MATHEMATICSCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM
REKENAFDELING

A PUNCHED-CARD SET-UP FOR LINEAR PROGRAMMING

by

M.L. Potters

MR 22

1956

A PUNCHED-CARD SET-UP FOR LINEAR PROGRAMMING

MR 22

1. Introduction

In this paper a procedure is described for „linear programming" with the help of punched-card equipment. The Simplex Method we use is supposed to be well-known; for the sake of completeness a short outline of this method is given. For theoretical considerations and special cases we refer to the existing literature on this subject a.o[1), 2), 3)]. The procedure has been designed for an equipment of the following composition:

IBM Key Punch	type 016
IBM Sorter	type 080
IBM Reproducer	type 513

provided with at least one class-selector

IBM Tabulator	type 405
IBM Calculating Punch	type 602 A

provided with deviding device, all 8 storages, all 8 counters, all 17 pilot-selectors, all 12 coselectors

Probably it can also be used for type 604 and other machines of this kind.

2. The problem

Although a more general treatment is possible, we put the problem in the following form:

The linear function Z_0 of the p variables λ_j

$$Z_0 = \sum_{j=1}^{p} c_j \lambda_j \qquad (2.1)$$

is to be made extremal (e g maximal) under the conditions

$$\lambda_j \geqslant 0 \qquad (j = 1 \ldots p) \qquad (2.2)$$

$$\sum_{j=1}^{p} a_{ij}\lambda_j \leqslant 0 \qquad (i = 1 \ldots m) \qquad (2.3)$$

The constants b_i in the right-hand member of (2.3) are required to be non-negative

$$b_i \geqslant 0 \qquad (i = 1 \ldots m) \qquad (2.4)$$

For the c_j and a_{ij} there are no restrictions.

The m inequalities (2.3) are replaced by equalities if one defines m extra variables by

$$\lambda_{p+i} = b_i - \sum_{j=1}^{p} a_{ij}\lambda_j \qquad (i = 1 \ldots m) \qquad (2.5)$$

From (2.3) we have the conditions

$$\lambda_{p+i} \geqslant 0 \qquad (i = 1 \ldots m) \qquad (2.6)$$

Furthermore we define

$$c_{p+i} = 0 \qquad (i = 1 \ldots n) \qquad (2.7)$$

The problem now has assumed the following form:
Maximize the linear function Z_0 of n variables λ_l

$$Z_0 = \sum_{l=1}^{n} c_l \lambda_l \qquad (l=1\ldots\ldots n) \tag{2.8}$$

where n = p + m, under the conditions

$$\lambda_l \geqslant 0 \qquad (l=1\ldots\ldots n) \tag{2.9}$$

$$\sum_{j=1}^{p} a_{ij}\lambda_j + \lambda_{p+i} = b_i \qquad (i=1\ldots\ldots m) \tag{2.10}$$

3. Outline of the Simplex Method

We consider systems of m linear equations with n unknowns of the following type

$$\lambda_{s_i} + \sum_{j=1}^{p} a_{ij}\lambda_{t_j} = b_i \qquad (i=1\ldots\ldots m) \tag{3.1}$$

where

$$b_i \geqslant 0 \qquad (i=1\ldots\ldots m) \tag{3.2}$$

and $n = p + m$.

Only those solutions are admitted, for which holds

$$\lambda_l \geqslant 0 \qquad (l=1\ldots\ldots n) \tag{3.3}$$

The n variables appearing in (3.1) can be divided into two classes:

(a) the m variables λ_{s_i} (i = 1....m) called „basic" variables
(b) the p variables λ_{t_j} (j = 1....p) called „non-basic" variables.

In each of the m equations (3.1) only one of the basic variables appears; each of those variables appears in only one equation.
We define the "basic solution" of (3.1) as follows:

$$\begin{aligned} \lambda_{s_i} &= b_i \qquad (i=1\ldots\ldots m) \\ \lambda_{t_j} &= 0 \qquad (j=1\ldots\ldots p) \end{aligned} \tag{3.4}$$

This solution indeed satisfies (3.3).
The value of Z_0 connected with the basic solution is

$$Z_0 = \sum_{i=1}^{m} c_{s_i} b_i \tag{3.5}$$

The equations (2.10) are a system such as (3.1) Here we have $s_i = p + i$, $t_j = j$; The basic solution $\lambda_{p+i} = b_i$, $\lambda_j = 0$ is the so-called „trivial solution" because $Z_0 = 0$.
Now we consider transformations of (3,1) with the following properties:

(a) the transformed system is of the same type as (3.1)

$$\lambda_{s_i'} + \sum_{j=1}^{p} a'_{ij}\lambda_{t_j'} = b'_i \qquad (i=1\ldots\ldots m) \tag{3.6}$$

where

$$b_i' \geqslant 0 \qquad (i=1\ldots\ldots m) \tag{3.7}$$

(b) one basic variable, say λ_{s_r} , has turned into a non-basic variable; one non-basic variable, say λ_{t_k} , has turned into a basic variable. The value of Z_0, connected with the basic solution of (3.6) is

$$Z_0' = \sum_{i=1}^{m} c_{s_i'} b_i' \tag{3.8}$$

It can be proved that the maximum of Z_0 is reached for the basic solution of a system which has derived from (3,1) by a finite number of transformations.

In the Simplex-Method the maximum of Z_0 is found by succesive transformations, starting as a rule from the trivial solution. The choice of the variables λ_{s_r} and λ_{t_k} which will change their roles as basic and non-basic variables is determined by the conditions:

$$Z_0' \geqslant Z_0 \tag{3.9}$$

$$b_i' \geqslant 0 \quad (i = 1 \ldots m) \tag{3.10}$$

This can be worked out in an simple calculating scheme.

4. The calculating scheme.

According to section 3 the following scheme is set up.
The state of the problem is fully characterized by a matrix (x_{ij}) of $(m + 2)$ rows and $(p + 1)$ columns as shown in fig. 1.
The rows are numbered $i = 1 \ldots m, Z, S$.
The columns are numbered $j = 1 \quad p, B$. This matrix contains as elements

(a) For $i = 1 \ldots m$; $j = 1 \ldots p$:

$x_{ij} = a_{ij}$, the coefficients of the non-basic variables in the equations (3.1)

To each row i is joined the number s_i of the basic variable, called „row variable number".

To each column j is joined the number t_j of the non-basic variable, called „column variable number".

(b) For $i = 1 \ldots m$; $j = B$:

$x_{ij} = b_i$, the constants of the righthand member of (3.1)

(c) For $i = Z$; $j = 1 \ldots p$:

$$x_{ij} = Z_j \text{ defined by } Z_j = \sum_{i=1}^{m} c_{s_i} a_{ij} - c_{tj} \tag{4.1}$$

For the system (2.10) we have $Z_j = - c_j$

(d) For $i = Z$; $j = B$:

$x_{ij} = Z_0$ defined by (3.5). For (2.10) $Z_0 = 0$.

(e) $i = S$; $j = 1 \ldots p$:

$$x_{ij} = S_j \text{ defined by } S_j = -\sum_{i=1}^{m} a_{ij} - Z_j \tag{4.2}$$

(f) For $i = S$; $j = B$:

$$x_{ij} = S_0 \text{ defined by } S_0 = - \sum_{i=1}^{m} b_i - Z_0 \tag{4.3}$$

The elements in the row S are defined in such a way that

$$\sum_{i=1}^{m,Z,S} x_{ij} = 0 \qquad (j = 1\ldots p,B) \qquad (4.4)$$

This row is added to the matrix for checking purposes.
Now the row number r and the column number k have to be fixed. The column k is the column in which the most negative value of Z_j appears. When there is no negative Z_j, the optimal solution is reached.
The row r is the row in which the smallest positive quotient $\frac{b_i}{a_{ik}}$ is found.
In cases where these conditions do not lead uniquely to a decision ("degeneracy") extra conditions can be found in the literature. The transformation of (3.1) is determined by the indication of r and k. It leads to a new matrix (x_{ij}'). In fig. 2 the elements of (x_{ij}') are shown expressed in those of (x_{ij}); they are calculated by six different operations:

(a) For $i = 1\ldots m,Z$, but $i \neq r$; $j = 1\ldots p,B$, but $j \neq k$:

$$x_{ij}' = x_{ij} - \frac{x_{rj}x_{ik}}{x_{rk}}$$

(b) For $i = r$; $j = 1\ldots p,B$ but $j \neq k$:

$$x_{ij}' = \frac{x_{ij}}{x_{ik}}$$

(c) For $i = 1\ldots m,Z$ but $i \neq r$; $j = k$:

$$x_{ij}' = \frac{x_{ij}}{x_{rj}}$$

(d) For $i = r$; $j = k$:

$$x_{ij}' = \frac{1}{x_{ij}}$$

(e) For $i = S$; $j = 1\ldots p,B$ but $j \neq k$:

$$x_{ij}' = x_{ij} - \frac{x_{rj}}{x_{rk}}(1 + x_{ik})$$

(f) For $i = S$; $j = k$:

$$x_{ij}' = -1 - \frac{1}{x_{rj}}(1 + x_{ij})$$

The new row variables are:

For $i \neq r$: $s_i' = s_i$

For $i = r$: $s_i'(=s_r') = t_k$

The new column variables are

For $j \neq k$: $t_j' = t_j$

For $j = k$: $t_j'(= t_k') = s_r$

		column	1	2		k		p	B
		column-variable	t_1	t_2		t_k		t_p	
row	row variable								
1	s_1		a_{11}	a_{12}		a_{1k}		a_{1p}	b_1
2	s_2		a_{21}	a_{22}		a_{2k}		a_{2p}	b_2
r	s_r		a_{r1}	a_{r2}		a_{rk}		a_{rp}	b_r
m	s_m		a_{m1}	a_{m2}		a_{mk}		a_{mp}	b_m
Z			Z_1	Z_2		Z_k		Z_p	Z_0
S			S_1	S_2		S_k		S_p	S_0

fig. 1

row	row-variable	column	1	2		k		p	B
		column variable	t_1	t_2		s_r		t_p	
1	s_1		$a_{11}-\frac{a_{r1}a_{1k}}{a_{rk}}$	$a_{12}-\frac{a_{r2}a_{1k}}{a_{rk}}$		$-\frac{a_{1k}}{a_{rk}}$		$a_{1p}-\frac{a_{rp}a_{1k}}{a_{rk}}$	$b_1-\frac{a_{1k}}{a_{rk}}b_r$
2	s_2		$a_{21}-\frac{a_{r1}a_{2k}}{a_{rk}}$	$a_{22}-\frac{a_{r2}a_{2k}}{a_{rk}}$		$-\frac{a_{2k}}{a_{rk}}$		$a_{2p}-\frac{a_{2p}a_{2k}}{a_{rk}}$	$b_2-\frac{a_{2k}}{a_{rk}}b_r$
r	t_k		$\frac{a_{r1}}{a_{rk}}$	$\frac{a_{r2}}{a_{rk}}$		$\frac{1}{a_{rk}}$		$\frac{a_{rp}}{a_{rk}}$	$\frac{b_r}{a_{rk}}$
m	s_m		$a_{m1}-\frac{a_{r1}a_{mk}}{a_{rk}}$	$a_{m2}-\frac{a_{r2}a_{mk}}{a_{rk}}$		$-\frac{a_{mk}}{a_{rk}}$		$a_{mp}-\frac{a_{rp}a_{mk}}{a_{rk}}$	$b_m-\frac{a_{mk}}{a_{rk}}b_r$
Z			$Z_1-\frac{a_{r1}}{a_{rk}}Z_k$	$Z_2-\frac{a_{r2}}{a_{rk}}Z_k$		$-\frac{Z_k}{a_{rk}}$		$Z_p-\frac{a_{rp}}{a_{rk}}Z_k$	$Z_0-\frac{b_r}{a_{rk}}Z_k$
S			$S_1-\frac{a_{r1}}{a_{rk}}(1+S_k)$	$S_2-\frac{a_{r2}}{a_{rk}}(1+S_k)$		$-1-\frac{1+S_k}{a_{rk}}$		$S_p-\frac{a_{rp}}{a_{rk}}(1+S_k)$	$S_0-\frac{b_r}{a_{rk}}(1+S_k)$

5. The punched-card lay out

For each step in the procedure we have a deck of (m+2)(p+1) cards, one for each row i and column j. On these cards the new matrix elements x_{ij}' are calculated and the new row and column-variable numbers are ounched.

To this end the cards contain the following fields:

field no.	information punched	symbol	operation no.
(1)	indicative data about the problem	ind.	B1
(2)	new step number	step	B1
(3)	11-punch, for i=S	X_S	B1
(4)	row-number	i	B1
(5)	11-punch, for i=r	X_r	B5
(6)	column number	j	B1
(7)	11-punch, for j=k	X_k	B2
(8)	new row variable number	s_i'	C
(9)	new column variable number	t_j'	C
(10)	new matrix element	x_{ij}'	C
(11)	old matrix element	x_{ij}	B1
(12)	old matrix element in the same row, and column k, for $j \neq k$	x_{ik}	B4
(13)	old row variable nr.	s_i	B1
(14)	old column variable number	t_j	B1
(15)	old column variable number in column k, for $j \neq k$	t_k	B4

Remarks:

(a) to fields (4) and (6). The row numbers Z and S and the column number B are punched alphabetically in the last position of these fields.

In the last position but one is punched - for sorting purposes -

in field (4): 11-punch for i = Z
12-punch for i = S.

in field (6) 11-punch for j = B.

The cards are taken from the sorter in the order 0,1,2,3,4,5,6,7,8,9,11, 12.

(b) to fields (8) and (13) For i = Z or S these fields contain zeros.

(c) to fields (9) and (14) For j = B these fields contain zeros.

(d) to fields (12) and (15) These fields are left blank for $j = k$.

(e) to fields (10)(11)(12) Negative numbers are indicated by an 11-punch in the last position.

6. Flow of the operations

Suppose we have reached the stage that a deck of cards has completely been punched. The cards then come from the multiplier sorted with respect to column number.

Now the following set of operations is to be performed:

A1 In the tabulator some data are listed and tabulated according to the subjoined scheme:

symbol	ind.	step	i	j	x_{ij}	x_{ij}'	s_i'	X_r	t_j'	X_k
list first card of column	(1)	(2)	(4)	(6)	(11)	(10)	(8)	(5)	(9)	(7)
list each card	-	-	(4)	-	(11)	(10)	(8)	(5)	-	-
total	-	-	-	-	(11)	(10)	-	-	-	

Remarks:

(a) The numbers i and j are printed by alphabetic type-bars to get the row numbers Z and S and the columnnumber B.

(b) The data X_r and X_k are printed as an asterisk next to the changed row, resp. column variable numbers.

(c) Negative matrix elements x_{ij} and x_{ij}' are provided with a CR-mark.

(d) The fields (10) and (11) are tabulated; a total is given over each column (with columnnumber and column variable number as indications; this supplies a check for the new column variable number). The totals should be zero, apart from rounding-off errors.

A2 The row r and column k are indicated according to section 4. The number k is found by inspection of the printed matrix; whereas r requires a few simple calculations on a desk-machine.

B1 In behalf of the next step a new deck of cards is made. The following fields are reproduced (with comparing) or gang-punched:

old deck	new deck
(1)	(1)
-	(2) gang-punched
(3)	(3)
(4)	(4)
(6)	(6)
(10)	(10)
(8)	(13)
(9)	(14)

After this operation the old deck is not used any longer.

B2 The column k is seperated from the deck. In field (7) of it an X_k is gang-punched.

B3 The column k is added to the deck as leading column. Then the cards are sorted to row number.

B4 In an off-set gang-punch operation with X_k-cards as master-cards the following fields are punched:

master card	detail cards
(11)	(12)
(14)	(15)

B5 The row r is separated from the deck. In field (5) of it an X_r is gang-punched.

B6 The row r is added to the deck as leading row. Then the cards are sorted to column number.

C. The multiplier calculates x_{ij}' in field (10) according to the formulae in section 4. The six different cases mentioned in that section are indicated in the cards by:

NOX_r, NOX_s, NOX_k	NOX_r, NOX_s, X_k
NOX_r, X_s, NOX_k	NOX_r, X_s, X_k
X_r, NOX_s, NOX_k	X_r, NOX_s, X_k

The multiplier also performs the "administrative" operation of punching the fields (8) and (9). This is easier than doing it on the reproducer because of the larger selection possibilities. The deck must be fed into the multiplier in the given order, i.e. sorted to column number with X_r-cards as leading cards in each column.

Initial operations

In the first deck of cards the fields (3),(4),(6),(11),(13) and (14) are key-punched. The fields (1) and (2) can be gang-punched. As a rule the first deck contains the trivial solution, for which we have $s_i = p + i$, $t_j = j$; $Z_j = -c_j$; $Z_0 = 0$. After this punching the cards are sorted with respect to row number and column number. Then the normal operations are performed starting with A1, but B1 will be skipped the first time. In special cases one can start with a solution other then the trivial one. This requires an artifice proposed by Wijvekate and described in 3). Particularly if a solution is known (not necessary basic) which lies near to the optimal solution one can save much work by applying this process.

7. Scaling and precision

From a point of view of precision it is preferable to scale the problem in such way that all matrix elements are of the order of unity.

Firstly, when no extremely high numbers occur, the optimal number of decimals can be used in the calculations.

Secondly, the error inherent to the procedure will be smaller. Indeed it is known from experience that digits may be lost in the course of the calculations.

Of course it is difficult to predict whether this condition will be fulfilled during the whole procedure even when the original matrix satisfies it.

When it happens that after a number of steps the matrix becomes too bad one can scale the matrix once again (by introducing new variables). One can also start the problem anew and apply the process mentioned at the end of the preceding section.

For larger problems which require many steps, double-length calculations are recommended but then a multiplier with higher capacity than the 602 A is necessary e.g. the 604.

8 References and acknowledgements

1) G.B. Dantzig, Maximilization of a linear function of variables subject to linear inequalities, Chapter XXI in "Activity Analysis of production and allocations", John Wiley and Sons (1951)

2) A. Charnes, W.W. Cooper and A. Henderson, "An introduction to linear programming", John Wiley and Sons, (1953)

3) J. Kriens, Cursus "Operations Research", Hoofdstuk II, Lineaire programmeringstheorie, Report S 166 (C9), Mathematical Centre, Amsterdam. (1955)

We wish to express our thanks to Mr. J. Kriens for his ample explanation about the Simplex Method.
Further we are indebted to Miss Ammie Berends who performed a test-case on our IBM punched-card set and to Miss Julie Schalij for the preparation of the manuscript.

Appendix

A Panel for the IBM calculating punch type 602 A.

Of course there are several ways to realize the operation C of section 6 on the wiring panel. On the board described below the machine performs for each card the operation

$$X = A + \frac{B}{C} x D$$

with a special choice of the quantities A, B, C and D in each of the six possible cases. This operation requires only one division and a few transfers.

The numbers we are dealing with are supposed to be scaled in such a way that they have at most 3 digits before and 4 digits behind the decimal point: xxx,xxxx.

The description consists of

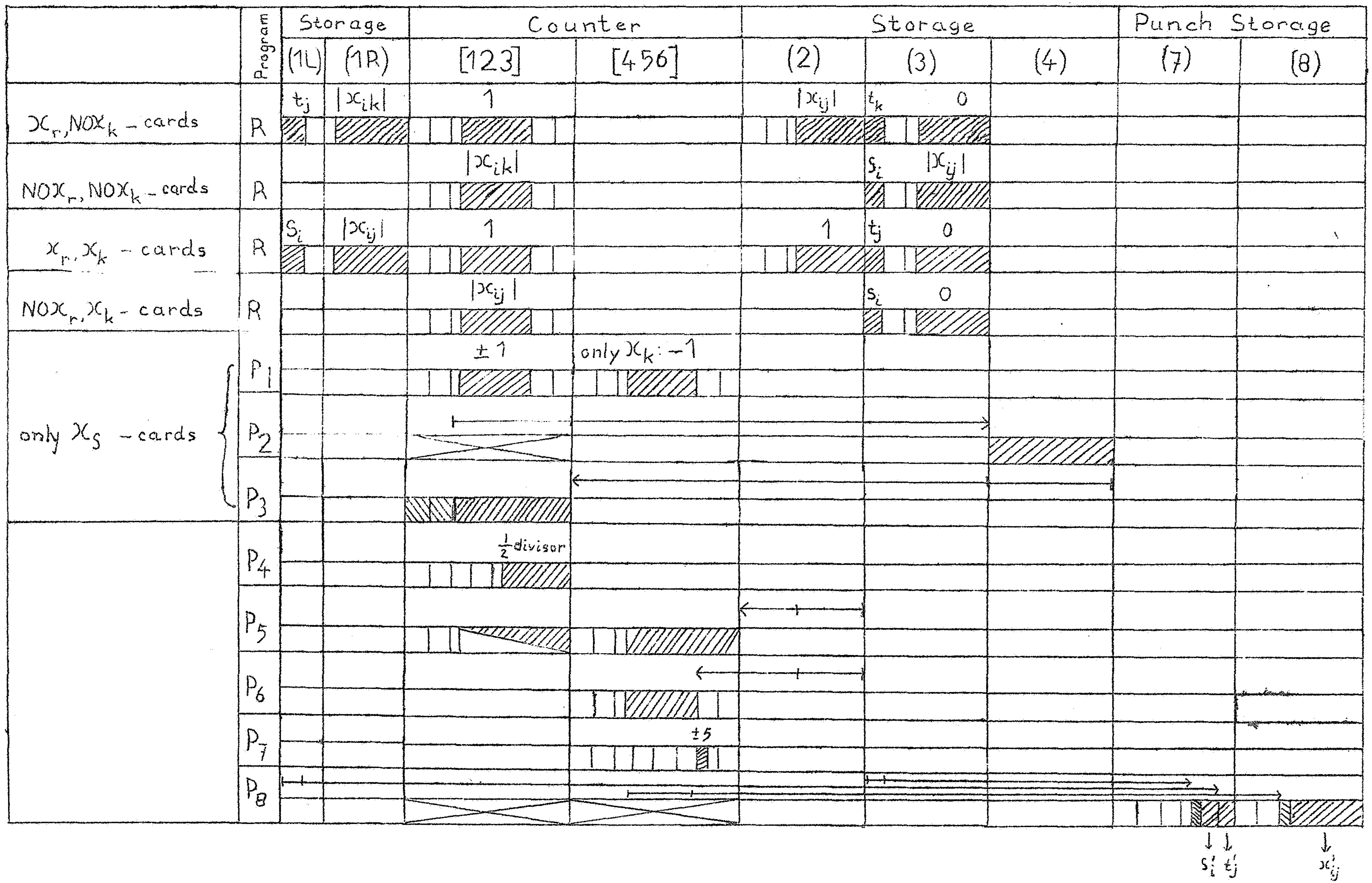
(a) Operations chart

(b) Table of selectors

(c) Scheme of channels

(d) Control impulses

a) OPERATIONS CHART

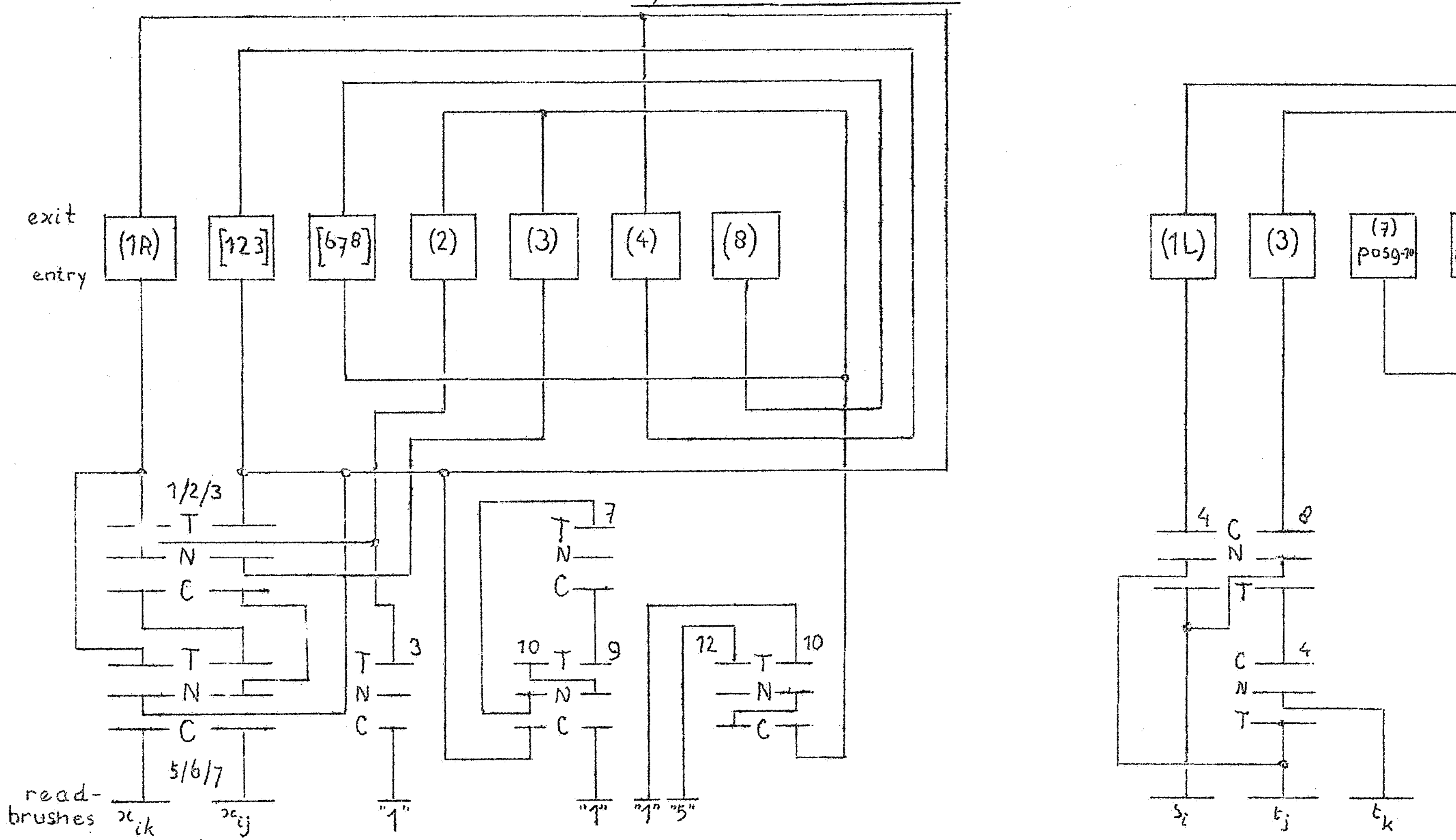


b) TABLE OF SELECTORS

Pilot-selector	Co-selector	Pick-up	Drop-out
1 2 3	5/6/7/8 1/2/3/4	CBr.X_r CBr.X_k CBr.X_s	RDO RDO RDO
5 6		RBr.X_r RBr.X_k	RDO RDO
10 11/12 13 14		only X_r:RBr.sign(IR) RBr." [123] only X_r:RBr." (2) RBr." (3)	only X_r:RDO RDO only X_r:RDO RDO
15/16 17		NB [123] NB [678]	RDO RDO
	9 10 11 12	Rcpl. P1cpl. P6cpl. P7cpl.	

CBr. = Control Brushes
RBr. = Read Brushes

c) SCHEME OF CHANNELS



d) CONTROL IMPULSES

Control-reading	PUPSel 1,2 and 3	
R-cpl	PUCoSel9	
R	RI + [123]	
	RI (3)	
	PUPSel 5,6, 11/12 and 14	
	RDOPSel 1,2,3,5,6,11/12, 14,15/16 and 17	
	RI(1) and (2)	only for X_r - cards
	PUPSel 10 and 13	
	RDOPSel 10 and 13	
	Skip to P4	only for NOX_s - cards
P1-cpl	PUCoSel 10	
P1	RI ± [123]	sign: PSel 11/12
	NB [123]→PUPSel 15/16	
	RI - [678]	only for X_k - cards
P2	RR [123]	
	RI (4)	
P3	RO (4)	
	RI ± [123]	sign: PSel 15/16
P4	RO (1R)	
	RI + [123]	
	X5, tenths	
P5	Div.	
	RO (1R)	
	RO (2)	
	RI - [123]	
	RI ∓ [678]	Sign: PSel 5, 10, 11/12 13, 15/16
P6-cpl	PUCoSel 11	
P6	RO (3)	
	RI ± [678]	sign: PSel 14
	NB [678]→PUPSel 17	
P7-cpl	PUCoSel12	
P7	RI ±[678]	sign: PSel 17
P8	RR [123]	
	RR [678]	
	RO(1L)	
	RO (3)	
	RI and pch (7) and (8)	
	Read.	

CONTENTS