جمهورية العربية المتحدة

معهد التخطيط القومي

Memo. No. 486

The Simplex Multipliers and the Shadow prices in Linear Programming

By

Dr. Roshdi Amer

September 1964

Operations Research Centre

القاهرة

٣ شارع محمد مظهر - بالزمالك

1. Introduction

The attempt to describe the simplex algorithm in matrix form may appear to be unjustified since the standard simplex method is much more simple from the theoretical point of view; also numerically both methods appear to be identical since both of them pass through the same set of feasible basic solutions. But in fact there are a lot of advantages of the matrix form or the method of multipliers (which are sometimes called "shadow prices" because of their meaning in some economical applications).

(1) The first obvious advantage, is that we are dealing with square tables whose size depends only on the number of basic variables (which is equal to the number of restrictions); so that if we have a large number of variables but only a few restrictions (5 variables and 2 equations in the example solved in section 5), a lot of necessary core storage and computation time can be saved. If the number of variables is small while the number of restrictions is large, the same advantage can be attained by solving the dual problem*)

(2) A further advantage is that the original data** of the problem will not be transformed through the pivoting operations of the simplex algorithm so that it will not be deformed by rounding errors; if at any stage of the calculations there is any doubt about the numerical accuracy of any table this table can be recalculated directly from the original data, this can always be done with the last table to ensure the accuracy of the final results.

*) See Memo. 483 "Basic principles in linear programming"

**) If the necessary storage for the original data is small (i.e. if the coefficients are defined by a small number of digits) it will be kept in core storage. For too large amount of data the method of multipliers is most useful, since these can be kept on auxiliary storage without affecting the speed of computation very much.

A very useful feature of the matrix form is that the shadow prices are available (these are not computed by the standard simplex algorithm); these, besides their practical meaning in some problems, can be used for any post-optimization studies, such as the effect of modifying the constraints of the problem on its optimal solution, or any other sensitivity analysis problem.

In problems where we are liable to meet degenerated situations*) it is almost necessary to use the method of multipliers since this is the only method where the lexicographic rule for treating degeneracy can be applied.

In large-scale mathematical programming problems, where the techniques of decomposition have to be applied, we cannot procede without the matrix form, not only because of the large amount of data to be handled but also because the shadow prices (or multipliers) provided by the master problem will be needed to determine the preference functions for the different sub-problems; in fact the iteration process of the method of decomposition cannot be initiated without assuming initial shadow prices (these can either be estimated from the practice of the economical situation in the problem, or if this is not possible zero values can be assumed on the expence of the computation time).

---

*) See Memo 483
The author also intends, in a further paper to introduce another method for treating degenaracy in the standard simpex algorithm.

2. The Simplex Algorithm in matrix form

Consider the linear programming problem with equations constraints :

Minimize the objective function

$$Z = C_1 x_1 + C_2 x_2 + \dots + C_j x_j + \dots + C_N x_N \qquad (1\text{-a})$$

subject to the equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1j}x_j + \dots + a_{1N}x_N &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2j}x_j + \dots + a_{2N}x_N &= b_2 \\ \vdots \qquad\quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\quad \vdots & \\ a_{M1}x_1 + a_{M2}x_2 + \dots + a_{Mj}x_j + \dots + a_{MN}x_N &= b_M \end{aligned} \qquad (1\text{-b})$$

Such that

$$x_1 \ , \ x_2 \ , \ \dots \ , \ x_j \ , \ \dots \ , \ x_N \geq 0 \qquad (1\text{-C})$$

This is equivalent to the following problem in an M-dimensional space;

Given N vectors :

$$P_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{M1} \end{bmatrix} , \ P_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{M2} \end{bmatrix} , \ \dots , \ P_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{Mj} \end{bmatrix} , \ \dots , \ P_N = \begin{bmatrix} a_{1N} \\ a_{2N} \\ \vdots \\ a_{MN} \end{bmatrix}$$

and the vector $Q = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix}$

It is then requried to determine a linear combination of the vectors $P_1, P_2, \dots, P_j, \dots, P_N$ with non-negative coefficients

$$x_1, x_2, \dots, x_j, \dots, x_N$$

such that the function

$$Z = c_1x_1 + c_2x_2 + \dots + c_jx_j + \dots + c_Nx_N = \text{minimum} \quad (2\text{-a})$$

and such that

$$x_1P_1 + x_2P_2 + \dots + x_jP_j + \dots + x_NP_N = Q \quad (2\text{-b})$$

This problem can be solved using the standard simplex method, by constructing a sequence of basic feasible solutions which are obtained by representing the given system (1-b) in a canonical form with respect to the basic variables in each step. In this section we shall study a method to construct the canonical system at any step k directly from the original system (1-b) or (2-b).

Assume that at step k of the simplex algorithm it is known that the basic variables are

$$x_{j_1}, x_{j_2}, \dots, x_{j_i}, \dots, x_{j_M}.$$

The system must thus be in canonical form with respect to these variables; i.e. the system at step k must have the following form:

$$Z = c'_1x_1 + c'_2x_2 + \dots + c'_jx_j + \dots + c'_Nx_N + z'_o \quad (3\text{-a})$$

$$x_1p'_1 + x_2p'_2 + \dots + x_jp'_j + \dots + x_Np'_N = Q' \quad (3\text{-b})$$

such that :

$$c'_{j_1} = 0, \quad c'_{j_2} = 0 , \ \dots , \ c'_{j_i} = 0, \ \dots , \ c'_{j_M} = 0 \qquad (3\text{-}c)$$

and

$$P'_{j_1} = U_1 , \ P'_{j_2} = U_2 , \ \dots , \ P'_{j_i} = U_i , \ \dots , \ P'_{j_M} = U_M \qquad (3\text{-}d)$$

where

$$U_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} , \quad U_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} , \quad \dots\dots , \quad U_M = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

In order to deduce the system (3-b) from the original system (2-b), consider the matrix B consisting of the column-vectors of the basic variables in the original system (these vectors are called the "basis" of the system).

$$B = (P_{j_1} , P_{j_2} , \dots , P_{j_i} , \dots , P_{j_M}) \qquad (4)$$

i.e. $|B| \neq 0$

This matrix must be regular and has an inverse B^{-1}.

Since

$$B^{-1} B = B^{-1} (P_{j_1} , P_{j_2} , \dots , P_{j_i} , \dots , P_{j_M})$$

$$= (U_1, U_2, \dots , U_i , \dots , U_M)$$

it is clear that

$$B^{-1}.P_{j_1} = U_1 , \ B^{-1}.P_{j_2} = U_2 , \ \dots , \ B^{-1}P_{j_i} = U_i, \ \dots , \ B^{-1}P_{jM} = U_M \qquad (5)$$

Thus, we can obtain the system (3-b) from the original system (2-b) by multiplying the system (2-b) by the matrix B^{-1}.

This matrix musut be recorded at each steip k; any vector of the system (3-b) can then be calculated using the following rule :

$$P'_j = B^{-1} P_j \qquad (6\text{-a})$$

Similarly for the constants vector

$$Q' = B^{-1} Q \qquad (6\text{-b})$$

However it is more convenient to record the vector Q' beside the matrix B^{-1} and to modify both of them in each step, since the elements of Q' will be used for choosing the pivot elements in the pivoting operations. The basic solution in step k is obtained also from the vector Q' according to the following rule :

$$\text{The basic variables, } x_{j_i} = b'_i \quad , i = 1,2 , \ldots , M \qquad (7)$$

and the non basic variables are given zero values

The basic solution will be feasible if each element b'_i of the vector Q' has a non-negative value; the feasibility consition can be written

$$Q' \geq 0 \qquad (8)$$

In order to test the basic feasible solution at step k for beeing an optimal solution, we need the coefficients of the non-basic variables in the objective function. The new equation of the objective function (3-a) is obtained by eleminating the basic variables from the original equation (2-a)

Assume that this elemination is done by multiplying the equations of the system (1-b) by multipliers

$$\pi_1 , \pi_2 , \dots , \pi_i , \dots , \pi_M$$

then each modified equation is substracted from the equation (1-a). The coefficients of the new equation can then be calculated using the rule

$$\left.\begin{aligned} c'_j &= c_j - \pi P_j \quad , \quad j = 1, 2, \dots , N \\ \text{where the vector} \\ \pi &= (\pi_1, \pi_2 , \dots , \pi_M) \end{aligned}\right\} (9)$$

The multipliers can be obtained by applying the condition (3-c) to the equations (9) for the coefficients of the basic variables,

$$0 = c'_{j_i} = c_{j_i} - \pi . P_{j_i}$$

$$\text{or} \quad c_{j_i} = \pi P_{j_i} \quad \text{for } i = 1,2, \dots , M \qquad (10)$$

Thes M equations (10) can be written in matrix form

$$(c_{j_1} , c_{j_2} , \dots , c_{j_i} , \dots , c_{j_M})$$

$$= \pi . (P_{j_1} , P_{j_2} , \dots , P_{j_i} , \dots , P_{j_M})$$

$$\text{or} \quad \gamma = \pi . B \qquad (11)$$

$$\text{where } \gamma = (c_{j_1} , c_{j_2} , \dots , c_{j_M})$$

Again, using the inverse B^{-1} of the basis B, the multipliers can be calculated

$$\Pi = \gamma . \quad B^{-1} \qquad (12)$$

If the simplex multipliers Π are known at each step k of the algorithm, then the coefficients c'_j of the objective function can be calculated using the rule (9). If one of these coefficients c'_j is found to be negative (the corresponding variabl x_j must therefore be non-basic) then a better basic feasible solution can be obtained by introducing the variable x_j into the set of basic variables; i.e. by introducing the vector P_j into the basis. The basic variable x_{j_i} and its corresponding vector P_{j_i} which must be removed from the basis is determined from the elements of the constants vector Q' and the vector P'_j (which is calculated from P_j using the rule (6-a) according to the known rule :

Corresponding to each positive element $a'_{ij} > 0$ of the vector P'_j calculate the characteristic quotient :

$$q_i = b'_i / a'_{ij} \quad , \quad a'_{ij} > 0 \qquad (13)$$

where

$$P'_j \begin{bmatrix} a'_{1j} \\ a'_{2j} \\ \vdots \\ a'_{ij} \\ \vdots \\ a'_{Mj} \end{bmatrix} \quad , \quad Q' = \begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_i \\ \vdots \\ b'_M \end{bmatrix}$$

The basic variable x_{j_i} corresponding to the smallest quotient q_i will be removed from the set of basic variables; its corresponding vector P_{j_i} will be replaced by the vector P_j in order to get the new basis B' for step k+1 of the algorithm. The new inverse B^{-1} is obtained by modifying B^{-1} according to this replacement. The new multipliers π of step k+1 can be then calculated using rule (12) :

$$\pi' = \gamma' . B'^{-1}$$

where $\gamma' = (C_{j_1}, C_{j_2}, \ldots, C_j, \ldots, C_{j_M})$ (14)

(C_j instead of C_{j_i})

It is more convenient practically, and theoretically (for some applications), to include the matrix B^{-1} and the simplex multipliers in one matrix.

$$\bar{B}^{-1} = \begin{array}{c|ccc} 1 & -\pi_1 & -\pi_2 \ \ldots & -\pi_M \\ \hline 0 & & & \\ \vdots & & B^{-1} & \\ 0 & & & \end{array} \qquad (15)$$

It can be sum that this is the inverse of the matrix

$$\bar{B} = \begin{array}{c|ccc} 1 & C_{j_1} & C_{j_2} \ \ldots\ldots & C_{j_M} \\ \hline 0 & & & \\ \vdots & & B & \\ 0 & & & \end{array} \qquad (16)$$

This can be proved by multiplying the two matrices

The system (2-a) , (2-b) can be written in the combined vector form

$$\bar{P}_o'.(z) + \bar{P}_1 x_1 + \bar{P}_2 x_2 + \dots + \bar{P}_j x_j + \dots + \bar{P}_N x_N = \bar{Q} \quad (17$$

where the M+1 dimensional vectors

$$\bar{P}_o = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad , \quad \bar{P}_j = \begin{bmatrix} c_j \\ \hline P_j \end{bmatrix} \quad j = 1,2, \dots , N, \quad \text{and } \bar{Q} = \begin{bmatrix} 0 \\ \hline Q \end{bmatrix}$$

The system (3-a) , (3-b) will then be

$$\bar{P}_o'. (-z) + \bar{P}_1' x_1 + \bar{P}_2' x_2 + \dots + \bar{P}_j' x_j + \dots + \bar{P}_N' x_N = \bar{Q}' \quad (18)$$

Equations (18) must be in canonical form with respect **to** the variables $(-z)$, x_{j_1} , x_{j_2} , ... , x_{j_i} , ... , x_{j_M} , i.e. the vectors

$$\bar{P}_o' = U_o = \begin{bmatrix} 1 \\ \hline 0 \\ \vdots \\ 0 \end{bmatrix} ,$$

and

$$\bar{P}_{j_i}' = U_i = \begin{bmatrix} 0 \\ \hline 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \underline{\text{Component no.}} \\ 0 \\ 1 \\ \vdots \\ M \end{matrix} \quad , i = 1,2,\dots,M$$

System (18) can thus be obtained by multiplying system (17) by the inverse $\bar{B}^{-1}$ of the matrix :

$$\bar{B} = (\bar{P}_o , \bar{P}_{j_1} , \bar{P}_{j_2} , \dots , \bar{P}_{j_i} , \dots , \bar{P}_M) \quad (19)$$

From which we have the following rule for calculating any columns vector in the system (18) of step k of the algorithm from the original system (17) using the inverse $\overline{B}^{-1}$

$$\overline{P}'_j = \overline{B}^{-1} \overline{P}_j \qquad (20\text{-a})$$

and

$$\overline{Q}' = \overline{B}^{-1} Q \qquad (20\text{-b})$$

The matrix $\overline{B}^{-1}$ must be recorded at each step of the algorithm, and modified from step to step according to the change in the basic variables and their vectors. It is also convenient to record the vector $\overline{Q}'$ and to modify it at the same time with the matrix $\overline{B}^{-1}$ using a pivoting operation at each step. The rules for modifying the matrix $\overline{B}^{-1}$ and the vector $\overline{Q}^{-1}$ are given in the following section.

3. Transformation from step k to step k+1

Besides the original system (2-a), (2-b) or (17) the following informations are recorded at each step k:

1) The basic variables

$$x_{j_1} , x_{j_2} , \ldots , x_{j_i} , \ldots , x_{j_M}$$

2) The matrix $\overline{B}^{-1}$ containing :

a) the negatives of simplex multiplieres

$$\pi_1, \pi_2 , \ldots , \pi_M$$

and

b) The inverse B^{-1} of the basis

3) The constants vector $\overline{Q}'$

These are represented in the following table :

Basic variables

(-z)	$-z_o$	$-\pi_1$ $-\pi_2$ .... $-\pi_i$ ... $-\pi_M$
x_{j_1}	b'_1	
x_{j_2}	b'_2	
⋮	⋮	B^{-1}
x_{j_i}	b_i	
⋮	⋮	
x_{j_M}	b_M	

Table (1-a)

The table of multipliers at step k

Each coefficient C'_j in the objective function is then calculated using the multipliers according to rule (9). If some are found to be negative, then the column P'_j corresponding to the most negative coefficient C'_j is calculated according to (20-a) , this column will be written in an extra column in the table (1-b). The pivot element a'_{ij} is determined according to the rule (13). The pivoting operation is then performed on the table (1-b), resulting in the table (1-c) where the old basic variable x_{j_i} in the pivot row i is replaced by the new basic variable x_j. If the objective function has no negative coefficient C'_j, then the basic feasible solution, obtained from the table according to rule (7), is an optimal solution

Basic variables

Extra Column

$(-z)$	$-z_o$	$-\pi_1$ $-\pi_2$ ... $-\pi_i$ ... $-\pi_M$	c'_j
x_{j_1}	b'_1		a'_{1j}
x_{j_2}	b'_2		a'_{2j}
$\vdots$	$\vdots$	B^{-1}	$\vdots$
x_{j_i}	b'_i		a'_{ij} pivot element
$\vdots$	$\vdots$		$\vdots$
x_{j_M}	b'_M		a'_{Mj}

Table (1-b)

Basic Variables

$(-z)$	$-z''_o$	$-\pi'_1$ $-\pi'_2$ ... $-\pi'_i$ ... $-\pi'_M$
x_{j_1}	b''_1	
x_{j_2}	b''_2	
$\vdots$	$\vdots$	B'^{-1}
new basic variable → x_j	b''_i	
$\vdots$	$\vdots$	
x_{j_M}	b''_M	

Table (1-c)

The table of multipliers at step k+1

4. The two-phase algorithm using multipliers

In order to solve the linear program given by the system (1-a), (1-b), (1-c) using the two-phase method, the following procedure is used for starting the algorithm :

1) The equations (1-b) with $b_i < 0$ are multiplied by -1.

2) A number of M artificial variables

$$x_{N+1} \ , \ x_{N+2} \ , \ \cdots \ , \ x_{N+M}$$

are introduced.

3) The objective function for phase 1 is defined by

$$w = x_{N+1} + x_{N+2} + \cdots + x_{N+M}$$

The artificial variables are then eleminated from the w-equation by substracting the equations (1-b).

The result is the following table :

1	(-w)	(-z)	x_1	x_2		x_j		x_N	x_{N+1}		x_{N+M}
$-w_o$	1		d_1	d_2	...	d_j	...	d_N	0	...	0
0		1	c_1	c_2	...	c_j	...	c_N	0	...	0
b_1			a_{11}	a_{12}	...	a_{1j}	...	a_{1N}	1		
b_2			a_{21}	a_{22}	...	a_{2j}	...	a_{2N}		1	
⋮			⋮	⋮		⋮		⋮			
b_M			a_{M1}	a_{M2}	...	a_{Mj}	...	a_{MN}			1

Table 2.

where

$$d_j = -\sum_{i=1}^{M} a_{ij} \quad \text{for } j=1,2, \ldots, N$$

$$\text{and} \quad -w_o = \sum_{i=1}^{M} b_i \qquad (21)$$

The columns of the artificial variables, and the columns of the variables (-w), (-z) need not be stored as these will not be used by the algorithm. The original table 2 will remain unchanged during the algorithm; which is one of the advantages of the method of multipliers, because the original data of the problem will not be a subject to errors due to rounding off in the computer.

The artificial variables are taken as the basic variables at the start, we have thus the following table of multipliers :

Basic variables				
(-w)	$-w_o$	0	....	0
(-z)	0	0	....	0
x_{N+1}	b_1	1		
x_{N+2}	b_2		1	
⋮	⋮			
x_{N+M}	b_M			1

Table (3-a)

First table of multipliers

The table will be transformed from step to step according to the rules given in section 3, where the first row (the w-row) is taken as the row of the objective function during phase 1, and the second row (the z-row) during phase 2.

At any step k we have the following table

Basic variables							
$(-w)$	(w_o)	$-\sigma_1$	$-\sigma_2$	...	$-\sigma_i$	...	$-\sigma_M$
$(-z)$	$(-z_o)$	$-\pi_1$	$-\pi_2$	...	$-\pi_i$	...	$-\pi_M$
x_{j_1}	b_1						
x_{j_2}	b_2			B^{-1}			
$\vdots$	$\vdots$						
x_{j_i}	b_i						
$\vdots$	$\vdots$						
x_{j_M}	b_M						

Table (3-b)

The table of multipliers at step k

we have two sets of multipliers

$$\sigma_1 , \sigma_2 , \cdots , \sigma_i , \cdots , \sigma_M$$

to be used in phase 1 with the first objective function and

$$\pi_1 , \pi_2 , \cdots , \pi_i , \cdots , \pi_M .$$

to be used in phase 2 with the second objective function

The multipliers π_i are calculated during phase 1, so that they will be ready for use immediately at the start of

phase 2. The first row of both tables 2 and 3-b can be ommited in phase 2.

The algorithm is illustrated by the example given in the following section. A complete flow chart and the FORTRAN program are given at the end of this paper.

5. An Example:

Consider the following linear programming problem:

Minimize $z = x_1 + 6x_2 - 7x_3 + x_4 + 5x_5$

subject to the equations:

$$x_1 - x_2 + 5x_3 - x_4 + x_5 + x_6 = 8$$

$$5x_1 - 4x_2 + 13x_3 - 2x_4 + x_5 + x_7 = 20$$

The objective function w for phase 1 is introduced and the resulting extended problem is written (in the form of table 2 of section 4) in the following table:

1	(-w)	(-z)	x_1	x_2	x_3	x_4	x_5
-28	1		-6	5	-18	3	-2
0		1	1	6	-7	1	5
8			1	-1	5	-1	1
20			5	-4	13	-2	1

Table 4

The columns of the artificial variables x_6, x_7 are not entered since they will not be needed for further computations.

a) Phase 1

The first table of multipliers (Table 3-9 section) is then prepared with the artificial variables x_6, x_7 as basic variables

-w	-28	0	0
-z	0	0	0
x_6	8	1	0
x_7	20	0	1

Table 5,a1

The new basic variable x_3 is chosen according to the rule (9) o section 3; the corresponding extra column is computed according the rule (6-a) and then added to the table

				x_3
-w	-28	0	0	-18
-z	0	0	0	- 7
x_6	8	1	0	5
x_7	20	0	1	13

Table 5,a1'

After choosing the pivot and performing the pivoting operation the extra column is dropped and we have the following table wit the new basic variable x_3

-w	-0.30769	0	1.38461
-z	10.7692	0	0.53846
x_6	0.30769	1	-0.38462
x_3	1.53846	0	0.07692

Table 5,a2

Again a new basic variable x_5 is chosen and its corresponding extra column is computed and added to the table

			x_5	
-w	- 0.30769	0	1.38461	-0.61538
-z	10.7692	0	0.53846	5.53846
x_6	0.30769	1	-0.38462	0.61538
x_3	1.53846	0	0.07692	0.07692

Table 5,a2

After performing the pivoting operation and dropping the extra column we get the following table with the new basic variable x_5

-w	0	1	1
-z	8	- 9	4
x_5	0.5	1.625	-0.625
x_3	1.5	-0.125	0.125

Table 5,a3

Phase 1 is thus terminated because the artificial variables x_6, x_7 are not more in the basic set of variables; note also that the value of the objective function w of phase 1 in now equal to zero.

b) Phase 2

The row -w of the multipliers corresponding to the objective function of phase 1 is cancelled from the table of multipliers; also the first row of the original table 4 is to be forgotten during the phase 2 operations.

Thus, we start phase 2 with the following table of multipliers:

-z	8	-9	4
x_5	0.5	1.625	-0.625
x_3	1.5	-0.125	-0.125

Table 5 b,1

The new basic variable x_2 is chosen and its extra column is introduced:

		x_2		
$-z$	8	-9	4	-1
x_5	0.5	1.625	-0.625	0.875
x_3	1.5	-0.125	-0.125	0.625

Table 5 b,1'

After the pivoting operation we have the table with the new basic variable x_2

$-z$	8.57142	-7.14285	3.28571
x_2	0.57142	1.85714	-0.71428
x_3	1.71428	0.57143	-0.14285

Table 5 b,2

Computing the coefficients c_j' of the objective function (in order to choose a new basic variable), we find that all these coefficients are non-negative, which means that we have reached the optimal solution which can be immediatly read from the last table 5b,2 of multipliers:

optimal value of the objective function:

$$z = 8.57142$$

The basic variables

$$x_2 = 0.57142$$
$$x_3 = 1.71428$$

The non-basic variables

$$x_1 = x_4 = x_5 = 0$$

6. The Fortran Program :

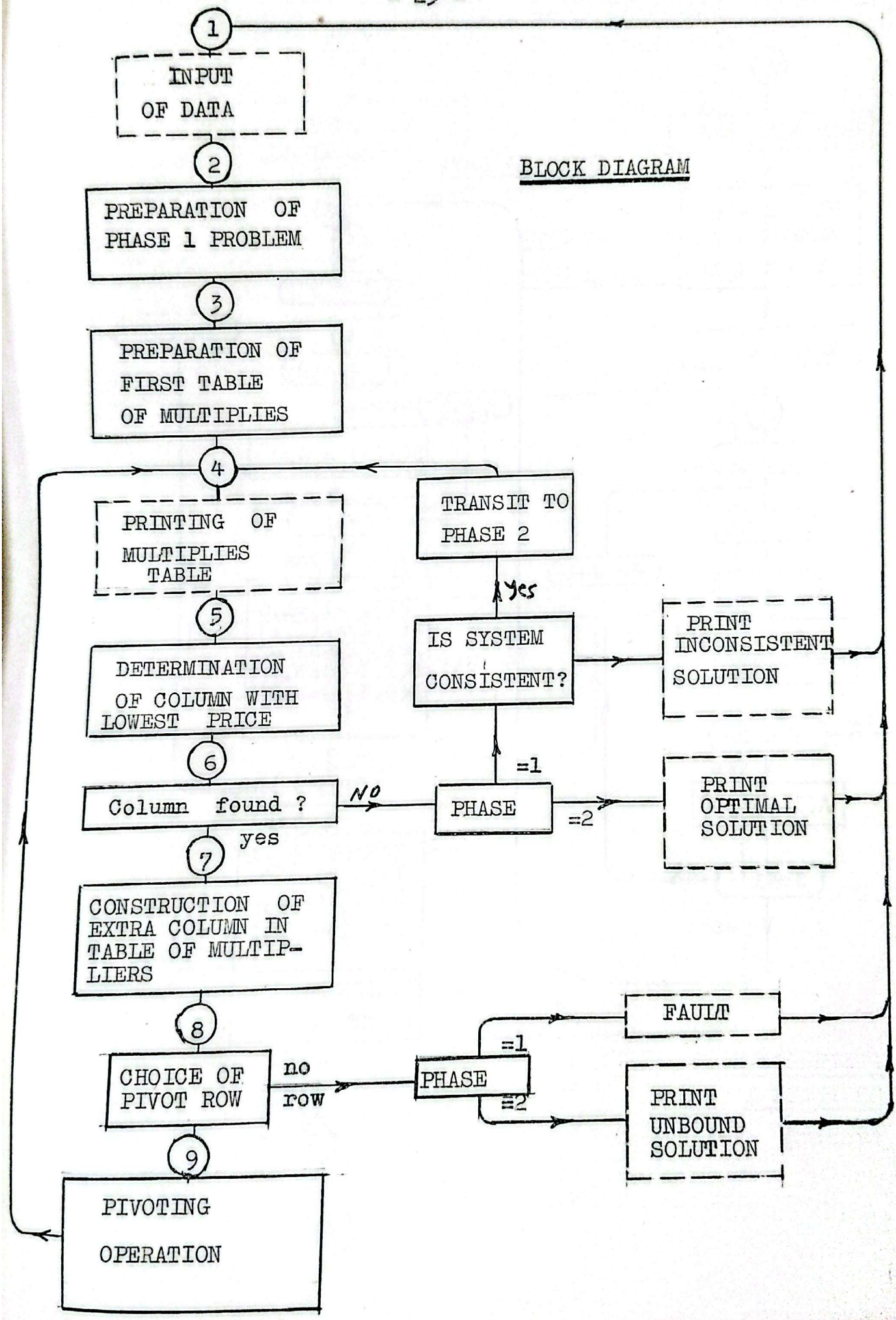
```
C     SIMPLEX  ALGORITHM
C     METHOD OF MULTIPLIERS

      DIMENSION A(20,40),B(20,20),X(60),IBAS(20)
C     INPUT OF DATA
    1 READ 2,NPROB,M,N,EPS
    2 FORMAT(I6,2I3,E8.1)
      PRINT 3,NPROB,M,N,EPS, (I,I=1,N)
    3 FORMAT(///////14HPROBLEM NUMBER,I6//3HM =,I3,5X,3HN =,I3
     1       , 10 X,6HEPS = ,E8.1///5H  1  ,5(9X, 1HX,I 2)/)
      M1=M+1
       M2=M+2
       N1=M+1
       DO 4 I=2,M2
      READ 5,(A(I,J),J=1,N1)
    4 PRINT 6,(A(I,J),J=1,N1)
    5 FORMAT(8F10.4)
    6 FORMAT(E10.3,2X))
      DO 8 J=1,N1
      C=0.0
       DO 7 I=3,M2
    7 C=C+A(I,J)
    8 A(1,J)=-C

C      FIRST TABLE
      DO 10 I=3,M2
      DO  9 J=2,M1
    9 B(I,J)=0.0
      B(1,I-1)=1.0
      B(I,1)=A(I,1)
      IBAS(I)=N+I-2
      B(1,I-1)=0.0
   10 B(2,I-1)=0.0
      IBAS(1)=-1
      IBAS(2)=0
      B(1,1)=A(1,1)
      B(2,1)=A(2,1)
```

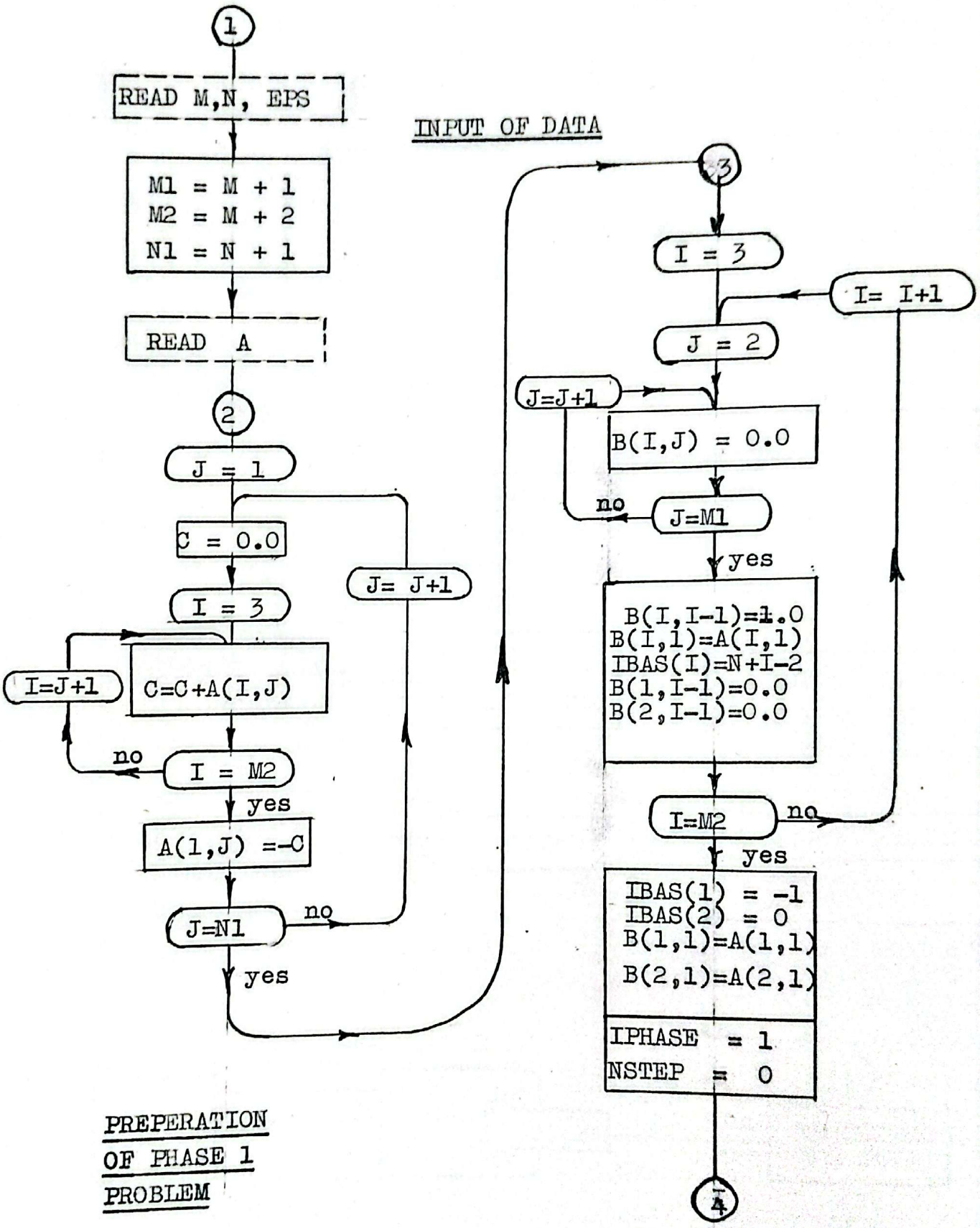
```
C          PRINTING OF TABLE
   11      IPHASE=1
           NSTEP =0
   12      PRINT 13, IPHASE,NSTEP
   13      FORMAT(////5HPHASE,I2, 10X, 5HSTEP,I2//)
           IF(SENSE SWITCH 1) 14, 16
   14      PUNCH 13,IPHASE,NSTEP
           DO 15 I=IPHASE,M2
   15      PUNCH 19,IBAS(I),(B(I,J),J=1,M1)
   16      IF(SENSE SWITCH 2) 17,20
   17      DO 18 I=IPHASE,M2
   18      PRINT 19,IBAS(I),(B(I,J),J=1,M1)
   19      FORMAT(1HX,I2,5(3X,E12.5))

C          LOWEST PRICE
   20      CD=0.0
           DO 23 J=2,N1
           CDJ=A(IPHASE,J)
           DO 21 I=3,M2
   21      CDJ=CDJ+B(IPHASE,I-1)*A(I,J)
           IF(CDJ-CD)22,23,23
   22      JP=J
           CD=CDJ
   23      CONTINUE
           IBV=JP-1

C
C          TRANSIT PHASE 1 TO 2
           IF(CD+EPS) 28,24,24
   24      GO TO(25,45), IPHASE
   25      IF(B(1,1)-EPS)26,26,41
   26      IPHASE=2
           NSTEP=0
           GO TO 12
C
C          EXTRA COLUMN
   28      GO TO (128,328), IPHASE
  128      DC=A(2,JP)
           DO 228 I=3,M2
  228      DC=DC+B(2,I-1)*A(I,JP)
           B(2,M2)=DC
  328      B(IPHASE,M2)=CD
           DO 30 I=3,M2
           C=0.0
           DO 29 J=3,M2
   29      C=C+B(I,J-1)*A(J,JP)
           B(I,M2)=C
   30        CONTINUE
```

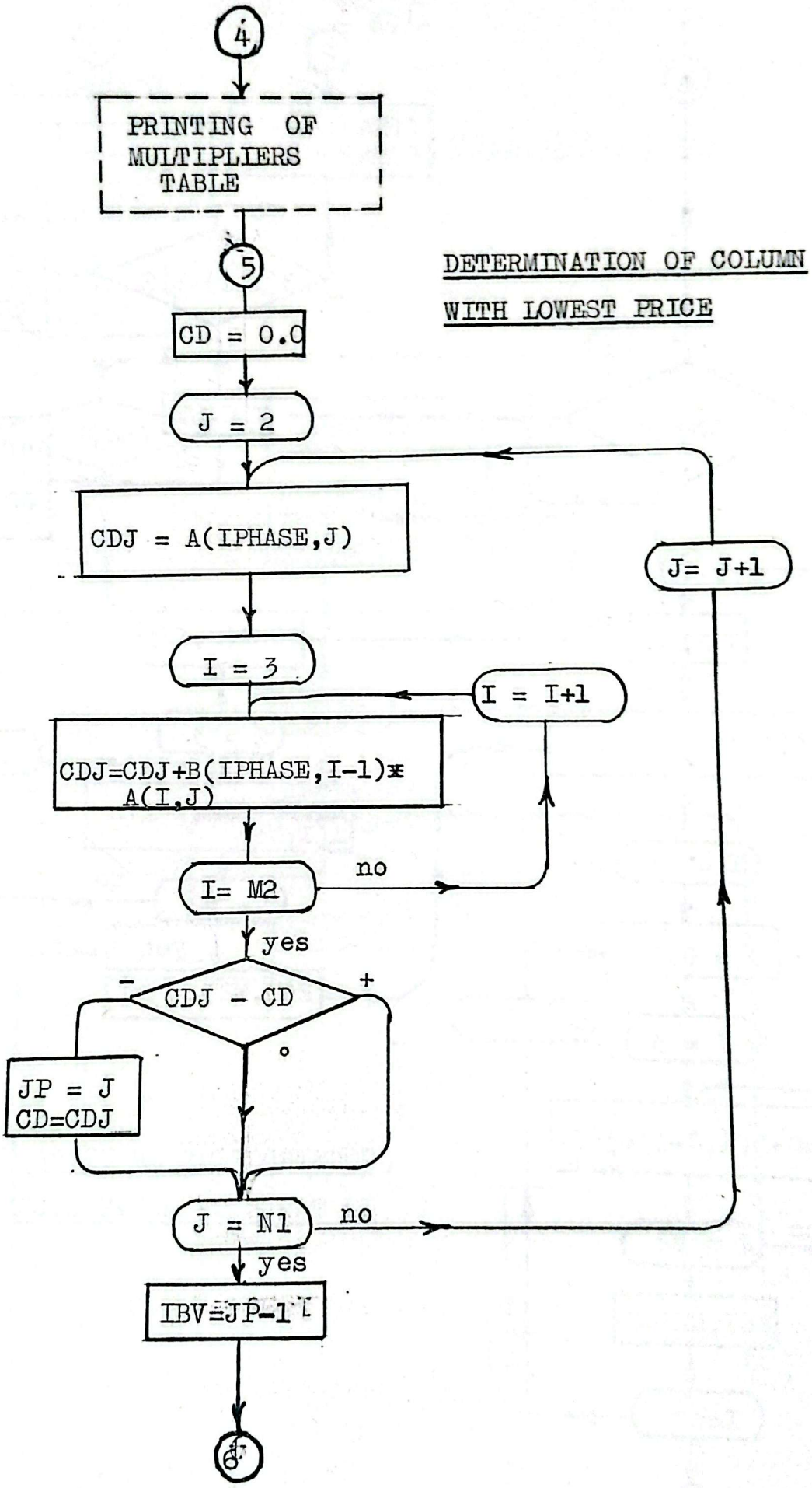
```
C       PIVOT ROW
        IP=-1
         Q=1.0E+90
        DO 33 I=3,M2
        IF(B(I,M2))33,33,31
   31   QI=B(I,1)/B(I,M2)
        IF(QI-Q)32,33,33
   32   Q=QI
         IP=I
   33    CONTINUE
         IF(IP)34,34,35
   34   GO TO(52,37),IPHASE

C       PIVOTING OPERATION
   35   PIVINV=1./B(IP,M2)
        B(IP,M2)=0.0
        DO 36 J=1,M1
        C=B(IP,J)*PIVINV
        B(IP,J)=C
        DO 36 I=IPHASE,M2
   36   B(I,J)=B(I,J)-C*B(I,M2)
        IBAS(IP)=IBV
        NSTEP=NSTEP+1
        GO TO 12

C       RESULTS

C       UNBOUND SOLUTION
   37   PRINT 38,IBV
   38   FORMAT(//16HUNBOUND SOLUTION/1HX,I2,14H = T = INFINITY//)
        DO 39 I=2,M2
   39   PRINT 40,IBAS(I),B(I,1),B(I,M2)
   40   FORMAT(1HX,I2,3H = ,E12.5,   2X ,E12.5,4H * T)
        GO TO 55

C       INCONSISTENT
   41   PRINT 42,B(1,1),B(2,1)
   42   FORMAT(//22HEQUATIONS INCONSISTENT///6HW    = ,E12.5/
       1       6HZ    = ,E12.5/)
        NVAR=M+N
         GO TO 47

C       OPTIMAL SOLUTION
   45   PRINT 46,B(2,1)
   46   FORMAT(//16HOPTIMAL  SOLUTION///6HZ    = ,E12.5/)
        NVAR=N
```

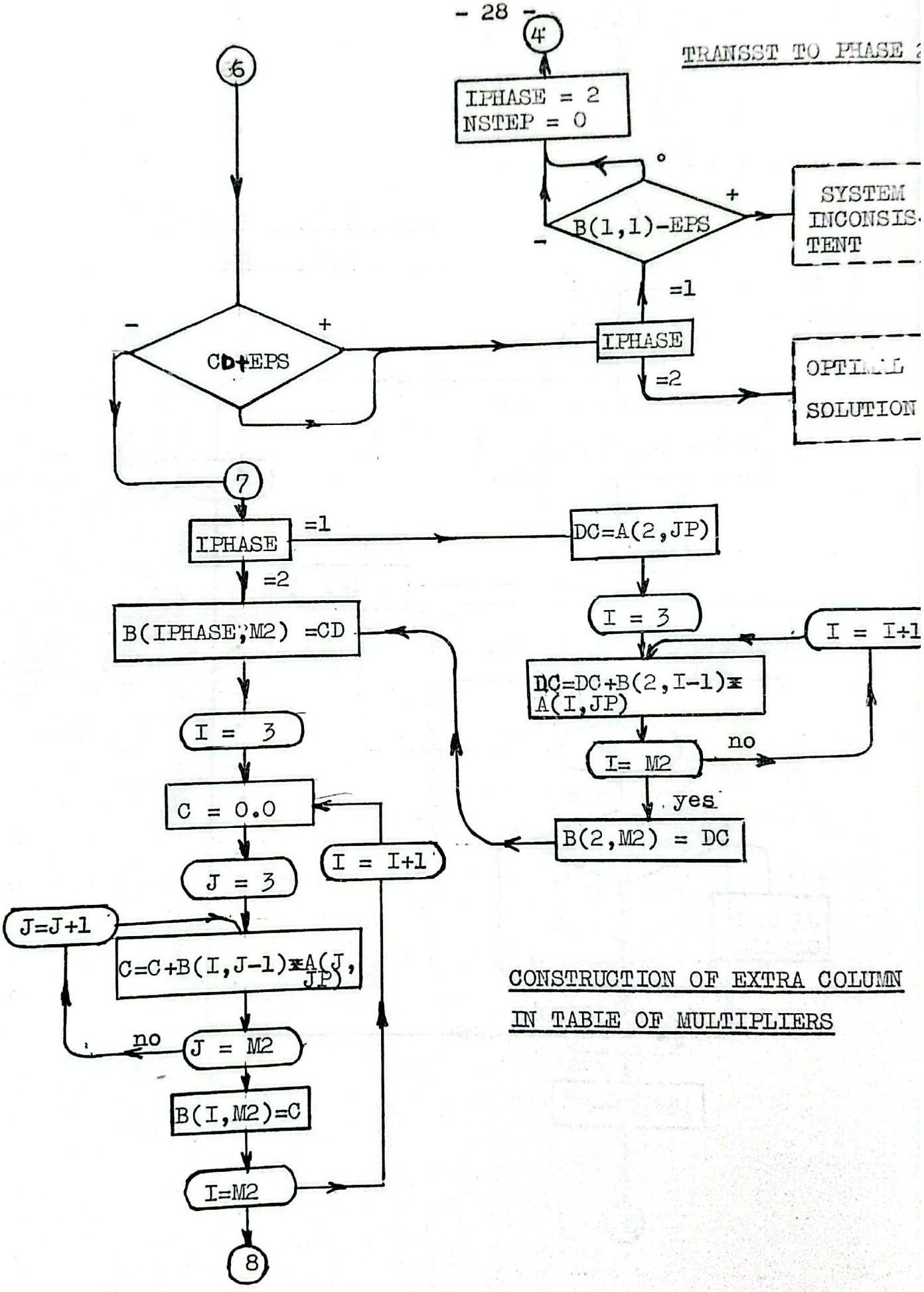
```
C       DETERMINATION OF X
   47   DO 48 J=1,NVAR
   48   X(J)=0.0
        DO 49 I=3,M2
        I X=IBAS(I)
   49   X(IX)=B(I,1)
        DO 50 I=1,NVAR
   50   PRINT 51,I,X(I)
   51   FORMAT(1HX,I2,3H = ,E12.5)
        GO TO 55

C       FAULTS
   52   PRINT 54,IPHASE
   54   FORMAT(5HFAULT,I3)
   55   GO TO 1
        END
```

BLOCK DIAGRAM

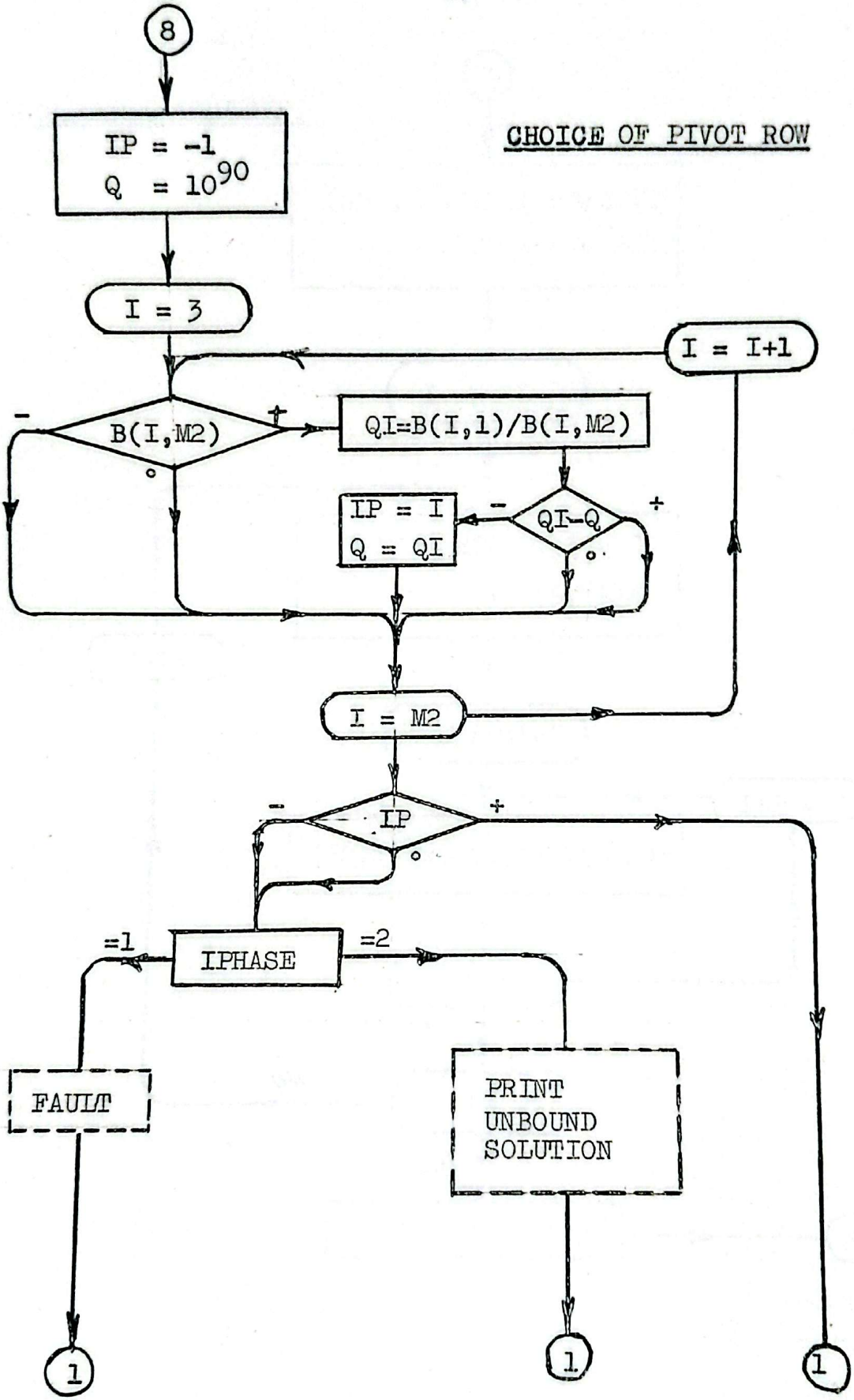



1
READ M,N, EPS
M1 = M + 1
M2 = M + 2
N1 = N + 1
READ A
2
J = 1
C = 0.0
I = 3
C=C+A(I,J)
I=J+1
no
I = M2
yes
A(1,J) =-C
J=N1
no
yes
J= J+1
INPUT OF DATA
3
I = 3
I= I+1
J = 2
J=J+1
B(I,J) = 0.0
no
J=M1
yes
B(I,I-1)=1.0
B(I,1)=A(I,1)
IBAS(I)=N+I-2
B(1,I-1)=0.0
B(2,I-1)=0.0
I=M2
no
yes
IBAS(1) = -1
IBAS(2) = 0
B(1,1)=A(1,1)
B(2,1)=A(2,1)
IPHASE = 1
NSTEP = 0
4
PREPERATION
OF PHASE 1
PROBLEM


4
PRINTING OF MULTIPLIERS TABLE
5
DETERMINATION OF COLUMN WITH LOWEST PRICE
CD = 0.0
J = 2
CDJ = A(IPHASE,J)
J= J+1
I = 3
I = I+1
CDJ=CDJ+B(IPHASE,I-1)*A(I,J)
I= M2
no
yes
CDJ - CD
-
+
o
JP = J
CD=CDJ
J = N1
no
yes
IBV=JP-1
6



CONSTRUCTION OF EXTRA COLUMN IN TABLE OF MULTIPLIERS



CHOICE OF PIVOT ROW

PIVOTING OPERATION