetts 02181

HONEYWELL EDP

SOFTWARE BULLETIN

SERIES 200

LINEAR PROGRAMMING SYSTEM H (BASIC) ADDENDUM #1

GENERAL SYSTEM: SERIES 200/OPERATING SYSTEM - MOD 1 (TAPE RESIDENT)

SUBJECT: Corrections to the manual entitled Linear Programming System H (Basic), dated May 27, 1966.

SPECIAL INSTRUCTIONS: The attached pages should be used to replace the corresponding pages in the manual. They will be incorporated into the next edition of the manual.

DATE: June 30, 1966

FILE NO.: 123.8305.001H.0-377*

9043
5666
Printed in U.S.A.

*When ordering this publication please specify Title and Underscored portion of File Number.

Copyright 1966
Honeywell Inc.
Electronic Data Processing Division
Wellesley Hills, Massachusetts 02181

PREFACE

This manual describes how to use the Series 200 Linear Programming System H (Basic), called LP H (Basic). The system consists of those agenda (programs) most frequently used in solving linear programming problems. The system is modular: additional agenda can be added to produce an extended LP H system. Linear programming (LP) is a group of mathematical procedures used to calculate what amounts of resources, such as raw material or manpower, should be allocated either to maximize some objective such as profit or to minimize some objective such as cost.

The value of linear programming is that it can help to make precise decisions about problems which previously were too large and complex to be solved except by trial and error. It helps to solve such problems without incurring the uncertainties and losses which result when less rigorous and precise methods are used.

LP H is a load-and-go system written in the Fortran D language. The minimum equipment required depends on the size of the source program, the size of the object program, and the size of the LP problem to be solved. The source program can be compiled on any Series 200 system that can compile Fortran D, and that has at least 32,768 characters of memory. Revision 3.0 of the Fortran D Compiler System Tape (CST) must be used.

This manual is divided into six sections. Section I describes how to formulate a linear programming problem. Included is a discussion of the mathematical formulation required, definitions of terms used, kinds of solution obtained, and method of solution. Section II describes how to prepare the required input data. Included is a discussion of the kinds of data used and how the data are coded. Section III describes the agenda individually. Included is a discussion of the names and purposes of the agenda and a description of how their call cards are coded. Section IV describes the operating procedures for the system. Included are basic equipment requirements, the general procedure used, the names and explanations printed during the run, the SENSE switch options available, and special source-language considerations for introducing array dimensioning and tolerance adjustments. Section V describes how to interpret the output from an LP H run. Included are a list of printout headings and an explanation of each. Finally, Section VI relates all the previous sections to the actual solution of a sample problem. Included is a statement of the sample problem, including how it is formulated, the agenda required, the input procedure, the output obtained, and changes made in the problem to see how these changes affect the original solution.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont)

LIST OF TABLES

SECTION I

PROBLEM FORMULATION

The general linear programming problem can be formulated as follows. Find nonnegative values of the variables:

$$x_1, x_2, \ldots, x_n$$

which maximize (or minimize) the expression:

(1-1) $$c_1x_1+c_2x_2+\ldots+c_nx_n = Z$$

and which satisfy a given set of simultaneous equations:

(1-2)
$$\begin{array}{l} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n = b_1 \\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n = b_2 \\ \quad\cdot \qquad \cdot \qquad\quad \cdot \qquad \cdot \\ \quad\cdot \qquad \cdot \qquad\quad \cdot \qquad \cdot \\ \quad\cdot \qquad \cdot \qquad\quad \cdot \qquad \cdot \\ a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n = b_m \end{array}$$

Equation (1-1) is called the <u>objective function.</u> Equations (1-2) are called the <u>constraints</u> of the problem. The requirement that the variables be nonnegative in value is called the <u>nonnegativity condition</u> for the problem. The a's, b's and c's are constants; the x's are variables. The a's are called input/output <u>(I/O) coefficients.</u> The b's are called the <u>constraint values.</u> The c's are called either <u>cost coefficients</u> or <u>objective coefficients.</u> In this manual, they are called <u>objective coefficients.</u> The Z in equation (1-1) is called the <u>objective value.</u> The column of b's is called the <u>right-hand side,</u> the <u>constraint vector</u>, the <u>requirements vector</u>, or the <u>stipulations vector</u>. In this manual, it is called the <u>right-hand side.</u>

A numerical example of (1-1) is:

(1-3) $$2x_1+3x_2+4x_3+5x_4 = Z$$

where: $2 = c_1$

$3 = c_2$

$4 = c_3$

$5 = c_4$

and Z is the objective value to be minimized or maximized.

A numerical example of (1-2) is:

(1-4)
$$7x_1+8x_2+9x_3+10x_4 = 110$$
$$4x_1+5x_2+6x_3+7x_4 = 120$$
$$3x_1+6x_2+9x_3+12x_4 = 150$$

where:
$$a_{11}=7, a_{12}=8, a_{13}=9, a_{14}=10, b_1 = 110$$
$$a_{21}=4, a_{22}=5, a_{23}=6, a_{24}=7, b_2 = 120$$
$$a_{31}=3, a_{32}=6, a_{33}=9, a_{34}=12, b_m = 150$$

The formulation shown in (1-1) through (1-4) is the standard form of the LP problem; all the constraints are initially written as equations or have been reduced to equations. However, LP methods are so general that the constraints initially can be written as inequalities. Two kinds of inequalities are acceptable to LP H, viz., the "equal to or greater than" type (symbolized by $\geq$) and the "equal to or less than" type (symbolized by $\leq$). Such constraints cause no difficulty; they always can be made into equalities by adding or subtracting a compensating quantity on the left side of each inequality.

If an inequality constraint is of the "equal to or less than" type:

(1-5)
$$a_{i1}x_1+a_{i2}x_2+ \dots +a_{in}x_n \leq b_i$$

it always can be made into an equation by adding a nonnegative value s_i to the left-hand side. The result is:

(1-6)
$$a_{i1}x_1+a_{i2}x_2+\dots+a_{in}x_n+(1)s_i = b_i$$

Being a variable, s_i is another unknown in the equation and is called either a slack variable or a positive slack. In this manual, it is called a positive slack.

If an inequality constraint is of the "equal to or greater than" type:

(1-7)
$$a_{i1}x_1+a_{i2}x_2+ \dots +a_{in}x_n \geq b_i$$

it can always be made into an equation by subtracting a nonnegative value s_i from the left-hand side. The result is:

(1-8)
$$a_{i1}x_1+a_{i2}x_2+ \dots +a_{in}x_n+(-1)s_i = b_1$$

In this case, s_i is called either a surplus variable or a negative slack. In this manual, it is called a negative slack. Subtraction is effected by assigning the slack variable a negative coefficient, as shown above.

Whenever a problem constraint is initially an equality rather than an inequality, as:

(1-9) $$a_{i1}x_1+a_{i2}x_2+ \ldots +2_{in}x_n = b_i$$

a variable A_i (finally driven to zero) is added to the left-hand side of the equation. The result is:

(1-10) $$a_{i1}x_1+a_{i2}x_2+ \ldots +a_{in}x_n+(1)A_i = b_i$$

Being a variable, A_i is another unknown in the equation and is called an artificial variable.

Both slacks and artificials are adjoined to the problem constraints as a computational convenience to simplify the task of obtaining an initial (basic) solution. Slacks may be carried along as successive solutions are obtained, but artificials must be eliminated or driven to zero before the optimal solution is obtained; otherwise the solution is infeasible (not a valid solution).

LP H introduces slacks and artificials automatically; the user is not required to introduce them into his formulation of the LP problem.

KINDS OF SOLUTION

The number of variables in an LP problem is generally greater than the number of constraints. When this happens, the result is that the system is underdetermined. This means that there can be an infinite number of solutions to the problem. Any solution of the constraints which satisfies the nonnegativity condition is called a feasible solution to the problem.

Because the number of variables is usually greater than the number of constraints, the constraints cannot generally be solved to obtain all the values of the variables. In fact, the constraints can be solved for all values of the variables only if the number of variables equals the number of constraints or if the extra variables are set equal to zero. Since pertinent variables in a problem cannot be thrown out to make the number of variables equal the number of constraints, the excess variables must be set equal to zero. To obtain all possible solutions by this method, every combination of the excess variables is set equal to zero and the problem is solved repeatedly for the remaining variables. Any solution of the constraints which is obtained by setting the excess variables equal to zero is called a basic solution to the problem. The collection of variables in the basic solution is called the basis of the problem. A basic solution that satisfies the nonnegativity condition is called a basic feasible solution to the problem. Any feasible solution that minimizes Z in the objective function (1-1) is called an optimal feasible solution to the problem. Any basic feasible solution that minimizes Z is called an optimal basic feasible solution to the problem. If the constraints do not confine the objective function to finite values, the problem is said to be unbounded. LP H determines only basic solutions.

METHOD OF SOLUTION

After the LP problem has been formulated as shown in (1-1) through (1-4), the programmer puts the resulting objective function and constraints in matrix form. A matrix is a rectangular array of numbers arranged in rows and columns. The numbers are the objective coefficients of the objective function (the c's of (1-1)), and the I/O coefficients of the constraints (the a's of (1-2)). The right-hand side of the constraints (the b's in (1-2)) is placed to the right of the matrix as a separate column.

The resulting matrix and right-hand side are shown below:

(1-11)

$$\begin{array}{cc}
c_1\ c_2 \cdots c_n & \\
a_{11} a_{12} \cdots a_{1n} & b_1 \\
\cdot \quad \cdot \qquad \cdot & b_2 \\
\cdot \quad \cdot \qquad \cdot & \cdot \\
\cdot \quad \cdot \qquad \cdot & \cdot \\
\underbrace{a_{m1} a_{m2} a_{mn}}_{\text{Matrix}} & \underbrace{b_n}_{\text{Right-Hand Side}}
\end{array}$$

According to the example given in (1-3) and (1-4), (1-11) becomes:

(1-12)

$$\begin{array}{cc}
2\ 3\ 4\ 5 & \\
7\ 8\ 9\ 10 & 110 \\
4\ 5\ 6\ 7 & 120 \\
\underbrace{3\ 6\ 9\ 12}_{\text{Matrix}} & \underbrace{150}_{\text{Right-Hand Side}}
\end{array}$$

Each number in the matrix is called a matrix element. The matrix together with the right-hand side is called a tableau.

The data in (1-11), together with arbitrarily chosen names for each row and column of (1-11), are loaded into a Series 200 computer for processing by the LP H System (see Section II). If any of the problem constraints are equalities, the agendum INPUT (see Section III) generates the required artificial variables. The agendum INPUT also generates the initial basic solution to the problem, including any slacks that may be required. If the initial basic solution is feasible, the system immediately goes to phase II of the agendum NORMAL, which is discussed below.

If the initial solution is infeasible, phase I of the agendum NORMAL is initiated. Phase I attempts to derive a basic feasible solution from the solution created by INPUT. If the problem is inherently infeasible, phase I recognizes this and gives an appropriate message; the user may then introduce a new set of data and begin the process again.

If phase I determines a basic feasible solution, phase II of the agendum NORMAL is initiated. Phase II takes the initial feasible solution produced by phase I or by INPUT and attempts to optimize it. An optimal solution may be determined, in which case Z in (1-1) has a maximum or minimum value.

If an optimal solution cannot be obtained, the problem may be unbounded. The problem is unbounded if Z in (1-1) has no maximum or minimum.

The algorithm used by the agendum NORMAL is a minimizing algorithm. It uses a version of the revised simplex method called the product form of the inverse.

SECTION II

INPUT DATA

The two main classes of cards submitted to LP H are agendum call cards and matrix cards.

The agendum call cards and the parameters punched on these cards constitute the instruction set for LP H and are specified in Section III. The agendum call cards prepared for use on a specific problem constitute the agenda deck for the problem. One or more agenda decks can be submitted via the card reader as part of a run deck, as described in Section IV. For example, Figure 2-1 illustrates a one-problem agenda deck containing the minimum number of agendum call cards required to solve a problem.

This section describes the makeup of the matrix file (see Figure 2-2) and contains specifications for the cards in that file. The matrix file contains two kinds of cards: announce cards and problem data cards. Problem data cards are organized under announce cards as described below. They contain row names, column names, and the numeric constants c_j, a_{ij}, and b_i that are elements of the problem and on which the agenda operate.

Each INPUT card in the card reader calls the agendum INPUT, which processes the corresponding matrix file to produce an A tape containing the initial basic solution. The A tape, in turn, is input to NORMAL for optimization. Generally, for any agendum deck in the card reader, the corresponding matrix file is submitted as card images on the I tape. The I tape, illustrated in Figure 2-2, is prepared before the production run.[1] Matrix files for many problems can be stacked in any sequence on the I tape; any INPUT card in the card reader can specify processing of any matrix file on the I tape.

Alternatively, the matrix file can be submitted as a card deck embedded in the agendum deck as illustrated in Figure 2-1. In this case, a PARAMS agendum call card must be used to assign the card reader as logical device I; the PARAMS card is placed before the INPUT card.

ANNOUNCE CARDS

Any card in the matrix file is an announce card (also known as a control card or indicator card) if it contains one of the following words punched beginning in column one:

---

[1] In an ongoing linear programming application, the I tape could be produced as a byproduct of other data processing or process control operations.

1. ROW ID — This card announces to the LP H monitor that (1) all the problem data cards for one problem lie after this card and before the next ENDATA card, assuming a forward search of the matrix file, and (2) the cards that both follow the ROW ID card and precede the next announce card are problem data cards containing row names for the problem matrix.

2. MATRIX — This card announces that the cards that both follow the MATRIX card and precede the next announce card are problem data cards containing matrix elements. The matrix elements consist of cost coefficients, c_j, and input/output coefficients, a_{ij}. During INPUT, a matrix element is excluded from the A matrix if the element's row name does not match one of the row names in the row-name list created from the ROW ID card set above. Thus, a given constraint in the MATRIX card set can be suppressed by removing the appropriate row-name card from the ROW ID card set. To announce the beginning of the constraints, a parameter on the INPUT agendum card specifies how many of the first rows of the matrix are cost rows; the remaining matrix rows are treated as constraint rows, containing a_{ij}, and are indexed 1, 2, ..., m. To announce the objective function, a parameter on the NORMAL agendum card specifies which of the cost rows is to be used as the objective function. The other cost rows are ignored during NORMAL; they are transformed during successive iterations but are not used in determining the optimal solution. NORMAL may be used again to optimize a cost row previously ignored, or these rows may be used by another agendum such as DO. PCR.

3. RHSIDE — This card announces that the cards that both follow the RHSIDE card and precede the next announce card are problem data cards containing right-hand-side elements, b_i.

4. ENDATA — This card announces that the matrix file contains no more cards for the current problem.

5. EOF — If the matrix file is on the I tape, the EOF card image is needed to announce the logical end of the file. The EOF card is followed by a blank card to terminate double buffering.

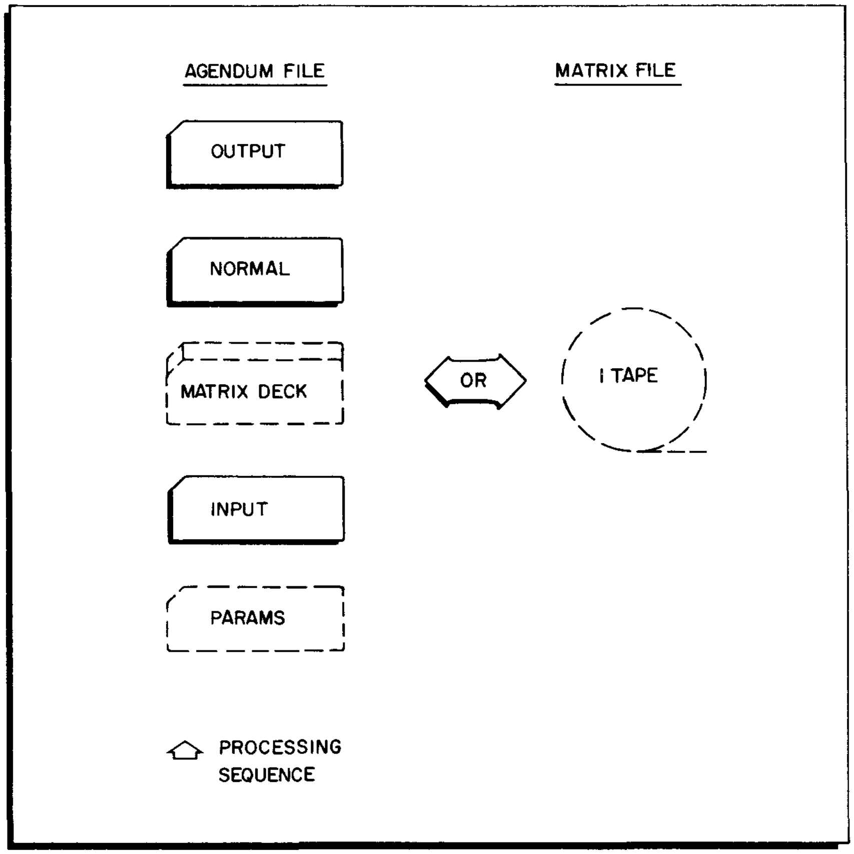


Figure 2-1. Minimal Input for a Problem

PROBLEM DATA CARDS

The manner in which problem data cards are coded is specified below. A linear programming coding form, compatible with all Honeywell LP software, is available for use in preparing the problem data. In the following specifications, the filled-in coding form illustrates the layout of the card. Below the form are specifications for the parameters shown on the form. In general, upper-case letters shown in the form are coded literally in the positions indicated, whereas lower-case letters indicate the position of the parameter but take values defined by the corresponding paragraph of the specification below the form. For each parameter, the specification indicates the columns occupied by the parameter, the parameter under consideration, and the description or meaning of the parameter together with format and value limitations, etc.

Fortran Format

Problem data cards punched in MATRIX format or RHSIDE format accept a numeric constant in the "VALUE OF ELEMENT" field, i.e., in columns 19-30. The specifications for MATRIX and RHSIDE format given below restrict the decimal point to column 24, effectively limiting the value of the element to the range (in absolute value) $0000.000000 \leq n \leq 9999.999999$. This format (viz., decimal point in column 24) should be followed if the programmer wants to achieve compatibility with all Honeywell LP software.

However, these numbers are read using an F12.6 format, so any number configuration compatible with this format is acceptable. If the programmer wants a larger range of numbers and is willing to sacrifice format compatibility with some other LP software, he can punch a decimal point anywhere in columns 20-30. Then the absolute value of the element lies in the range $.0000000000 \leq n \leq 9999999999$.

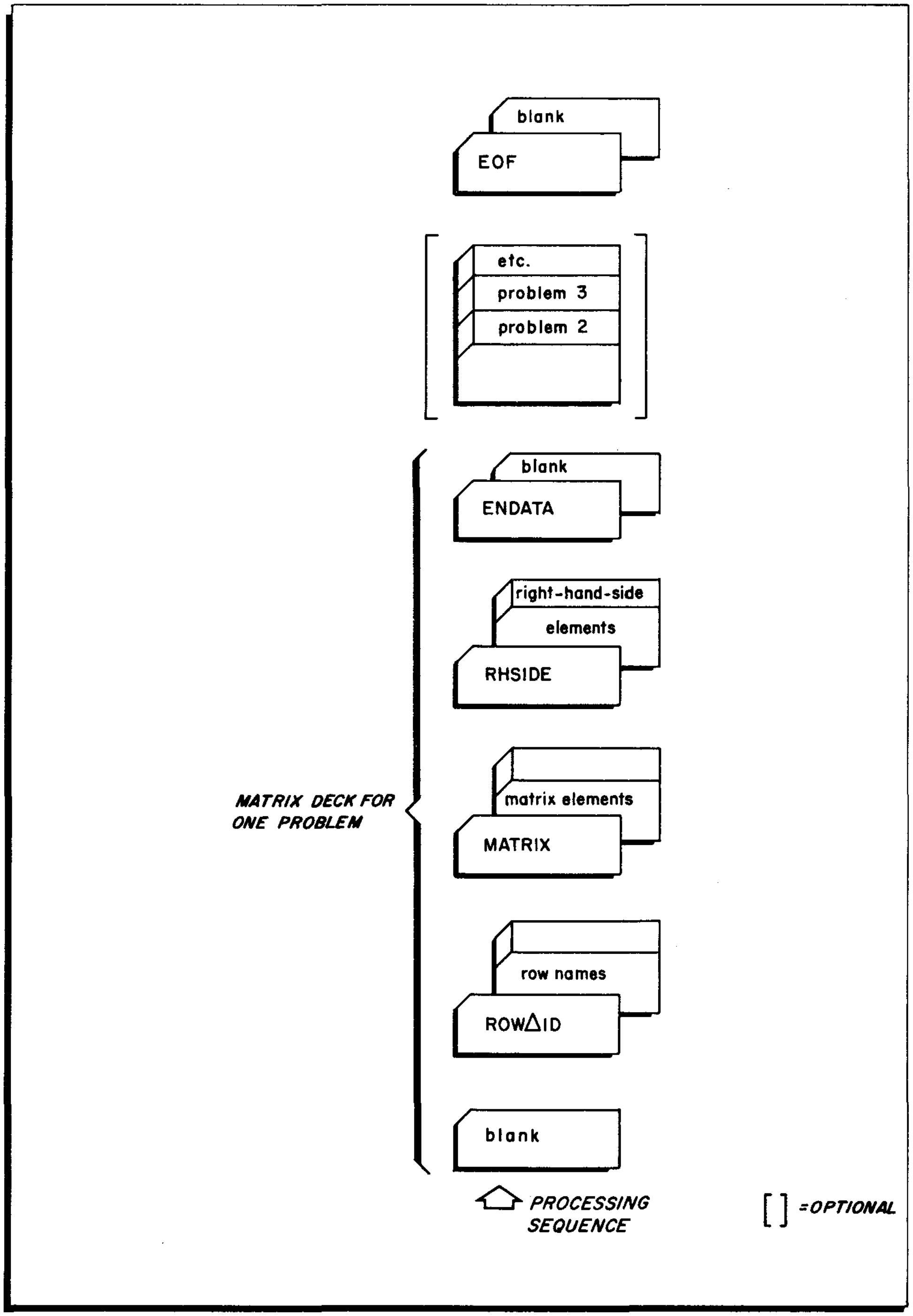
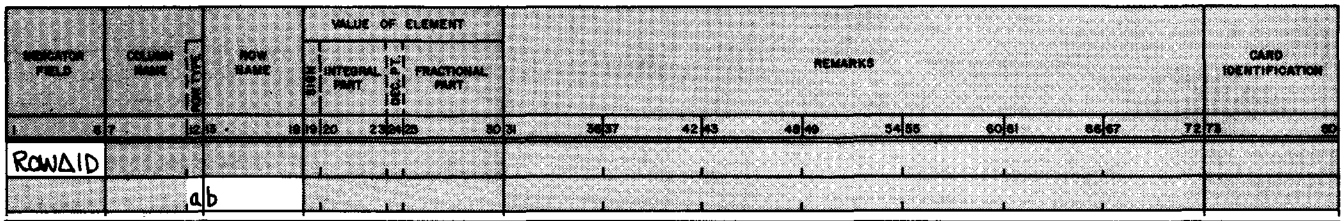


Figure 2-2. Makeup of the Matrix File for INPUT

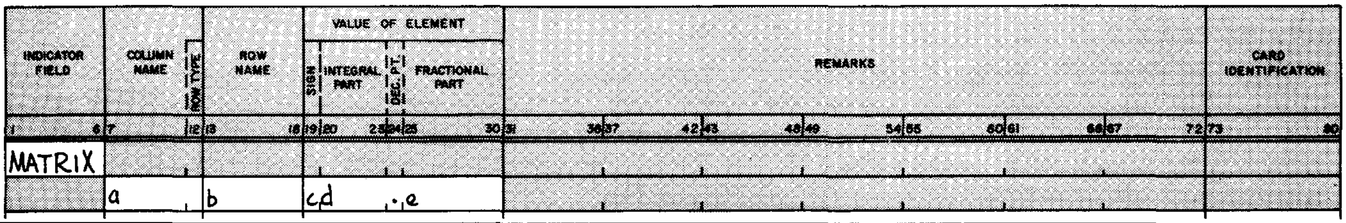
Problem Data Specifications

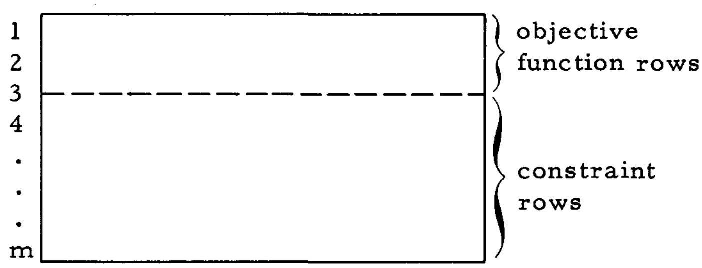
ROW ID FORMAT



Columns	Parameter	Description
1-6	ROWΔID	ROW ANNOUNCE CARD. This card specifies that the cards that follow consecutively contain row-names. One row name is punched per card as follows:
12	a	ROW TYPE. If the row is an inequality row of the sense (≤), parameter "a" must be a "+". If the row is an equality row of the sense (≥) parameter "a" must be "-". If the row is a strict equality row (=), as is an objective function row, parameter "a" must be blank or zero.
13-18	b	ROW NAME. Alphanumeric. This name cannot be blank or duplicate a column name; it should be unique within the problem. The names of the objective function rows (cost rows) must precede the names of the constraint rows, and all row names must precede the MATRIX card described below. There is one row name per card, and each name must be six characters in length or less.

MATRIX FORMAT



Columns	Parameter	Description
1-6	MATRIX	MATRIX ANNOUNCE CARD. This card specifies that the cards that follow consecutively are matrix-element cards. Vectors cannot be split (i.e., all elements for a given matrix column must appear together), vectors can appear in random order, and elements can appear in random order within vectors. Only non-zero entries need be specified. Matrix elements are punched one per card as follows:
7-12	a	COLUMN NAME. This alphanumeric name cannot be blank and cannot duplicate a column or row name.
13-18	b	ROW NAME. This alphanumeric name should be the same row name as listed in parameter "b" under ROW ID (see preceding page). The matrix rows are indexed by the LP system beginning with one, as follows: 
19	c	SIGN OF ELEMENT. If the element is zero or positive, leave this column blank or enter "+"; if the element is negative enter "-".
20-23	d	INTEGRAL PART OF ELEMENT. These decimal digits must be right justified against the decimal point. Leading zeros can be omitted.
24	.	DECIMAL POINT OF ELEMENT.
25-30	e	FRACTIONAL PART OF ELEMENT. These decimal digits must be left justified against the decimal point. Trailing zeros can be omitted. The numbers are read using Fortran F12.6 format (see page 2-3), so any number format compatible with F12.6 is acceptable in columns 19-30.

RHSIDE

	INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
	1 6	7 12	12	13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
1	RHSIDE									
2		a		b	c	d	.	e		

Columns	Parameter	Description
1-6	RHSIDE	RIGHT-HAND-SIDE ANNOUNCE CARD. This card specifies that the cards which follow consecutively are right-hand-side element cards. More than one right-hand side can be introduced. Each right-hand side requires its own card set. No duplicate right-hand-side element names can be used. Elements can appear in random order within the vector. Only non-zero elements need be punched. Right-hand-side elements are punched one per card as follows:
7-12	a	NAME OF RIGHT-HAND-SIDE VECTOR. This alphanumeric name cannot be blank and cannot duplicate a column or row name.
13-18	b	ROW NAME. This alphanumeric name should be the same row name as listed in parameter "b" under ROW ID (see page 2-5).
19	c	SIGN OF ELEMENT. If the element is zero or positive, leave this column blank or enter "+"; if it is negative, enter "-".
20-23	d	INTEGRAL PART OF ELEMENT. These decimal digits must be right justified against the decimal.
24	.	DECIMAL POINT OF ELEMENT.
25-30	e	FRACTIONAL PART OF ELEMENT. These decimal digits must be left justified against the decimal point. Trailing zeros can be omitted. The numbers are read using Fortran F12.6 format (see page 2-3); any number format compatible with F12.6 is acceptable in columns 19-30.

ENDATA FORMAT

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7 12		13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
ENDATA									

Columns	Parameter	Description
1-6	ENDATA	END OF DATA. This card must appear in the input deck in the position indicated in Figure 2-1.

EOF FORMAT

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7 12		13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
EOF									

Columns	Parameter	Description
1-3	EOF	END OF FILE. This card must appear in the input deck following the last problem when more than one problem is stacked.

LOAD Format

Cards punched in LOAD format are used by the agendum LOAD to rewrite the H region, establishing a new basis in memory. Cards punched by the agendum PUNCH are in LOAD format; they contain the matrix row indices and the corresponding vector names which are to overwrite the basis list in memory.

Each card must be punched beginning in column seven and can contain a maximum of five contiguous fields which have 12 columns each. In each field, the first six columns contain a vector name; the last six columns contain a row name or all blanks. If a slack vector is being inserted in the basis, the first six columns of the field contain the name of the row to which the slack corresponds.

The set of LOAD cards is terminated by an ENDATA card followed by a blank card.

To know how to code these revision cards, the reader needs to know how the agendum LOAD interprets them. For each 12-column field of each revision card, the agendum loads the column name into the position of the H region specified by the row name, as follows:

1. If the row name is blank, the H region is searched for the name of an artificial vector. If such a name is found, it is replaced with the vector name of the card field; else the vector name of the card field is ignored and a message is put on the M tape.
2. If the row name is not blank and is identical to a row name in an (n^{th}) position of the row-name list, the vector name of the card field is stored in the n^{th} position of the H region; else the vector name of the card field is ignored and a message is put on the M tape.
3. When the agendum LOAD is finished, INVERT should be called. When INVERT is finished, the next card is read from the card reader.

LOAD FORMAT

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS							CARD IDENTIFICATION
1–6	7–12	12	13–18	19	20–23	24	25–30	31–36	37–42	43–48	49–54	55–60	61–66	67–72	73–80
	a		b	a			b	a	b	a	b	a	b		

Columns	Parameter	Description
1-6		Load cards, when used, follow the LOAD agendum call card (see the LOAD and PUNCH agenda.) These first six columns are always left blank.
7-12, 9-24, 33-36, 43-48, 55-60	a	COLUMN NAME. These card columns contain matrix column names. Any name can be six characters or less in length. The unused card columns are left blank. A maximum of five column names can be punched per card. If the column names are fewer than the number of column-name card fields available, the unused card columns are left blank.
13-18, 25-30, 37-42, 49-54, 61-66,	b	ROW NAME. These card columns contain matrix row names or are blank. Any name can be six characters or less in length. The unused card columns are left blank. A maximum of five row names can be punched per card. If the rownames are fewer than the number of row-name card fields available, the unused card columns are left blank.
67-80		The last 14 card columns are always left blank.

Introduce Format

An introduce card contains row and/or column names of the problem matrix. The names can be all row names, all column names, or a combination of row and column names. Introduce cards, when used, follow the BRANGE or CRANGE agendum. The card set is terminated by an ENDATA card followed by a blank card.

INTRODUCE FORMAT

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS							CARD IDENTIFICATION
1 6	7	12	13 18	19	20 23	24	25 30	31 36	37 42	43 48	49 54	55 60	61 66	67 72	73 80
	a		a	a			a	a	a	a	a	a	a	a	

Columns	Parameter	Description
1-6		These first six columns are always left blank.
7-12 13-18 19-24 25-30 31-36 37-42 43-48 49-54 55-60 61-66 67-72	a	ROW OR COLUMN NAME. These card columns contain row names if the agendum using these cards is BRANGE. They contain matrix-column names if the agendum using these cards is CRANGE; a slack vector can be specified by coding the corresponding row name. Any name can be six characters or less in length. The unused card columns are left blank. A maximum of 11 names can be punched per card. If the names are fewer than the number of card fields available, the unused card columns are left blank.
73-80		The last eight card columns are always left blank.

SECTION III
AGENDA

This section describes the agenda provided by LP H. A description of the nature and purpose of each agendum is included, together with specifications for the parameters. Table 3-1 lists the agenda required to perform the functions provided in LP H (Basic).

Agenda are programs within the LP H system. Each agendum performs one or more LP operations upon the data introduced into the system. Each is called by its own agendum call card. When introduced into the system, call cards are interpreted by a resident control program which loads the agenda one at a time into memory and initiates their execution. Agendum call cards, then, provide complete control over all the programs of the system, the sequence in which the programs operate, and the final output produced. As a result, complete system flexibility is always available.

The coding form described in Section II is used to code agendum call cards. Shown first is a filled-in coding form illustrating the layout for each card. Below this form are specifications for the parameters shown on the form. The specifications indicate the columns to be punched, the parameter under consideration, and the description or meaning of the parameter involved. Unless otherwise stated, all parameters are left justified.

Agenda names are punched one to a card beginning in column one. The agenda cards are fed into the card reader in the order in which the agenda are to be called. Only those agenda which are actually used in the run need be called.

Table 3-1. LP Functions and Required Agenda

Function	Agenda
Input	INPUT
Optimization of Initial Solution	NORMAL
Output of Matrix, Solutions	OUTPUT
Restarting the Problem	GETOFF, RESTART; PUNCH, LOAD LDHREG
Sensitivity (Range) Analyses	BRANGE; CRANGE
Parametric Programming	DO.PLP; DO.PCR
Processing Control	PARAMS; STOP; HALT; REWIND; TITLE; INVERT

The agenda INPUT, NORMAL, and OUTPUT must be executed for each problem. This sequence must be preserved even when other agenda are interspersed. Other agenda can be used in a given run, depending on the functions desired.

REQUIRED AGENDA

INDICATOR FIELD (1-6)	COLUMN NAME (7-12)	ROW NAME (13-18)	VALUE OF ELEMENT (19-30)	REMARKS (31-72)	CARD IDENTIFICATION (73-80)
INPUTΔ	a b	LISTΔΔ	REWINDCARDSΔ	CHECKΔ	

Columns	Parameter	Description
1-6	INPUT	This agendum loads problem data into memory. If the INPUT card is not followed by the data, INPUT reads the problem data from the I tape. Output from this agendum is a matrix tape called the A tape.
7-9	a	NUMBER OF OBJECTIVE FUNCTIONS. This optional, right-justified parameter is an integer having a maximum of three digits. It is specified only if more than one objective function is included in the matrix. Parameter a on the NORMAL agendum card specifies which of these functions is to be minimized; the other cost rows are ignored by NORMAL.
10-12	b	NUMBER OF PROBLEMS TO BE SKIPPED. This optional, right-justified parameter is a fixed point number indicating which problem on the I tape is to be read into memory. If this parameter is blank, the first problem encountered is read into memory. If any number, n, is punched, the first n problems are skipped and the following problem is read into memory.
13-18	LIST	MATRIX LISTING. This optional parameter is used to request an analysis of the matrix data, in which case the following matrix information is written on the output (M) tape for later printing: (1) The name of each matrix column (structural vector) together with the number of nonzero elements in the column. (2) The name of each matrix row together with the number of nonzero elements in the row.
19-24	REWIND	REWIND THE I TAPE. This optional parameter causes the I tape to be rewound after INPUT has finished. It is used only when the data cards are on tape. To rewind the I tape before INPUT, see the REWIND agendum.
25-30	CARDS	IGNORE CARDS. This optional parameter causes INPUT to ignore all row names after the ROW Δ ID card image on the I tape. It is used when a selected subset of row names is to be read from the card reader; the matrix and right-hand side are edited accordingly. The row name cards are punched in standard format as specified in Section II. They must be preceded by a ROWΔID card and followed by an ENDATA card. This deck of cards must immediately follow the INPUT agendum card.

Columns	Parameter	Description
31-36	CHECK	MATRIX NAME AND ELEMENT CHECKING. This optional parameter requests that a check be made for duplicate column names and split matrix columns.

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7	12	13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
NORMAL	a b								

Columns	Parameter	Description
1-6	NORMAL	This agendum solves the specified problem on the A tape using the revised simplex algorithm. First, it attempts to find a feasible solution. If one is found, it attempts to find an optimal feasible solution. If there is no feasible solution, or the solution is feasible but unbounded, a message is given and control exits from NORMAL.
7-9	a	NUMERICAL INDEX OF OBJECTIVE FUNCTION. This optional parameter is an unsigned, right-justified, fixed-point number. It is used only if more than one objective function is contained in the LP problem. The rows of the matrix are indexed starting with 1. For example, if there are two objective functions and the second is to be optimized, parameter a should be 2.
10-15	b	NAME OF RIGHT-HAND SIDE TO BE USED. This optional parameter is used only if the right-hand side currently in memory is not to be used. Parameter b is the name of the new right-hand side to be read into memory from the A tape.

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7	12	13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
OUTPUT	OLD								

Columns	Parameter	Description
1-6	OUTPUT	This agendum must be used during each run to list the output generated during the run. OUTPUT writes two end-of-information records at the current position of the M tape, rewinds the tape, and edits it onto the printer, giving edited printout of the M tape.

Columns	Parameter	Description
7-9	OLD	EDITED PRINTOUT. This optional parameter is used only to edit the M tape at some time other than at the normal termination of the current run.

INTERRUPT AND RESTART AGENDA

Tape-Oriented

INDICATOR FIELD	COLUMN NAME	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 – 6	7 – 12	13 – 18	19	20 – 23	24	25 – 30	31 – 72	73 – 80
GETOFF	a							

Columns	Parameter	Description
1-6	GETOFF	This agendum writes the pertinent (current) restart information on a tape for later use by the agendum RESTRT. The restart information is merely copied from memory. GETOFF can be followed consecutively by any agendum. It is used for any of these reasons: (1) As an emergency measure to free the machine. (2) To save problem status at a desired point for later use in a complicated run. (3) To provide an extra copy of the A tape (matrix tape) for long runs with extensive data by using GETOFF immediately after INPUT.
7	a	U TAPE. This optional parameter is used when restart information is to be written on a work tape called the U tape. If this parameter is omitted, the restart information is written on the logical A tape.

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7	12	13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
LDHREG	a								

Columns	Parameter	Description
1-6	LDHREG	This agendum overwrites the current H region with a copy (specified by parameter a) of the H region previously saved and stored on the A tape. Copies of the H region are stored periodically on the A tape during NORMAL, DO.PCR, and DO.PLP before these agenda call INVERT. A problem can be restarted by loading one of the saved H regions (using LDHREG) and calling INVERT. Then NORMAL, DO.PCR, or DO.PLP can be called, as desired.
7-10	a	BASIS LIST IDENTIFICATION. If parameter a is blank or 1, it specifies the last H region saved. If parameter a is 2, it specifies the next-to-last H region saved, etc. Parameter a is a right-justified, unsigned, fixed-point number.

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7	12	13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
RESTRT	a b								

Columns	Parameter	Description
1-6	RESTRT	This agendum retrieves information from the GETOFF tape (i.e., from the A tape unless parameter a is used). Memory and tapes are set up exactly as they were found by the previous execution of GETOFF. Any agendum can follow RESTRT.
7-9	a	RETRIEVE INFORMATION FROM U TAPE. This optional parameter specifies that the restart information is to be taken from the U tape.
10-12	b	GETOFF PROBLEM TO BE RETRIEVED. This optional parameter is stored as a right-justified, fixed-point number. If this parameter is 1, the first GETOFF problem is retrieved from the tape; if it is 2, the second problem is retrieved, etc. If this parameter is blank, the last GETOFF problem is retrieved.

Card-Oriented

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7	12	13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
LOAD	MOD								

Columns	Parameter	Description
1-6	LOAD	This agendum modifies the current basis or the original basis created by INPUT. The modification information comes from hand-punched cards or from the cards punched by the PUNCH agendum. When LOAD ΔΔ is punched without parameters, the original basis is retrieved and modified according to LOAD cards which follow the agendum card consecutively. See "LOAD Format" in Section II. The next agendum after LOAD must be INVERT.
7-9	MOD	CURRENT-BASIS MODIFICATION. If this parameter is used, the agendum modifies whatever H region is currently in memory according to the LOAD revision cards; else the agendum reads into memory the basis created by INPUT and modifies that H region according to the revision cards.

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7	12	13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
PUNCH									

Columns	Parameter	Description
1-5	PUNCH	This agendum punches into cards the H region in memory, i.e., the row indices and the corresponding elements of the basis list. The cards contain the minimum information needed to restart processing of the NORMAL agendum. PUNCH can be used to save an optimal solution for later use. The cards punched are called load cards (see "LOAD Format" in Section II). The load cards can be used by the LOAD agendum to restart the run.

POST-OPTIMAL AGENDA

Sensitivity Analyses

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	VALUE OF ELEMENT: INTEGRAL PART	VALUE OF ELEMENT: DEC. PT.	VALUE OF ELEMENT: FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7	12	13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
BRANGE	NAMES								

Columns	Parameter	Description
1-6	BRANGE	This agendum computes the ranges over which the values of right-hand-side elements, b_i, can be changed one at a time without producing infeasibility. Generally, this agendum follows NORMAL and precedes DO. PLP.
7-11	NAMES	NAMES OF ROWS TO BE RANGED. This parameter is optional. If it is not used, the entire right-hand side is ranged. If it is used, it is followed by cards which name the specific right-hand-side elements which are to be ranged. These cards are punched as described under "Introduce Format" in Section II.

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	VALUE OF ELEMENT: INTEGRAL PART	VALUE OF ELEMENT: DEC. PT.	VALUE OF ELEMENT: FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7	12	13 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
CRANGE	NAMES								

Columns	Parameter	Description
1-6	CRANGE	This agendum computes the ranges over which the objective coefficients, c_j, can be changed one at a time without requiring a change in the basis to maintain optimality. Generally, this agendum follows NORMAL and precedes DO. PCR.
7-11	NAMES	NAMES OF ROWS TO BE RANGED. This parameter is optional. If it is not used, all the basic objective coefficients are ranged; else it is follwed by cards which name the specific basic objective coefficients which are to be ranged. These cards are punched as described under "Introduce Format" in Section II.

Parametric Programming

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	VALUE OF ELEMENT: SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1 6	7	12	15 18	19	20 23	24	25 30	31 36 37 42 43 48 49 54 55 60 61 66 67 72	73 80
DO.PCR	a b		c						

Columns	Parameter	Description
1-6	DO. PCR	This agendum is used to derive one optimal solution from another, when the cost coefficients have changed, without re-executing the NORMAL agendum. DO. PCR produces a composite objective-function row which is a linear combination of the objective-function row last used and a change objective-function row times a scale factor, phi, generated by DO. PCR.
7-9	a	CHANGE-ROW INDEX. This optional parameter specifies the numerical index of the objective-function row to be used as the change row. Parameter a is a right-justified, unsigned integer. If it is omitted, the second objective-function row in the matrix is used.
10-15	b	MAXIMUM VALUE OF PHI. This optional parameter, called a specified maximum, is the limit of phi, set by the user. If this parameter is omitted, DO.PCR iterates until phi becomes unbounded or until it reaches an absolute maximum value determined by the characteristics of the problem. If parameter b is used, it must be an unsigned number having six or fewer digits: it is read using the Fortran F6.3 format.
16-21	c	DELTA PHI. If this optional parameter is specified, it controls the number of current-solution printouts produced during DO.PCR. If phi increases to a value greater than the assigned delta phi, a current-solution printout is produced. The greater the number of current solution printouts desired, the smaller should be the assigned value of delta phi. A current-solution printout consists of the current basis list and current solution vector. If this parameter is omitted, no solution printout occurs until DO.PCR terminates. A current-solution printout for each iteration can be obtained by setting parameter c equal to .00001. The parameter is an unsigned integer consisting of 6 or fewer digits; it is read using the Fortran F6.3 format. In addition to obtaining a new optimal solution when cost coefficients change, DO.PCR can be used to determine what change of basis is required at some limit determined by CRANGE.

INDICATOR FIELD	COLUMN NAME	ROW NAME	VALUE OF ELEMENT: INTEGRAL PART	VALUE OF ELEMENT: FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
DO.PLP	a	b	c	d		

Columns	Parameter	Description
1-6	DO.PLP	This agendum is used to derive one optimal solution from another, when the right-hand-side has changed, without re-executing the NORMAL agendum. DO.PLP produces a new right-hand side which is a linear combination of the right-hand side last used and a change right-hand-side vector times a scale factor, theta, generated by DO.PLP.
7-12	a	CHANGE-VECTOR NAME. The name of the change right-hand-side vector to be multiplied by the scale factor. This name should be punched as in the input data deck (e.g., with or without leading blanks).
13-18	b	MAXIMUM VALUE OF THETA. This optional parameter, called a <u>specified maximum</u>, is the limit of theta set by the user. If this parameter is omitted, DO.PLP iterates until theta becomes unbounded or until it reaches an <u>absolute maximum</u> value determined by the characteristics of the problem. If parameter b is used, it must be an unsigned number having six or fewer digits; it is read using the Fortran F6.3 format.
19-24	c	DELTA THETA. If this optional parameter is specified, it controls the number of current-solution printouts produced during DO.PLP. When theta increases by a value greater than the assigned delta theta, a current-solution printout is produced. The greater the number of current-solution printouts desired, the smaller should be the assigned value of delta theta. A current solution printout consists of the current basis list and current solution vector. If this parameter is omitted, no solution printout occurs until DO.PLP terminates. A current-solution printout for each iteration can be obtained by setting parameter c equal to .00001. The parameter is an unsigned number consisting of 6 or fewer digits; it is read using the Fortran F6.3 format.
25-30	d	NEW RIGHT-HAND SIDE ON THE A TAPE. If this optional parameter is omitted, nothing is written on the A tape when DO.PLP terminates. If it is used, a composite right-hand side named as parameter d is written on the A tape. The composite right-hand side equals: (Original right-hand-side) + Theta_L (Change right-hand-side), where Theta_L is (1) the specified or absolute maximum value of theta or (2) the last finite value of theta computed in the case theta has become unbounded.

Columns	Parameter	Description
25-30 (cont)	d	If this composite right-hand side is submitted to NORMAL (as parameter b), the optimal solution determined by NORMAL should equal the last solution printed by DO. PLP. Thus, NORMAL can be used to verify that DO. PLP has operated correctly.

PROCESSING CONTROL AGENDA

INDICATOR FIELD (1-6)	COLUMN NAME (7-12)	ROW TYPE	ROW NAME (13-18)	VALUE OF ELEMENT: SIGN (19)	INTEGRAL PART (20-23)	DEC. PT. (24)	FRACTIONAL PART (25-30)	REMARKS (31-72)	CARD IDENTIFICATION (73-80)
HALT									

Columns	Parameter	Description
1-4	HALT	This agendum causes a halt in processing. To continue, press the RUN button, permitting the next agendum card to be read.

INDICATOR FIELD (1-6)	COLUMN NAME (7-12)	ROW TYPE	ROW NAME (13-18)	VALUE OF ELEMENT: SIGN (19)	INTEGRAL PART (20-23)	DEC. PT. (24)	FRACTIONAL PART (25-30)	REMARKS (31-72)	CARD IDENTIFICATION (73-80)
INVERT	SAVE								

Columns	Parameter	Description
1-6	INVERT	This agendum inverts the current basis, to reduce digital error and iterating time. Uses of INVERT: (1) If CRANGE, BRANGE, DO. PLP, or DO. PCR is to be used, some time can be saved by calling INVERT immediately after NORMAL finds an optimal solution. (2) When NORMAL finds an optimal solution, the digital error in the solution can be reduced by calling INVERT and then recalling NORMAL. NORMAL should exit immediately, verifying that the optimal solution is correct.
7-11	SAVE	SAVE MEMORY. This option saves the current H-region (basis list) by writing it onto the A tape before INVERT is started. Processing of the problem can be restarted using a saved H region; see the agendum LDHREG. If SAVE is omitted, the current H region is not saved. To restart NORMAL using a saved H region, see the agendum LDHREG.

INDICATOR FIELD (1-6)	COLUMN NAME (7-12)	ROW TYPE	ROW NAME (13-18)	VALUE OF ELEMENT: SIGN (19)	INTEGRAL PART (20-23)	DEC. PT. (24)	FRACTIONAL PART (25-30)	REMARKS (31-72)	CARD IDENTIFICATION (73-80)
PARAMS	MAX		a b						

Columns	Parameter	Description
1-6	PARAMS	This agendum enables the programmer to change certain cells in that portion of memory interrogated by more than one chain or subroutine. This communication region is called common storage. Changing certain cells in common storage facilitates running a problem. When used, the PARAMS card must follow INPUT and must precede NORMAL, except as otherwise noted.
10-12	MAX	This parameter determines whether the objective function of the problem is to be maximized or minimized. If MAX is used, the problem following is maximized by NORMAL; else the objective function is minimized.
13-15	a	ITERATION MAXIMUM. This optional right-justified parameter specifies the maximum number of iterations permitted to solve the problem; if the maximum is reached a message is given and control exits to the next agendum. If parameter a is blank, a standard value of 2500 is used. Parameter a is a positive fixed-point number.
16-18	b	DEVICE I ADDRESS. This right-justified parameter specifies the peripheral address of input device I. Usually, device I is a tape having the same address as the U tape. To assign the card reader as device I, use the agendum PARAMS (before INPUT) with parameter b the card reader address. This assignment remains in effect until changed again by another PARAMS card or until the system is reloaded.

INDICATOR FIELD (1-6)	COLUMN NAME (7-12)	ROW TYPE	ROW NAME (13-18)	VALUE OF ELEMENT: SIGN (19)	INTEGRAL PART (20-23)	DEC. PT. (24)	FRACTIONAL PART (25-30)	REMARKS (31-72)	CARD IDENTIFICATION (73-80)
REWIND	a a		a						

Columns	Parameter	Description
1-6	REWIND	This agendum is used to rewind a maximum of three magnetic tapes.
7-9 10-12 13-15	a	TAPE DESIGNATION. Right-justified parameters a specify the physical tape addresses of the tapes to be rewound.

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1–6	7–12	12	13–18	19	20–23	24	25–30	31–72	73–80
STOP									

Columns	Parameter	Description
1-4	STOP	This agendum signals the end of the run. A stop card must appear as the last card in every LP run deck. When the LP monitor reads this card, it exits to the Fortran monitor.

INDICATOR FIELD	COLUMN NAME	ROW TYPE	ROW NAME	SIGN	INTEGRAL PART	DEC. PT.	FRACTIONAL PART	REMARKS	CARD IDENTIFICATION
1–6	7–12	12	13–18	19	20–23	24	25–30	31–72	73–80
TITLE	b								

Columns	Parameter	Description
1-5	TITLE	This agendum prints a title (parameter b) after each agendum name in the printed output.
6-8		These columns are always left blank.
9-80	b	These columns contain the title to be used. A maximum of 72 characters can be printed.

SECTION IV

OPERATING PROCEDURES

EQUIPMENT REQUIREMENTS

The minimum equipment configuration required for LP H depends on the size of the source Fortran program, the size of the object program, and the size of the linear programming problem to be processed, as follows.

Source Program

Until Fortran H is available, the LP H (Basic) System can be compiled by Fortran D; 32,768 characters of memory are required.

Object Program

The object program requires the following minimum equipment configuration:

1. A Series 200 processor having 32,768 characters of memory.
2. Five magnetic tape units.
3. One card reader.
4. One printer.
5. One card punch.

Optional equipment includes:

1. Additional memory to a usable maximum of 131,072 characters.
2. Two additional magnetic tape units.
3. Multiply/Divide instructions.

PROBLEM SIZE

Whether a given problem will fit in memory depends upon the number of constraints (matrix rows) and the size of the memory. Table 4-1 indicates the relation of memory size to problem size.

Table 4-1. Memory Size vs. Problem Size

Core Size (Memory)	Maximum Problem Size (Number of Constraints)
32K	70
40K	125
48K	250
65K	500

TOLERANCES

The standard tolerances used by LP H are shown in Table 4-2.

Table 4-2. Standard LP H Tolerances

Tolerance	Value	Location Containing Value
Maximum Error Tolerance	.0001	QMAXER
Solution Value Zero Tolerance	.00009	QTOLZE
D/J Zero Tolerance	.0005	QTOLPR
Pivot Reject Tolerance	.0005	QTOLRJ
Pivot Zero Tolerance	.00005	QTOLPV

Changes

When changes are made to the standard tolerances, a new binary run tape (BRT) must be produced. To change the tolerances involves changing the first five instructions in the subroutine INIT, as desired. INIT is the first subroutine of the EXEC routine. The EXEC routine, in turn, is the first routine in the source-program deck.

REGION AND BUFFER SIZES

A standard region is defined as a memory area of 70 locations. A standard buffer size is 300 locations. If the size of the region or buffer is changed, note that the buffer size must be at least as great as the region size.

Changes

When changes are made to the standard buffer and region sizes, a new BRT must be produced; every pertinent Fortran dimension card must be changed to reflect the new sizes. A dimension card is used to specify to the compiler how much memory is required for the arrays used in the program. The cards specify the names of the arrays, the number of dimensions of each array named, and the maximum size of each dimension. For example, if the size of the regions is to be increased to 75 locations, each occurrence of a 70 on every dimension card must be changed to 75. The 17th and 18th instructions in the EXEC routine, referenced in "Standard Region and Buffer Sizes," should also be changed to reflect the new region and buffer sizes.

PERIPHERAL DEVICES

The standard peripheral device assignments are listed in Table 4-3.

Table 4-3. Standard Peripheral Device Assignments

Logical Device	Logical Tape Address	Program Cell Containing Address
Card Reader	2	NCRDRD
Printer	3	NPRINT
Punch	5	NPUNCH
M Tape	1	NMTAPE
A Tape	4	NATAPE
F Tape	7	NFTAPE
U Tape	6	NUTAPE
I Tape	6	NITAPE

Changes

When changes are to be made to the standard peripheral device assignments, a new BRT must be produced. The first 16 instructions in the EXEC routine, referenced in "Standard Peripheral Assignments," must be altered as desired.

GENERAL OPERATING PROCEDURES

The LP H source-program deck, as supplied to the user, is set up to make a Fortran BRT. Directions for this procedure are given in the manual Fortran Compiler D (Order No. 027) under the "go-later" mode of operation.

When the BRT has been produced, the following general operating procedures apply:

1. Put the BRT on drive zero in "protect" and rewound status.
2. Put the run deck in the card reader "Run Deck," (see below).
3. Cycle up the printer, card reader, and card punch.
4. Put work tapes on drives 1, 2, 3, 4, in "permit" status.
5. If the matrix file is to be read from tape, put this tape on drive 3 in "protect" status. After INPUT is finished, this tape must be replaced by a work tape; a HALT agendum card should follow the INPUT card.
6. Follow standard starting procedures for a go-later run.
7. When the STOP agendum card is read, the LP run is finished. The LP H system exits to the Fortran monitor.
8. Remove the BRT and input data tapes, the cards from the reader, and the printed output and punched cards. Return these items to the programmer.

Programmed Halts

LP H has no programmed halts except the one caused by the HALT agendum card. When errors occur during a problem run, a message is given and the next problem is processed.

Error Stops

If an error occurs during a run and the machine stops, the M tape should be edited, giving a listing of all information computed up to the point of breakdown. This is done as follows:

1. Form a run deck containing the agendum call card OUTPUT. Specify the OLD option.
2. Follow the general operating procedures given above.

SENSE Switch Settings

If SENSE switch 2 is turned on during NORMAL, DO. PCR, or DO. PLP, control exits at the end of the current iteration, and a full-solution printout is put on the M tape. Then the next agendum card is read.

If SENSE switch 4 is turned on during INPUT, then the next agendum card is read and executed when INPUT has finished, whether or not nonprohibitive errors have been found in the input data; else nonprohibitive errors found in the input data generate an internal call to OUTPUT for an error listing, after which the next problem is processed.

Run Deck

The cards mentioned in step (2) under "General Operating Procedures" consist of the following:

1. Console Call card. The format for this card is:

   * LP H ΔΔ 010 * ΔΔ P ΔΔΔΔ *
   ↑
   Column 1

2. Blank card.
3. Agendum and data cards for the first problem to be solved. An OUTPUT card should be included for each problem run, permitting the next problem to be processed even if prohibitive errors are detected in the current problem. When a prohibitive error (which prevents further processing) occurs, the EXEC monitor generates an internal call to OUTPUT, giving an error listing. Then the EXEC monitor reads into the output stacker successive agendum This card should be for the current problem rather than for the next problem.
4. Agendum and data cards for second problem to be solved, etc.
5. STOP agendum card.
6. *ENDATA card.
7. *ENDRUN card.

The above cards are shown in Figure 4-1. For any given problem, if the matrix file is embedded in the agendum file as in this illustration, the agendum PARAMS must be used to assign the card reader as logical device I (see Figure 2-1).

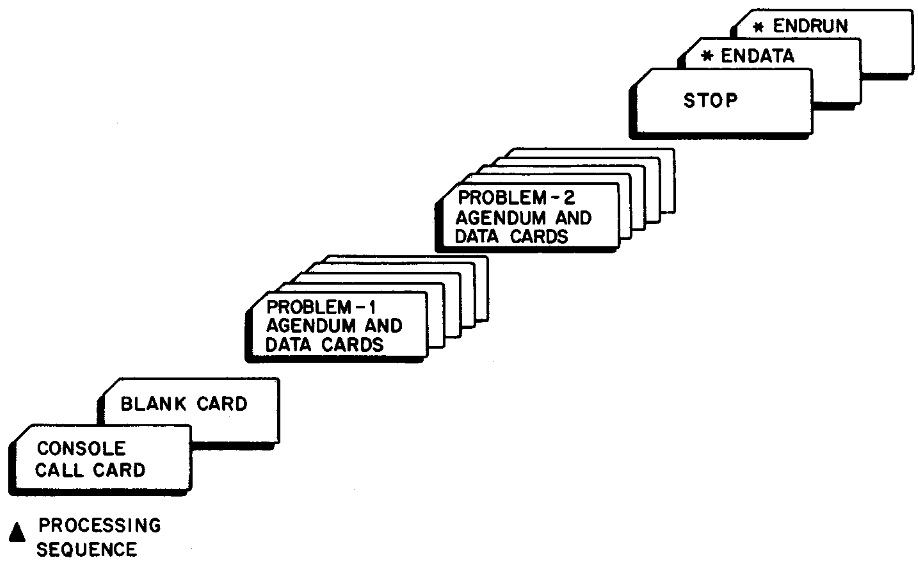


Figure 4-1. Standard Run Deck

MESSAGES

Each time an agendum is called, the name of that agendum is printed. In addition, intermediate messages are given to indicate the progress of the run. All messages are printed by OUTPUT. Messages, together with explanations and suggested actions, are listed in Table 4-4. The messages are associated with their particular agenda, and the agenda are listed in alphabetic order.

Table 4-4. Messages and Explanations

Agendum	Message	Explanation/Action
BRANGE	AGENDUM CARD ERRORS	The information on the agendum card last read violates the specifications for that agendum.
	SOLUTION IS INFEASIBLE	The current solution is infeasible. This agendum operates on an optimal solution.

Table 4-4 (cont). Messages and Explanations

Agendum	Message	Explanation/Action
	VECTOR REJECTED (vector name)	The listed rows were specified to be ranged during BRANGE but were not found in the row-names list.
CRANGE	AGENDUM CARD ERROR	See "Agendum card error" for BRANGE.
	SOLUTION IS INFEASIBLE	See "SOLUTION IS INFEASIBLE" for BRANGE.
	SOLUTION IS NOT OPTIMAL	During CRANGE or DO. PCR, the solution was found not to be optimal.
	VECTOR REJECTED	The listed column name (vector) was specified to be ranged during CRANGE but was not found in the basis.
DO. PCR	AGENDUM CARD ERROR	See "AGENDUM CARD ERROR" for BRANGE.
	MAXIMUM ERROR ROW b a	The current solution has been checked for the maximum error allowed in each row by the specified tolerance and has been found to exceed it. Parameter b is the actual error; a specifies the row in which it occurred.
	PHI AT MAXIMUM	Phi has reached its absolute maximum, or the maximum value of phi specified on the agendum card has been reached. See DO. PCR agendum in Section III.
	PHI IS UNBOUNDED	The value of phi can be increased without limit yet no change in basis is required to maintain optimality. See DO. PCR agendum in Section III.
	RHS MISSING: xxxxxx	The named right-hand side cannot be found on the A tape.
	SOLUTION IS INFEASIBLE	See "SOLUTION IS INFEASIBLE" for BRANGE.
	SOLUTION IS NOT OPTIMAL	See "SOLUTION IS NOT OPTIMAL" for CRANGE.
DO. PLP	AGENDUM CARD ERROR	See "AGENDUM CARD ERROR" for BRANGE.
	MAXIMUM ERROR ROW	See "MAXIMUM ERROR ROW" for DO. PCR.
	RHS MISSING: xxxxxx	See "RHS MISSING:" for DO. PCR.
	SOLUTION IS INFEASIBLE	See "SOLUTION IS INFEASIBLE" for BRANGE.
	THETA AT MAXIMUM	Theta has reached an absolute maximum, or the maximum value specified on the agendum card has been reached. See the DO. PLP agendum in Section III.

Table 4-4 (cont). Messages and Explanations

Agendum	Message	Explanation/Action
	THETA IS UNBOUNDED	The value of theta can be increased without limit, yet no change in basis is required to maintain feasibility. See the DO. PLP agendum in Section III.
GETOFF	AGENDUM CARD ERROR	See "AGENDUM CARD ERROR" for BRANGE.
INPUT	ANNOUNCE CARD ERROR	A required announce card is not in the problem data deck, or is out of place, or two such are contiguous. If only one problem is being processed, an error listing is given, the processing stops, and the program exits to the Fortran monitor. If more than one problem is being processed, an error listing is given, processing of the current problem stops, and the next problem is processed.
	DUPLICATE COLUMN NAME (vector name)	The indicated name has already been submitted at least once as a vector name.
	DUPLICATE ROW NAME (row name)	Two row-name cards contain the same row name. A row corresponding to the first such card is created, and the second card is ignored.
	ELEMENT GIVEN AGAIN (column name, row name, element value)	Two values were given for the same column element, or right hand-side element. The first value of the element is used; the second is ignored and is displayed in the message.
	ZERO ROW PROBLEM FORMED	No row-name cards could be found for the current problem.
	AGENDUM CARD ERROR	See "AGENDUM CARD ERROR" for BRANGE.
	CONSTRAINT ERROR (row name & constraint indicator)	The listed card has an invalid punch in column 12. The column is assumed to be blank; the row is considered to be an equality constraint.
INVERT	BAD H-REGION	The current H region contains names that are not valid vector names. If INVERT finds bad names in the H region, it replaces them with artificials. If the routine that periodically checks the solution finds bad names in the H region, it calls INVERT.

Table 4-4 (cont). Messages and Explanations

Agendum	Message	Explanation/Action
	BASIS SINGULAR xxxxxx	The current basis is singular. This means that the basis vector xxx is a nontrivial linear combination of other basis vectors. Singularity is usually caused by digital error. To prevent singularity, an artificial vector is put in the basis in place of xxx. The updated representation of xxx is displayed. A listing of the vectors already brought into the basis is written.
	RHS MISSING	See "RHS MISSING" for DO. PCR.
LDHREG	AGENDUM CARD ERROR	See "AGENDUM CARD ERROR" for BRANGE.
LOAD	VECTOR REJECTED	The listed columns (vectors) are specified to go into the basis during LOAD, but no place can be found for them.
NORMAL	AGENDUM CARD ERROR	See "AGENDUM CARD ERROR" for BRANGE.
	FULL REJECT LIST (5 vector names & 5 Z values)	The listed vectors have been successively rejected during one iteration as candidates to enter the basis for at least one of the following reasons: (1) The pivot element is too small. (2) The change in Z is positive and no infeasibility would be removed.
	MAXIMUM ERROR ROW	See "MAXIMUM ERROR ROW" for DO. PCR.
	MONOTON-ICITY ERROR	If the objective value has changed in the wrong direction during ten iterations, so as to produce the wrong sign, the message is given. This error is usually caused by accumulated digital error.
	PROBLEM IS INFEASIBLE	The problem has no feasible solution. Either the constraints are inconsistent or there exists no valid solution in nonnegative numbers. If this message occurs after a feasible solution has been found, the trouble is usually digital error. Increasing the tolerance value of QTOLZE may remove the difficulty.
	RHS MISSING	See "RHS MISSING" for DO. PCR.
	SOLUTION IS FEASIBLE	The current solution is feasible.
	SOLUTION IS INFEASIBLE	See "SOLUTION IS INFEASIBLE" for BRANGE.
	SOLUTION IS OPTIMAL	An optimal solution to the problem has been found.

Table 4-4 (cont). Messages and Explanations

Agendum	Message	Explanation/Action
	SOLUTION IS UNBOUNDED	No finite optimal solution exists. The objective value can be made as large in absolute value as desired.
	TOO MANY ITERATIONS	The total number of iterations is greater than the contents of NMAXIT in the communication region.
PARAMS	AGENDUM CARD ERROR	See "AGENDUM CARD ERROR" for BRANGE.
RESTART	AGENDUM CARD ERROR	See "AGENDUM CARD ERROR" for BRANGE.

SECTION V
OUTPUT

This section presents reproductions and explanations of the printout headings produced by LP H. The printout headings indicate what intermediate and final information is made available by the system.

PRINTOUT HEADINGS

The explanations of printout headings are given in Table 5-1. For a further explanation of the terms given in Table 5-1, consult reference 5 or 6 of Appendix A.

The headings through "Current D/J Value" occur in iteration printouts. These are short output summaries giving an indication of the progress of the problem, iteration by iteration. Such a printout is given for each iteration performed during the three agenda NORMAL, DO.PLP and DO.PCR. These agenda are called iterating algorithms.

The other headings (after "Current D/J Value") occur on what are called full-solution printouts. These printouts contain more complete information than do the iteration printouts. They occur at any exit of NORMAL, DO.PLP, or DO.PCR. During DO.PLP, the number in the "Current D/J Value" column is the current value of theta. The column under "B(I)" is the composite right-hand side (see DO.PLP in Section III). During DO.PCR, the number in the "Current D/J Value" column is the current value of phi. The column under "A(TAU, J) is the composite objective function (see DO.PCR in Section III).

Table 5-1. Printout Headings and Explanations

Printout Heading	Explanation
TOTAL ITERS	Total number of iterations done since start of problem.
NO. ETAS	Total number of eta transformations performed since start of problem or since last INVERT.
ETA REC	Number of eta records created since start of problem or since last INVERT.
ROW ID	Row number of the objective function being optimized.

Table 5-1 (cont). Printout Headings and Explanations

Printout Heading	Explanation
INFEASIBILITY SUM/ CURRENT VALUE OF Z	Measure of the infeasibility still present. It is replaced by current value of the objective function when solution is feasible.
CHOSEN VECTOR	Name of the column entering the basic solution during the current iteration.
VECTOR REMOVED	Name of the column being replaced in the solution during the current iteration.
NEG D/J * S	Number of negative D/J values found at the start of the current iteration.
NO. OF INFEAS	Number of infeasibilities still present at start of the current iteration.
CURRENT D/J VALUE	Value of D/J for chosen (j^{th})column
FULL SOLUTION PRINT ROW INFORMATION	This heading is the general heading under which the following special headings are listed: 1. ROW (I) 2. SLACK (I) 3. PI (I) 4. B (I)
ROW (I)	Names of the rows of the problem.
SLACK (I)	Level of the slack corresponding to row (I).
PI (I)	The pricing vector.
B (I)	The original right-hand side used.
COLUMN INFORMATION	This heading is the general heading under which the following specific headings are listed: 1. COLUMN 2. BETA (J) 3. D/J 4. A (TAU, J)
COLUMN	Names of structural vectors in the A matrix.
BETA (J)	Level of vector (J) in solution.
D/J	Relative cost factor for vector (J).
A (TAU, J)	Original cost of vector.

SECTION VI
SAMPLE PROBLEM

This section describes how a sample LP problem, called the Breakfast Food Problem, is set up and solved using Honeywell's LP H System. Included in the discussion is a statement of the problem, how it is formulated, and what input is needed to obtain the solution. Solution printouts are also included.

BREAKFAST FOOD PROBLEM

The Wholesome Cereal Company wishes to mix four breakfast foods in such a way as to achieve certain dietary properties while minimizing the cost of the resulting mixture. The dietary properties involved are the sodium, protein, and caloric constituents in each breakfast food. For each breakfast food, the cost and the dietary properties are shown in Table 6-1.

Table 6-1. Cereal Mixture Data

	Crispies	Crunchies	Crackles	Chortles	Cereal Mixture
Cost	4.0	7.0	8.0	6.0	
Calories	150.0	140.0	170.0	160.0	=150.0
Sodium	0.1	0.1	0.3	0.3	$\leq$ 0.2
Protein	2.0	4.0	5.0	3.0	$\geq$ 3.0

As shown in the table, the cereal mixture is to have exactly 150 calories, the sodium content is not to exceed 0.2 grams, and the protein content must be at least 3.0 grams. These and other data in the table are punched on data cards as described on the following pages.

FORMULATING THE BREAKFAST FOOD PROBLEM

The first step is to put the problem in mathematical form, using the data in Table 6-1. According to the LP terminology defined in Section I, the objective coefficients (c_j in equation 1-1) are the costs of each breakfast food in Table 6-1: 4.0 for CRISPIES, 7.0 for CRUNCHIES, 8.0 for CRACKLES, and 6.0 for CHORTLES. Each activity level, x_j, is the unknown amount of the corresponding breakfast food required to minimize the cost of the resulting mixture while maintaining dietary specifications. The objective value (the Z in equation 1-1) is the unknown minimum cost of the mixture. In mathematical form, the objective function is:

(6-1) $$4.0x_1+7.0x_2+8.0x_3+6.0x_4 = Z$$

where 4.0 = unit cost of CRISPIES

x_1 = amount of CRISPIES needed

7.0 = unit cost of CRUNCHIES

x_2 = amount of CRUNCHIES needed

8.0 = unit cost of CRACKLES

x_3 = amount of CRACKLES needed

6.0 = unit cost of CHORTLES

x_4 = amount of CHORTLES needed

The I/O coefficients, a_{ij}, are the caloric, sodium, and protein contents of each breakfast food in Table 6-1. The constraint values are the values shown in the cereal mixture column of the table. In mathematical form, the constraints are:

(6-2)
$$150.0x_1+140.0x_2+170.0x_3+160.0x_4 = 150.0$$
$$0.1x_1+ 0.1x_2+ 0.3x_3+ 0.3x_4 \leq 0.2$$
$$2.0x_1+ 4.0x_2+ 5.0x_3+ 3.0x_4 \geq 3.0$$

where 150.0 = calories per unit of CRISPIES

140.0 = calories per unit of CRUNCHIES

170.0 = calories per unit of CRACKLES

160.0 = calories per unit of CHORTLES

0.1 = sodium per unit of CRISPIES

0.1 = sodium per unit of CRUNCHIES

0.3 = sodium per unit of CRACKLES

0.3 = sodium per unit of CHORTLES

2.0 = protein per unit of CRISPIES

4.0 = protein per unit of CRUNCHIES

5.0 = protein per unit of CRACKLES

3.0 = protein per unit of CHORTLES

Finally, since the amount of each breakfast food used must be equal to or greater than zero, the nonnegativity condition (enforced automatically by the system) for this example is:

(6-3) $$x_j \geq 0 \text{ where } j = 1,2,3,4$$

Equations (6-1) through (6-3) constitute the mathematical formulation of this sample problem.

CODING THE BREAKFAST FOOD PROBLEM DATA

Coding the breakfast food problem data involves writing a matrix of detached coefficients, taken from the objective function (6-1) and the constraints (6-2). Rewriting (6-1) and (6-2) gives this:

(6-4)

$$
\begin{aligned}
4.0x_1 + 7.0x_2 + 8.0x_3 + 6.0x_4 &= Z \\
150.0x_1 + 140.0x_2 + 170.0x_3 + 160.0x_4 &= 150.0 \\
0.1x_1 + 0.1x_2 + 0.3x_3 + 0.3x_4 &\leq 0.2 \\
2.0x_1 + 4.0x_2 + 5.0x_3 + 3.0x_4 &\geq 3.0
\end{aligned}
$$

The numerical values are taken from (6-4) and are rewritten as in (6-5). The matrix of detached coefficients, together with the right-hand side of the constraints, constitutes the input tableau. Included are the chosen column names and row names to be used in coding.

(6-5)

Column Names	CRISPI	CRUNCH	CRACKL	CHORTL	ONEΔΔΔ	Right-Hand Side Name
	4.0	7.0	8.0	6.0		COST
	150.0	140.0	170.0	160.0	150.0	CALORY
	0.1	0.1	0.3	0.3	0.2	SODIUM
	2.0	4.0	5.0	3.0	3.0	PROTEN
	Matrix				Right-hand Side	Row Names

The coding for (6-5) is shown in Figure 6-1. This coding conforms to the problem-data specifications in Section II.

AGENDA REQUIRED

The three agenda INPUT, NORMAL, and OUTPUT are required to solve the problem. Consult Section III to see how the call cards for these agenda are coded.

INPUT PROCEDURE

The complete input deck (i.e., agendum file with embedded matrix file) is shown in Figure 6-2.

PROBLEM THE BREAKFAST FOOD PROBLEM DATE / / PAGE ___ OF ___

INDICATOR FIELD (1–6)	COLUMN NAME (7–12)	ROW TYPE (12)	ROW NAME (13–18)	VALUE OF ELEMENT: SIGN / INTEGRAL PART / DEC. PT. / FRACTIONAL PART (19–30)	REMARKS (31–72)	CARD IDENTIFICATION (73–80)
ROWΔID						
		Ø	COST			
		Ø	CALORY			
		+	SODIUM			
		-	PROTEN			
MATRIX						
	CRISPI		COST	4.Ø		
	CRISPI		CALORY	15Ø.Ø		
	CRISPI		SODIUM	Ø.1		
	CRISPI		PROTEN	2.Ø		
	CRUNCH		COST	7.Ø		
	CRUNCH		CALORY	14Ø.Ø		
	CRUNCH		SODIUM	Ø.1		
	CRUNCH		PROTEN	4.Ø		
	CRACKL		COST	8.Ø		
	CRACKL		CALORY	17Ø.Ø		
	CRACKL		SODIUM	Ø.3		
	CRACKL		PROTEN	5.Ø		
	CHORTL		COST	6.Ø		
	CHORTL		CALORY	16Ø.Ø		
	CHORTL		SODIUM	Ø.3		
	CHORTL		PROTEN	3.Ø		
RHSIDE						
	ONE		CALORY	15Ø.Ø		
	ONE		SODIUM	Ø.2		
	ONE		PROTEN	3.Ø		
ENDATA						

Figure 6-1. Coding of Input Data for Breakfast Food Problem

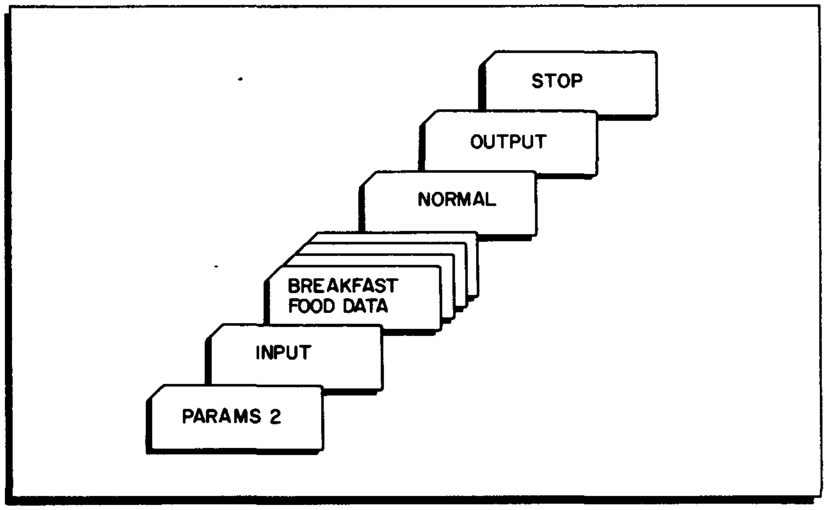


Figure 6-2. Agendum File for the Breakfast Food Problem

There are three basic steps in the input procedure:

1. Load the LP H System.
2. Put the input deck (Figure 6-2) in the card reader.
   a. Agendum cards must appear in the order shown.
   b. Each agendum card must be followed by the data cards required for that agendum, again as shown.
   c. When more than one problem or version of the same problem is being run (as in the two variations of this problem described later), the variants or different problems are stacked. In all cases, as shown in Figure 6-2, a card having the word STOP punched in columns 1-4 should appear at the end of input deck, following the last problem to be processed.
3. Start the program.

MESSAGES

Each time an agendum is called, the name of that agendum is printed. In addition, intermediate messages are given to indicate the progress of the run. These messages, together with explanations and suggested actions, are shown in Table 4-4.

OUTPUT

The output for this problem (from INPUT and NORMAL) is shown in Figure 6-3.

```
INPUT        LIST              CHECK
INPUT SUMMARY: ELEMENTS COLUMNS ROWS +SLACKS -SLACKS
                     16        4    4       1       1

VECTORS IN PROBLEM WITH ELEMENT COUNTS
   CRISPI      4 CRUNCH     4 CRACKL     4 CHORTL     4

NUMBER OF ELEMENTS IN EACH MATRIX ROW
  COST         4 CALORY     4 SODIUM     4 PROTEN     4


NORMAL
SOLUTION IS INFEASIBLE
   TOTAL    NO.    ETA    ROW    INFEASIBILITY SUM/     CHOSEN    VECTOR     NEG     NO. OF           CURRENT
   ITERS    ETAS   REC    ID.    CURRENT VALUE OF Z     VECTOR    REMOVED    D/J*S   INFEAS           D/J VALUE

       0      1     0      1      153.0000000                                  0         2      .0000000000E-99
       1      2     0      1      48.00000000            CRACKL    - PROTEN     4         1      -175.0000000
SOLUTION IS FEASIBLE
   TOTAL    NO.    ETA    ROW    INFEASIBILITY SUM/     CHOSEN    VECTOR     NEG     NO. OF           CURRENT
   ITERS    ETAS   REC    ID.    CURRENT VALUE OF Z     VECTOR    REMOVED    D/J*S   INFEAS           D/J VALUE

       2      3     0      1      5.268292600            CRISPI    000000       4         0      -82.00000000
SOLUTION IS OPTIMAL

FULL SOLUTION PRINT
ROW INFORMATION
 ROW(I)           SLACK(I)                PI(I)               B(I)

  COST        -5.268292600            1.000000000          .0000000000E-99
  CALORY       .0000000000E-99       -.9756096800E-02      150.0000000
+ SODIUM       .3170731600E-01        .0000000000E-99      .2000000000
- PROTEN       .0000000000E-99       -1.268292800          3.000000000
COLUMN INFORMATION
 COLUMN          BETA(J)                 D/J                 A(TAU,J)

 CRISPI       .5853658000            .0000000000E-99       4.000000000
 CRUNCH       .0000000000E-99        .5609753000           7.000000000
 CRACKL       .3658536800            .0000000000E-99       8.000000000
 CHORTL       .0000000000E-99        .6341462000           6.000000000
```

Figure 6-3. Output for the Breakfast Food Problem

The vectors, CALORY, SODIUM, and PROTEN which appear in Figure 6-3 are respectively the artificial and the positive and negative slacks introduced by LP H. LP H assigns a name to each slack and each artificial introduced.

POST-OPTIMAL ANALYSES

When the initial optimal soultion shown in Figure 6-3 has been obtained, the analyst may want to range the objective coefficients and the right-hand side, and he may want to change one or more of their values in order to see how the changes affect the initial optimal solution. These post-optimal changes to the Breakfast Food Problem are described below.

Objective Function (Cost Row) Ranging

This post-optimal procedure is used to test the limits to which the objective coefficients of the current basis vectors can be changed one at a time without producing nonoptimality. The agendum used is CRANGE. Consult Section III to see how the call card for this agendum is punched. The output from the CRANGE agendum is shown in Figure 6-4.

CRANGE

VECTOR NAME	CURRENT LEVEL	ORIGINAL COST	COST RANGES MINIMUM COST	MAXIMUM COST	INCOMING VECTOR AT MIN	AT MAX
+ SODIUM	0.03170731		-4.03508771	4.72727272	CRUNCH	CHORTL
CRISPI	0.58536585	4.00000000	NO BOUND	4.89655172		CHORTL
CRACKL	0.36585365	8.00000000	4.53333333	8.71875000	-PROTEN	CRUNCH

Figure 6-4. CRANGE Printout

Right-Hand Side Ranging

This post-optimal procedure is used to test the limits to which the right-hand-side elements can be changed one at a time without producing infeasibility. The agendum used is BRANGE. Consult Section III to see how the call card for this agendum is punched. The output from the BRANGE agendum is shown in Figure 6-5 below:

BRANGE — RIGHT-HAND SIDE RANGES

ROW NAME	CURRENT RHS VAL	PI VALUE	MINIMUM VALUE	MAXIMUM VALUE	OUTGOING VECTOR AT MIN	AT MAX
0 CALORY	150.00000000	-0.00975610	102.00000000	25.00000000	CRISPI	CRACKL
+ SODIUM	0.20000000	-	0.16829268	NO BOUND	+ SODIUM	
- PROTEN	3.00000000	-1.26829268	2.00000000	3.46428571	CRACKL	+ SODIUM

Figure 6-5. BRANGE Printout

Objective Coefficient Change

Assume that, due to a bad wheat crop, the cost of CRISPI in Table 6-1 is expected to change from 4.0 to 4.5 and that the Wholesome Cereal Company wishes to determine the effect of this cost change on the initial optimal solution. The agendum used is DO. PCR. In addition, a new row-name card and a new matrix-element card must be introduced into the original input data deck. The row-name card indicates that a change-cost row is to exist in the matrix; the matrix-element card indicates that the new element is 0.5 in row CHANGE of column CRISPI. The agendum INPUT is redone, as illustrated in Figure 6-13.

The DO. PCR agendum card is coded for the change as shown in Figure 6-6.

INDICATOR FIELD (1–6)	COLUMN NAME (7–12)	ROW TYPE	ROW NAME (13–18)	VALUE OF ELEMENT: SIGN (19)	INTEGRAL PART (20–23)	DEC. PT. (24)	FRACTIONAL PART (25–30)	REMARKS (31–72)	CARD IDENTIFICATION (73–80)
DO.PCR	ΔΔ2ΔΔΔ		1.ϕ						

Figure 6-6. DO. PCR Agendum Card Format

The "2" in Figure 6-6 is the index of the change-cost row. The "1.0" is the maximum value of phi, limiting the maximum value of CRISPI's objective coefficient to 4.0 + ϕ_{max} 0.5 = 4.0 + 1(0.5) = 4.5.

The output from the DO. PCR agendum is shown in Figure 6-7.

```
DO.PCR
PHI IS AT MAXIMUM
ROW INFORMATION
 ROW(I)      SLACK(I)          PI(I)              B(I)
0 COST       -5.26829268        1.00000000         -
0 CHANGE     -0.29268292        1.00000000         -
0 CALORY         -             -0.01585365      150.00000000
+ SODIUM      0.03170731           -               0.20000000
- PROTEN         -             -1.06097560         3.00000000
COLUMN INFORMATION
 COLUMN        BETA(J)          D/J               A(TAU,J)
  CRISPI      0.58536585           -               4.00000000
  CRUNCH         -              0.53658536         7.00000000
  CRACKL      0.36585365           -               8.00000000
  CHORTL         -              0.28048780         6.00000000
```

Figure 6-7. DO. PCR Printout

The change row-name card is coded as shown in Figure 6-8.

INDICATOR FIELD (1–6)	COLUMN NAME (7–12)	ROW TYPE (12)	ROW NAME (13–18)	SIGN (19)	INTEGRAL PART (20–23)	DEC. PT. (24)	FRACTIONAL PART (25–30)	REMARKS (31–72)	CARD IDENTIFICATION (73–80)
		Φ	CHANGE						

Figure 6-8. Change-Row-Name Card Format

The new matrix-element card is coded as shown in Figure 6-9.

INDICATOR FIELD (1–6)	COLUMN NAME (7–12)	ROW TYPE (12)	ROW NAME (13–18)	SIGN (19)	INTEGRAL PART (20–23)	DEC. PT. (24)	FRACTIONAL PART (25–30)	REMARKS (31–72)	CARD IDENTIFICATION (73–80)
	CRISPI		CHANGE		Φ	.	5		

Figure 6-9. Change-Matrix-Element Card Format

Right-Hand-Side Change

Assume that, due to a demand for low-calorie foods, the Wholesome Cereal Company decides to reduce the caloric content of its cereal mixture from 150 to 135. The company wishes to determine the effect of this change on the initial optimal solution. The agendum used is DO. PLP. Right-hand-side element cards for a new right-hand-side (which is used by DO. PLP as the change right-hand side), showing the change from 150 to 135, must be introduced into the original input data deck. The DO. PLP agendum card is coded as shown in Figure 6-10.

INDICATOR FIELD (1–6)	COLUMN NAME (7–12)	ROW TYPE (12)	ROW NAME (13–18)	SIGN (19)	INTEGRAL PART (20–23)	DEC. PT. (24)	FRACTIONAL PART (25–30)	REMARKS (31–72)	CARD IDENTIFICATION (73–80)
DO.PLP	TWOΔΔΔ		ΔΔΔ1.Φ						

Figure 6-10. DO. PLP Agendum Card Format

The "TWO" in Figure 6-10 indicates that a new right-hand side is to be used and that 1.0 is the maximum value of theta. A new right-hand side named TWO must be added to the A tape. In Figure 6-1, the "ONE" used for the RHSIDE data cards is the name of the original, unchanged right-hand side.

The output from the DO. PLP agendum is shown in Figure 6-11.

```
DO.PLP
THETA IS AT MAXIMUM
ROW INFORMATION

 ROW(I)        SLACK(I)          PI(I)             B(I)
O COST        -5.12195121       1.00000000          -
O CHANGE      -0.20121951        -                  -
O CALORY       -               -0.00975610      135.00000000
+ SODIUM       0.02804878        -                 0.20000000
- PROTEN       -               -1.26829268         3.00000000

COLUMN INFORMATION

 COLUMN         BETA(J)          D/J               A(TAU,J)
  CRISPI       0.40243902        -                 4.00000000
  CRUNCH        -               0.56097560         7.00000000
  CRACKL       0.43902440        -                 8.00000000
  CHORTL        -               0.63414634         6.00000000
```

Figure 6-11. DO. PLP Printout

The new right-hand-side element data card is coded as shown in Figure 6-12.

INDICATOR FIELD (1–6)	COLUMN NAME (7–12)	ROW TYPE	ROW NAME (13–18)	VALUE OF ELEMENT: SIGN (19)	INTEGRAL PART (20–23)	DEC. PT. (24)	FRACTIONAL PART (25–30)	REMARKS (31–72)	CARD IDENTIFICATION (73–80)
	TWOΔΔ		CALORY	-	15	.	Ø		

Figure 6-12. Change Right-Hand-Side Card Format

Since the calorie change is from 150 to 135, -15.0 is the "Value of Element" used in Figure 6-12.

The input data deck shown in Figure 6-13 for all of the post-optimal analyses described above consists of the original deck shown in Figure 6-1, together with the decks described in Figure 6-8 (the change row-name card), in Figure 6-9 (the change matrix-element card) and in Figure 6-12 (the change right-hand-side element card). In Figure 6-1, the change row-name card is placed between the 0-COST and 0-CALORY row-name cards. The change matrix-element card is placed between the CRISPI, COST and the CRISPI, CALORY matrix-element cards. The change right-hand-side element card is placed between the RHSIDE card and the ONE, CALORY card.

The entire post-optimal run deck is shown in Figure 6-13. Included are the input data cards and the agendum cards DO. PCR and DO. PLP from Figures 6-6 and 6-10.

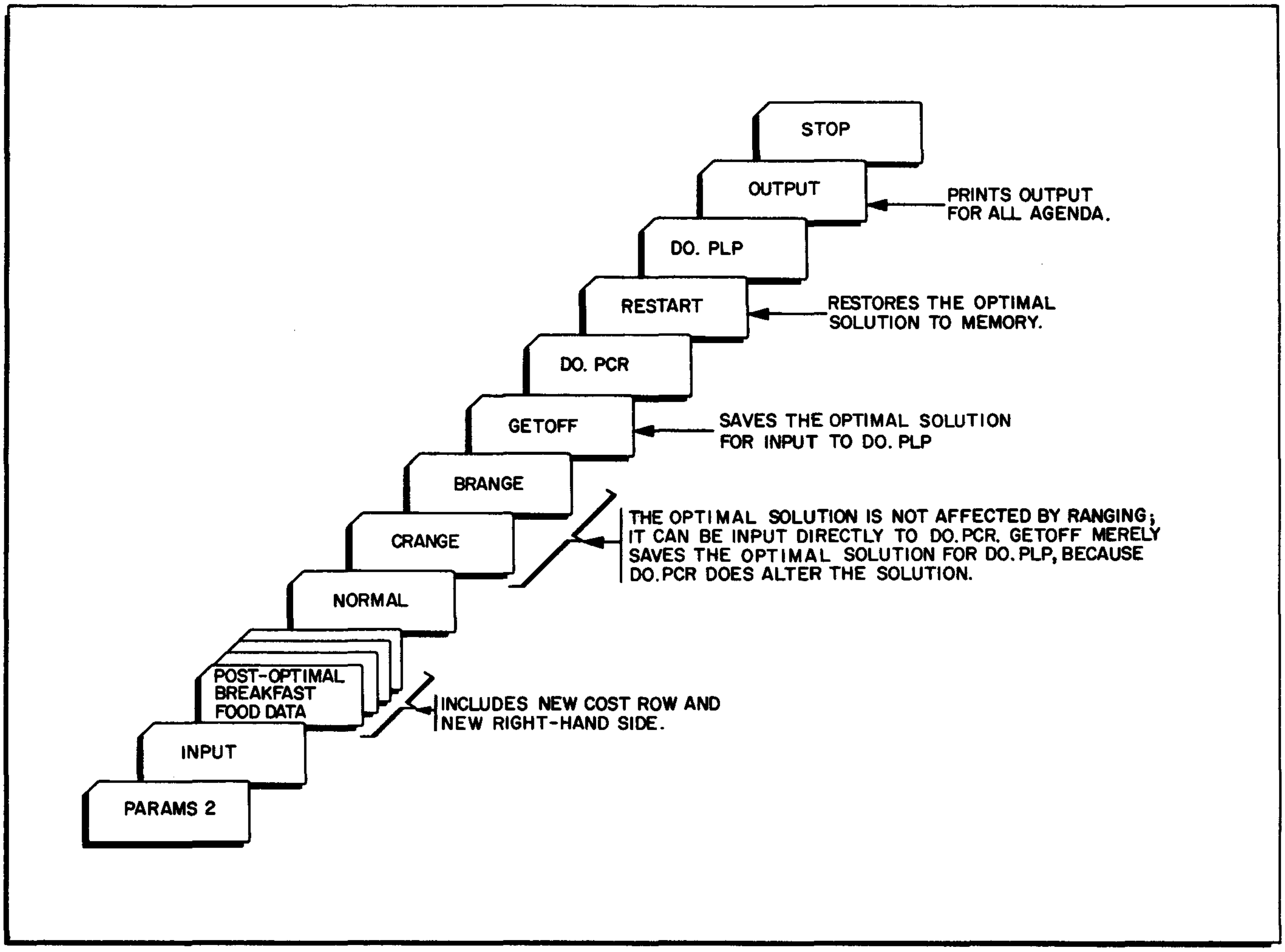


Figure 6-13. Post-Optimal Run Deck for the Breakfast Food Problem

APPENDIX A

REFERENCES

1. Dantzig, G. B.: Linear Programming & Entensions, Princeton University Press, 1963. This encyclopedic text deals with the simplex algorithm, its many variants, operations research, and mathematical programming in general.

2. Gass, S. I. and Riley, V.: Bibliography on Linear Programming and Related Techniques, Johns Hopkins Press, Baltimore, 1958. This bibliography includes more than 1000 items with abstracts arranged by application type.

3. Gass, S. I.: Linear Programming, McGraw-Hill Book Company, New York, N. Y., 1964.

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

5. Honeywell: Advanced Linear Programming System (ALPS), Order No. 171. Section I, Elements of Linear Programming, assumes no previous experience of linear programming. After an informal geometric treatment of convexity, pricing conventions, optimality, sensitivity analyses, and parametric programming, a machine scheduling application is discussed in detail and is solved by hand using the standard simplex algorithm. A discussion of the revised simplex algorithm for digital computers is included in Section II.

6. Orchard-Hays, W.: Matrices, Elimination, and the Simplex Method, CEIR, Arlington, Virginia, 1961. A concise mathematical description of the composite simplex algorithm, sensitivity analyses, and parametric programming.

COMPUTER-GENERATED INDEX

COMPUTER-GENERATED INDEX

NOTES

NOTES

HONEYWELL EDP TECHNICAL PUBLICATIONS
USERS' REMARKS FORM

TITLE: SERIES 200
LINEAR PROGRAMMING SYSTEM H (BASIC)
SOFTWARE MANUAL

DATED: MAY, 1966

FILE NO: 123.8305.001H.0-377

ERRORS NOTED:

Fold

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME ______________________ DATE ____________

COMPANY ______________________

TITLE ______________________

ADDRESS ______________________

____

Cut Along Line

FIRST CLASS
PERMIT NO. 39531
WELLESLEY HILLS
MASS.

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

POSTAGE WILL BE PAID BY

HONEYWELL
ELECTRONIC DATA PROCESSING DIVISION
60 WALNUT STREET
WELLESLEY HILLS, MASS. 02181

ATT'N: TECHNICAL COMMUNICATIONS DEPARTMENT

Cut Along Line

Honeywell
ELECTRONIC DATA PROCESSING