1,039,141

PATENT SPECIFICATION

DRAWINGS ATTACHED

1,039,141

Date of filing Complete Specification: May 15, 1963.

Application Date: May 21, 1962. No. 19463/62.

Complete Specification Published: Aug. 17, 1966.

Index at acceptance:—G4 A (15D, 16D)

Int. Cl.:—G 06 f

COMPLETE SPECIFICATION

A Computer Arranged for the Automatic Solution of Linear Programming Problems

We, THE BRITISH PETROLEUM COMPANY LIMITED, of Britannic House, Finsbury Circus, London, E.C.2, a British joint-stock Corporation, and PATRICK VINCENT SLEE and PAULINE MARGARET JOYCE HARRIS, both of the aforesaid Company's address and both of British nationality, do hereby declare the invention for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

This invention relates to linear programming means for controlling the operation of data processing apparatus.

Briefly linear programming, so far as it relates to the present invention, is concerned with the control of digital data processing machines in computing solutions to that class of problems which involve simultaneous linear functions. These functions may be equations and/or inequalities descriptive of relationships characterizing a physical system, e.g. a refinery.

The functions in general are satisfied by an infinite number of solutions and the digital computer is typically controlled or programmed to obtain a particular solution which will satisfy the set of functions. In the general case this particular solution is an optimum one which yields specific value for the component variables of the functions. When these values are applied to the physical system represented by the set, the system yields an optimum response.

In linear programming, it has been customary heretofore, to employ the "simplex" method of controlling the digital computer in obtaining a solution to the set of linear functions. The simplex method, together with variants thereof, is described in the book "Introduction to Linear Programming" by Garvin published by McGraw Hill, New York, 1960.

In the "simplex" method, the set of simultaneous linear functions is arranged as a first matrix which is stored in an appropriate storage device, the matrix comprising sets of elements which are defined as "columns" with respect to one ordinate of the matrix and as "rows" with respect to the other. In an exemplary application, the computer is controlled by the simplex linear programme device as follows, the procedure being defined in terms of matrix columns.

(1) The computer is programmed to successively scan or read the columns of the first matrix and the elements within each column to identify an element in one column which satisfies a predetermined condition. The selected column is referred to as a "pivot column" and is stored; the row therein which contains the selected element comprises the "pivot row" and the selected row and column define a "pivot point" in the matrix.

(2) The computer is programmed to perform operations on the entire first matrix, i.e. to carry out an iteration which comprises reading and operating on the entire first matrix using the stored pivot column to produce a second matrix which is stored. The operations referred to generally comprise arithmetic operations which involve the pivot point quantity and the quantities in the other elements of the matrix.

(3) The resultant second matrix is processed in the computer in the same manner as the first matrix; if upon reading the matrix a pivot column is found, it is stored and a second iteration on the entire second matrix is performed to yield a third matrix. The process continues with further iterations until a matrix is generated which is devoid of pivot columns. This final matrix is the solution matrix and gives the values for the component variables

[Pri

which when applied to the physical system, e.g. a refinery operation, optimizes or improves the operation thereof.

Although the simplex method is relatively simple it does not control the computer in a fully efficient manner. There is substantial lost time owing to the need for fast-operating components to await the completion of services performed by slower-working members. This lack of maximum utilization is particularly significant in view of the fact that present-day problems are so extensive and involve so many variables that it can take several days, or even weeks, computing in a computing engine, to produce results for some of the larger problems. The computing engines which may be used to solve these problems can be hired at a cost of up to £300 per hour of computing time. It can be seen that for resolving extensive problems the cost involved can be in the order of tens of thousands of pounds. It should, of course, be borne in mind that a problem solved by a typical computer in say four hours would require 130 years on a manual desk calculator and is therefore not practicably solvable except by automatic machine.

It is an object of the present invention to reduce the computation time required to produce results for such linear programming problems.

The present invention is based on the discovery that in the computation of a solution matrix, the machine processing of an iteration may be interrupted before the completion thereof and a new interation initiated if, in the course of the first iteration, a new pivot column is detected. The carrying forward of this process of partial concurrent iterations yields the solution matrix.

According to the invention a computer comprising quick-access storage, show-access storage and an arithmetic unit and being required to solve automatically a linear programming problem by means of an iterative algorithm, is arranged so that it automatically

(a) transfers data-representations from slow-acces to quick-access storage,

(b) transfers a suitable portion of data representations, as it becomes available by transfer as specified in (a), back and forth, without intermediate transfer to slow-access storage, between the quick-access storage and the arithmetic unit where the portion is processed in accordance with at least two iterations of the iterative algorithm during at least one performance of process (b), and

(c) returns the processed portion to slow-access storage.

As has been explained above an iterative algorithm for the solution of a linear programming problem requires that the data which is being processed at any one time is processed in conjunction with other data previously selected in accordance with the algorithm from the original data or from such data which has been partially processed in accordance with the algorithm. In this specification the term "pivot data" will be used to designate data which has been selected as described above and which is currently available for use.

In addition to providing a rule for the selection of pivot data the algorithm also provides a rule for discarding pivot data when it is no longer required. Thus the content of the pivot data undergoes a frequent modification and, in accordance with the present invention, the quantity of pivot data is also changed.

A computer which is arranged to operate according to our invention is preferably arranged to provide quick-access storage locations for pivot data representations and to continue every performance of process (b) until the back and forth transference has been repeated as many times as are necessary to enable all the available pivot data representations stored in the quick-access storage to be used. Preferably the computer is arranged to examine the processed portion of the data representations after every performance of process (b) and, if the portion is suitable for use as a pivot data and provided that there is room in quick-access storage, to retain a copy of the said portion in quick-access storage.

It is to be understood that the phrase "the back and forth transference has been repeated as many times as is necessary to enable all the available pivot data representations stored in the quick-access storage to be used" covers the case where no transference between the quick-access storage and arithmetic unit takes place because no available pivot data representations are available.

In the case of large linear programming problems it is desirable to record the problem to be solved as data representations on tape. It has already been explained that the problem can be represented as a matrix and the record on tape may be produced by recording each column of a matrix in turn on the tape; as an alternative each row may be recorded in turn. Tapes produced in both ways may be used by a computer arranged in accordance with the invention and in this specification a linear programming problem which has been recorded in this way will be described as being recorded vector by vector on tape. Each portion of data representations on the tape will be designated as a "vector" and where the problem is recorded column by column each vector represents the co-efficients associated with one of the variables of the problem; where the matrix is recorded row by row each vector represents the co-efficients of one of the equations or linear functions.

Where a computer is arranged, in accordance with the invention, for the solution of a linear programming problem recorded vector by vector on tape it is conveniently arranged so that it automatically

(i) transfers data representations from tape storage to quick-access storage,

(ii) transfers each vector in turn, as it becomes available after transfer as specified in (i), back and forth between the quick-access storage and the arithmetic unit where it is processed in accordance with the iterative algorithm in conjunction with pivot data stored in quick-access storage, and continues said transference until it has been repeated as many times as are necessary to enable all the available pivot data to be used.

(iii) examines each processed vector and, if it is suitable for use as pivot data and there is room in quick-access storage, retains a copy in quick-access storage for future use in process (ii), and

(iv) returns the processed vector to tape storage.

It is preferred that a computer arranged to operate as specified in the preceding paragraph is arranged so that it automatically counts the number of times each section of pivot data which corresponds to a vector selected in process (iv) is used in process (ii) and makes the portion non-available for further use when it has been used a number of times equal to the number of vectors stored on the tape.

An iterative algorithm for the solution of a linear programming problem normally provides that iterations continue until no pivot can be found and this indicates the end of the calculation. This provision may be incorporated in a computer arranged to operate according to the invention by providing that the computer automatically continues to recycle the data representations until no pivot data is contained in quick-access storage and there are no data representations suitable for selection as pivot data recorded in the slow-access storage.

The computer may be arranged so that it limits the maximum amount of pivot data representations stored in quick access storage, and hence limits the maximum time taken in any one performance of process (b) or process (ii), so as to balance as far as possible the time taken in transfer processes, i.e. processes (a), (c), (i) and (iv), with the time taken in computation processes, i.e. processes (b), (ii) and (iii).

In order that the invention can be more clearly understood, it will now be described with reference to the accompanying drawings, given by way of example, in which:—

Figure 1 shows a simple block schematic representation of a computer.

Figure 2 shows a more detailed block schematic of a computer.

Figure 3 shows a simple program flow diagram for a program to cause a computer to operate according to this invention.

Figure 4 shows a portion of punched paper tape.

The following description will show two examples of linear programming problems one involving a set of simultaneous linear equations having a unique solution and the other to maximise a set of simultaneous linear inequalities and for comparison purposes the manual working steps will be traced for solving the problems using, in the first case a simple iterative process and in the second case, the simplex method and the steps which data processing apparatus would take when operating according to the method of this invention to solve both problems. The solving of the problems will then be related to computers.

EXAMPLE 1.

The first example is a simple problem which comprises solving a set of four simultaneous equations with four variables. This first example is a simple theoretical one to provide some understanding of the invention before embarking upon the second, practical, problem. The problem is to solve for w, x, y and z in the following equations:—

$$\begin{aligned} 2w+x-3y+3z&=11\\ w-3x+y+z&=6\\ w-2x+2y+4z&=27\\ 2w-x+2y-z&=7 \end{aligned}$$

The first step in solving this problem using the simple iterative method is to rewrite the set of equations in the form of a matrix, as shown below:—

	w	x	y	z	Q
R_1	2	1	−3	3	11
R_2	1	−3	1	1	6
R_3	1	−2	2	4	27
R_4	2	−1	2	−1	7

By inspection it will be seen that this initial matrix comprises only the coefficients of the variables arranged with the coefficients of all the variables of each equation forming the rows and the coefficients of each variable forming the columns and the quantities forming a further column. It is now required to form the matrix into a unit matrix with each of the quantities in the right hand column denoting the value of one of the variables. This can be achieved by selecting each column in turn and using it as a so-called "pivot" column to operate on all of the columns of the matrix.

Taking column w as the first pivot column, it is required to reduce position R_1 of the column w to unity, reduce positions R_2, R_3 and R_4 to zero, and to operate on all the other columns of the matrix to maintain the initial relationship between the equations. Position R_1 of column w will be referred to as the "pivot point" of the pivot column.

By operating on all the columns of the initial matrix including the pivot column, with the pivot column, this being one iteration of the problem, a second matrix is produced as shown below:—

	w	x	y	z	Q
R_1	1	$\frac{1}{2}$	$-3/2$	$3/2$	$11/2$
R_2	0	$-7/2$	$5/2$	$-\frac{1}{2}$	$\frac{1}{2}$
R_3	0	$-5/2$	$7/2$	$5/2$	$43/2$
R_4	0	−2	5	−4	−4

This second matrix has been produced by dividing all the values in row R_1 of the initial matrix by the pivot point value. All the other values have been produced by the following set of rules which are standard for operating on a matrix: to find the value of any point in the second matrix form an imaginary rectangle on the initial matrix with the corresponding point on the initial matrix and the pivot point forming the diagonally opposed corners of the rectangle; the new point for the second matrix is given by subtracting, (the product of the values of the two diagonally opposed corners of the rectangle which do not include the pivot point, divided by the value of the pivot point) from (the value of the corresponding point in the initial matrix).

Hence to find position $R_3.y$ in the second matrix we form an imaginary rectangle having corners at points $R_1.w$, $R_1.y$, $R_3.y$ and $R_3.w$ on the initial matrix. Using the rule above position $R_3.y$ in the second matrix is given by $2-\dfrac{1\times -3}{2}$ which is 7/2.

Using the rules stated above a second iteration is executed to produce a third matrix using column x as the pivot column and point $R_2.x$ as the pivot point. This third matrix is as shown below:—

	w	x	y	z	Q
R_1	1	0	$-8/7$	$10/7$	$39/7$
R_2	0	1	$-5/7$	$1/7$	$-1/7$
R_3	0	0	$12/7$	$20/7$	$148/7$
R_4	0	0	$25/7$	$-26/7$	$-30/7$

Using the same rules, a third iteration is executed to produce a fourth matrix using column y as the pivot column and point $R_3.y$ as the pivot point. This fourth matrix is as shown below:—

	w	x	y	z	Q
R_1	1	0	0	$^{10}/_3$	$^{59}/_3$
R_2	0	1	0	$^{4}/_3$	$^{26}/_3$
R_3	0	0	1	$^{5}/_3$	$^{37}/_3$
R_4	0	0	0	$-^{29}/_3$	$-^{145}/_3$

Finally using the same rules, a fourth iteration is executed to produce a fifth matrix using column z as the pivot column and point $R_4.z$ as the pivot point. The fifth matrix is as shown below:—

	w	x	y	z	Q
R_1	1	0	0	0	3
R_2	0	1	0	0	2
R_3	0	0	1	0	4
R_4	0	0	0	1	5

By inspection it can be seen that the fifth matrix is a unit matrix and is the solution matrix. Also by inspection it can be seen from row R_1 that as the coefficient of w is unity and those of x, y and z zero $w=3$, similarly $x=2$, $y=4$, and $z=5$. In obtaining this result, four iterations were required to be executed on the whole matrix.

The method of operating data processing apparatus according to this invention will be referred to as the "polyplex" method. This polyplex method is not suited for use by a human workman but for explanation purposes the steps will now be explained. To solve the problem above we start by forming the initial matrix as used for the previous method, as is shown below:—

	w	x	y	z	Q
R_1	2	1	−3	3	11
R_2	1	−3	1	1	6
R_3	1	−2	2	4	27
R_4	2	−1	2	−1	7

It has been seen, in the previous method of solving this problem, that each of the columns in turn is used as a pivot column, and that the pivot point must be in a different row for each pivot column so that a unit matrix is produced.

To solve this problem using the polyplex method requires only one polyplex iteration instead of the previous four iterations. This one iteration produces a matrix as shown below:—

	w	x	y	z	Q
R_1	2	$\frac{1}{2}$	$-^{8}/_7$	$^{10}/_3$	3
R_2	1	$-^{7}/_2$	$-^{5}/_7$	$^{4}/_3$	2
R_3	1	$-^{5}/_2$	$^{12}/_7$	$^{5}/_3$	4
R_4	2	−2	$^{25}/_7$	$-^{29}/_3$	5

This matrix has been produced by merely rewriting column w as it was in the initial matriv and deciding that resultant column w in the new matrix will be a pivot column with point $R_1.w$ as the pivot point. Using the general rules as stated for the previous method, column x in the initial matrix has been operated on with resultant column w as a pivot column and point $R_1.w$ as the pivot point. It has been decided that resultant column x in the new matrix will be the second pivot column with point $R_2.x$ as the pivot point.

Resultant column y in the new matrix has been produced using the previously stated rules, by operating on column y in the initial matrix with resultant column w as a pivot column and then operating on the resultant with resultant column x as a pivot column. It has been decided that resultant column y in the new matrix will be the third pivot column with point $R_3.y$ as the pivot point.

Resultant column z in the new matrix has been produced using the previously stated rules, by operating on column z in the initial matrix with resultant column w as a pivot column, operating on the resultant with resultant column x as a pivot column and operating on the resultant with resultant column y as a pivot column. It has then been decided that resultant column z in the new matrix will be the fourth pivot column with point $R_4.z$ as the pivot point.

Resultant column Q in the new matrix has been produced, using the previously stated rules, by operating on column Q in the initial matrix with resultant columns w, x, y and z as pivot columns as described above. It can be seen that the same answers have been produced for w, x, y and z as were obtained using the previous method. It can also be seen that little is gained by using the polyplex method when a human workman is solving the equations as each step would have to be written and could not be done purely mentally. However when using a computing engine the polyplex method is far quicker than the previous method.

Referring now to Fig. 1 there is shown a simple block diagram of a data processing apparatus having a computing unit 10 and a slow-access storage device 11 linked by leads 12 and 13. In solving linear programming problems in practise a large amount of the processing time is taken in transferring data from the storage device 11 to the computing unit 10 to be processed and transferring the processed data back to the storage device 11.

We will now consider the data flow in the data processing apparatus for solving the above problem using both the known and polyplex methods.

Firstly using the known method the data processing apparatus is set up with the initial matrix stored in the storage device 11. Column w is transferred from the storage device 11 along lead 12 to the computing unit 10. Column w is stored in computing unit 10 in a location which is allotted to the pivot column. Column w is also stored in a second location in computing unit 10 which is allotted to the column which is to be operated on. The operation on column w with column w as a pivot column is then executed by the computing unit 10 and resultant column w of the second matrix is transferred along lead 13 to the storage device 11. Then column x is transferred from storage device 11 along lead 12 to computing unit 10, operated on with column w as a pivot column and then transferred along lead 13 back to storage device 11. This process is continued for each column until the second matrix is stored in storage device 11.

When storage device 11 contains the second matrix, column x of this matrix is transferred from the storage device 11 to the computing unit 10 and stored in the location allotted to the pivot column. Each of the columns of the second matrix is transferred, in turn, to the computing unit 10 and returned to the storage device 11 until the storage device 11 contains the third matrix.

This process is continued until the storage device 11 contains the fifth matrix which is the solution matrix. The problem has now been solved and the answer is read out of the apparatus.

Now to solve the problem using the polyplex method the computer is set up as before with the initial matrix stored in storage device 11. Column w is transferred from storage device 11 along lead 12 to computing unit 10, is stored in a first location allotted to a pivot column and returned along lead 13 to the storage device 11.

Column x is transferred from storage device 11 to computing unit 10 and is operated on with column w as a pivot column. The resultant column x is transferred to storage device 11 along lead 13 and is also stored in a second location in computing unit 10, which is allocated to a pivot column.

Column y is transferred from storage device 11 to computing unit 10 and is operated on with column w as the pivot column, and the resultant column is operated on with resultant column x as a pivot column. The resultant y column is transferred to the storage device 11 and also stored in a third location in computing unit 10, which is allotted to a pivot column.

This process is continued and when column Q is transferred to computing unit 10 it is operated on with resultant columns w, x, y and z as pivot columns and the resultant transferred to the storage unit 11. This resultant Q column is the solution to the problem, and it can be seen that it has been produced after the transfer of one (the initial) matrix to the computing unit 10 from the storage device 11, and the transfer of one matrix back to the storage device 11. The known method had required five such transfers to complete the problem. Therefore with the polyplex method the transfer time had been reduced to one fifth of the time taken using the known method.

In practice, of course, the problems are much larger than that of the example above, and can have matrices in the order of hundreds of columns and rows. With a problem having a finite solution involving a matrix having one hundred columns and rows, using the known method one hundred complete matrices would need to be transferred to and from the computing unit i.e. one hundred iterations, whilst using the polyplex method, if the processing apparatus is sufficiently large, only two complete matrices would need to be transferred to and from the computing unit i.e. two iterations.

In order that the importance of reducing the transfer time in data processing apparatus can be fully appreciated, a more detailed block diagram of a data processing apparatus is shown in Fig. 2.

The data processing apparatus shown in Fig. 2 has three main units, a working store 21, a programme control unit 22 and an arithmetic and logic operations unit or units 23. There is also a drum store (or stores) 24 connected via a transfer control unit 25 to the working store 21, and a tape unit (or units) 26 connected via a transfer control unit 27 to the working store 21. The units are interconnected by cables which are represented by solid lines on the drawing for cables for data flow, and dashed lines for cables for cables for control or instruction flow, arrows on the lines denoting the direction of flow. The cables 28 connecting working store 21 and programme control unit 22 only carry instruction words of the programme to and from the programme control unit 22. The working store 21 is a very fast access store comprising, for example, a magnetic core array, it being possible to read a word from this store in approximately 6 microseconds. The working store can have a capacity of, for example, between one thousand and fifty thousand alpha-numeric characters. The drum store 24 has a medium access speed, and comprises a drum of magnetizable material divided into strips round the drum, each strip being provided with a read/write head to allow access to and from its corresponding strip. In operation the drum is rotated so that all parts of the drum pass the read/write heads to allow alpha-numeric characters to be written onto or read from discrete locations on the strips. The average access time to read a character to or from a location on the drum is approximately 2.4 milliseconds, and the capacity of drum storage may be, for example, between thirty thousand to a quarter of a million alpha-numeric characters, depending on how many drums are provided. The tape unit (or units) 26 has a slow access speed and comprises a file reel containing magnetic tape which is wound onto and from a machine reel which remains on the unit. Between the reels are a number of read/write heads arranged so that writing an alpha-numeric character causes it to be spaced across the tape whilst it passes under the read/write heads from one reel to the other. The capacity of tape storage provided in a data processing apparatus can vary, for example, between two million and sixty four million alpha-numeric characters. To read or write a character onto or from tape requires 6.25 milliseconds if the required location is beneath the read/write heads but it can take up to ten minutes to read through the whole tape to find a required location.

To work a linear programming problem such as the one previously described but larger, the programme, comprising a number of instruction words for controlling the operations, is stored in the drum store 24. A block of instruction words is maintained in the working store 21 and transferred sequentially to and from the programme control unit 22 which controls the operations of the other units. The initial matrix is stored in the tape store 26 and is transferred sequentially a column at a time, to the working store.

To solve the problem described above with the data processing apparatus shown in Fig. 2 using the known method involves storing each pivot column in turn in the working store 21. Each of the columns of the matrix is read from one tape unit 26 via transfer control unit 27, which is controlled by the programme control unit 22, into the working store 21. A copy of the pivot column and the column to be operated on are transferred to the arithmetic unit 23, the operation is performed, and the resultant column is transferred to a second tape unit 26 (only one shown on the drawing). Therefore, for each operation in the arithmetic unit 23 a column must be read from tape unit 26. The arithmetic unit 23 comprises a number of electronic devices which operate at a speed which is much faster than the speed of operation of the tape unit 26, the speed of which is governed by the speed at which the tape passes the read/write heads. It can be seen that the arithmetic unit 23 is not operating all the time, but is waiting for columns to be read from the tape unit 26.

To solve the above problem using the polyplex method column w is read into the working store 21, from a first tape unit 26 containing the initial matrix, it is transferred to a pivot column location on the drum store 24 and is transferred back to a second tape unit 26 (only one shown on the drawing).

Column x is read into working store 21 from the first tape unit 26, a copy of column w is transferred to the working store 21, and both columns are transferred to arithmetic unit 23 where the operation is performed. The resultant column x is transferred via the working store 21 to the second tape unit 26 and also to a second pivot column location in the drum store 24.

Column y is transferred from the first tape unit 26 to the working store 21, and a copy of column w is transferred from the drum store 24 to the working store 21. Both columns are transferred to the arithmetic unit 23 where the operation on column y is executed, first resultant column y being transferred to the working store 21. A copy of pivot column x is now transferred to working store 21 and is then transferred with first resultant column y to the arithmetic unit 23. First resultant column y is operated on with pivot column x. The resultant column y is transferred via working store 21 to the second tape unit 26, and to a third pivot column location in drum store 24.

Thus when column z is read from first tape unit 26, three operations will be performed on it before it is transferred to second tape unit 26, similarly four operations are performed on column Q to produce the solution. The relative speeds of operation of arithmetic unit 23 and tape unit 26 are such that a plurality of opérations can be performed on a column in the arithmetic unit whilst one column is being read from tape unit 26.

What has been achieved is that the speed of operation of the data processing apparatus is being governed to a greater extent by the speed of operation of the fast operating arithmetic unit 23, instead of by the speed of the relatively slow operating tape unit 26, so increasing the operating efficiency of the machine.

EXAMPLE 2.

The second example is a simple practical problem related to a one-man woodworking business. In this business the man works an eight-hour day producing alpenstocks, bookshelves and chairs which sell at £3, £2 and £2 each respectively. He has the choice of working any one of three processes:—

Process P costs £6 an hour to run, in which time it produces two alpenstocks and three chairs.

Process Q costs £5 an hour to run, in which time it produces one alpenstock and two bookshelves.

Process R costs £7 an hour to run, in which time it produces one alpenstock, one bookshelf and two chairs.

The retail market for these items is up to nine alpenstocks, eleven bookshelves and nine chairs a day. The problem is finding the arrangement of processes which will produce the most profit.

This problem can be set up algebraically:—

If p, q and r are the numbers of hours worked on the three processes, then

Number of alpenstocks produced is:—

$$2p+q+r=9$$

Number of bookshelves produced is:—

$$2q+r=11$$

Number of chairs produced is:—

$$3p+2r=9$$

This problem as defined thus far has a unique solution and can be solved algebraically to give $p=1$ hour, $q=4$ hours and $r=3$ hours. The cost and revenue calculations using these figures show that a profit of £20 per day will be achieved.

The problem can be more fully defined and written in a modified form to produce a linear programming problem. This is achieved by converting the equations to inequations or restrictions and adding a further inequation. We now have:—

$$\begin{aligned} 2p+q+r &\leq 9 \\ 2q+r &\leq 11 \\ 3p \quad +2r &\leq 9 \\ p+q+r &\leq 8 \end{aligned}$$

The fourth restriction has been produced by knowing that production only continues for eight hours each day. There are also three further restrictions because the processes cannot be worked for a negative time, these are:—

$$p\geq o,\ q\geq o,\ r\geq o$$

To complete the linear programming problem it is required to realise what exactly is to be achieved. This is that processing should not continue for more than eight hours each day and that as much profit as possible should be made. With this expanded definition of the problem there is no longer a unique solution; the problem now is to find that solution which maximizes profit.

Considering process Q it can be seen that for each hour worked, £3 is realised on the alpenstock and £4 on the bookshelves at a cost of £5, hence £2 profit is made. Similarly P and R processes produce a profit of £6 and £2 an hour respectively.

Therefore the total profit is:—

$$6p+2q+2r=Z$$

where Z is the profit. This equation is usually called the "control function," and it is the control function that is to be maximized.

To complete the problem the inequalities are removed by introducing further variables s, t, u and v into the inequations to produce a set of equations, as below:—

$$
\begin{array}{lllllllll}
 & 2p+q+r & +s & & & & =9 \\
 & 2q+r & & +t & & & =11 \\
 & 3p+2r & & & +u & & =9 \\
 & p+q+r & & & & +v & =8 \\
\text{Maximize} & 6p+2q+2r & & & & & =Z
\end{array}
$$

The top four of the above simultaneous equations contain seven unknowns, hence, they have an infinite number of possible solutions. It is evident that the problem is to select the solution which maximizes the control function Z.

The above equations can be written as a matrix, which will be the initial matrix of the problem, as below:—

Labels		Quantity	p	q	r	s	t	u	v	Ratio Q/P
R_1	s	9	2	1	1	1				4.5
R_2	t	11		2	1		1			α
R_3	u	9	3		2			1		3
R_4	v	8	1	1	1				1	8
Control		0	−6	−2	−2					

The initial matrix has been formed as in the previous example, by using only the coefficients of the variables, the control row has been formed by moving the coefficients of the variables to the right hand side of the control function, so making them negative. The profit (control row, quantity column) is zero because in the initial matrix variables p, q and r all equal zero and $s=9$, $t=11$, $u=9$ and $v=8$.

Variables s, t, u and v have been included in a column under the heading "Labels" to show that the values in the quantity column are the values of the variables in the corresponding rows of the "Labels" column.

To solve this problem using the known simplex method, the matrix is examined and the column having the most negative value in the control row is used as the first pivot column. In the initial matrix above, column p has the most negative value in the control row. To find the pivot point the values in the p column are divided into the values in the quantity column, for each row, the point in the row having the lowest positive value of quotient being used as the pivot point. The lowest quotient is used so that the quantity column values are maintained positive, according to the previously stated restrictions.

Hence in the present problem the first pivot column is column p, and the pivot point is R_3.p. By operating on the whole of the matrix in accordance with column p as pivot column, with the rules stated in the first example, a second matrix is obtained as below:—

Labels		Quantity	p	q	r	s	t	u	v	Ratio Q/q
R_1	s	3		1	$-\frac{1}{3}$	1		$-\frac{2}{3}$		3
R_2	t	11		2	1		1			$5\frac{1}{2}$
R_3	p	3	1		$\frac{2}{3}$			$\frac{1}{3}$		α
R_4	v	5		1	$\frac{1}{3}$			$-\frac{1}{3}$	1	5
Control		18		−2	2			2		

By inspection it can be seen that the profit has now increased from zero to eighteen, with $s=3$, $t=11$, $p=3$, $v=5$, and q, r and u being zero. From the control row it can be seen that the profit can be increased by using column q as a pivot column. By dividing Q by q as stated above it is found that point R_1.q is the pivot point.

By operating on the whole of the second matrix with column q as pivot column, again using the rules stated for the first example, a third matrix is obtained as below:—

Labels		Quantity	p	q	r	s	t	u	v
R_1	q	3		1	$-1/3$	1		$-2/3$	
R_2	t	5			$5/3$	-2	1	$4/3$	
R_3	p	3	1		$2/3$			$1/3$	
R_4	v	2			$2/3$	-1		$1/3$	1
Control		24			$4/3$	2		$2/3$	

By inspection it can be seen from the third matrix that the profit has increased from eighteen to twenty-four, with $q=3$, $t=5$, $p=3$, $v=2$ and r, s and u zero. No further increase in profit can be achieved as the control row contains all positive quantities.

Now comparing the results above with the results obtained by solving algebraically, the profit has increased from £20 a day to £24, process P should be worked for three hours a day and process Q for three hours a day. Only six hours need be worked instead of eight and yet there is increased profit.

To solve this problem using the polyplex method the initial matrix is set up, as shown above for the simplex method and again below:—

Labels		Quantity	p	q	r	s	t	u	v
R_1	s	9	2	1	1	1			
R_2	t	11		2	1		1		
R_3	u	9	3		2			1	
R_4	v	8	1	1	1				1
Control		0	-6	-2	-2				

Using the polyplex method to maximise the control row, the first (left hand) column p is examined and is found to have a negative value in the control row. This will be the first pivot column. Dividing the values in the pivot column into the quantity column the lowest quotient is found in row R_3, so the pivot point is position R_3.p. The quantity column is operated on by column p as pivot column, using the rules stated for the first example, and is remembered. The next column, being column q, is operated on by column p as pivot column and the resultant column is inspected to see if there is a negative value in the control row. If the control row value of column q is negative, column q becomes the second pivot column, the values in column q are divided into the values in the remembered quantity column and the pivot point found. The quantity column is updated, i.e. operated on with resultant column q as pivot column, and column r is operated on by column p as pivot column and the resultant operated on by resultant column q as pivot column if it has been found to be a pivot column. Resultant column r is inspected to see if it has a negative value in the control row and if it has then it becomes the next pivot column.

This process is continued to the end of the matrix. For the present problem the second matrix is found to be as below:—

Labels		Quantity	p	q	r	s	t	u	v
R_1	q	3	2	1	$-1/3$	1		$-2/3$	
R_2	t	5		2	$5/3$	-2	1	$4/3$	
R_3	p	3	3		$2/3$			$1/3$	
R_4	v	2	1	1	$2/3$	-1		$1/3$	1
Control		24	-6	-2	$4/3$	2		$2/3$	

The above matrix has been produced by operating on column q and the quantity column with column p as pivot column. By inspection it was found that column q was a pivot column and so the resultant quantity column was operated on by column q as pivot column, having first found point R_1 q to be column, having first found point R_1.q to be the pivot point. The remainder of the columns have been operated on by both p and q as pivot columns and no other negative values were found in the control row. To complete the problem it remains to operate on column p with columns p and q as pivot columns and column q with column q as pivot column. This produces the solution matrix as below:—

Labels		Quantity	p	q	r	s	t	u	v
R_1	q	3		1	$-^1/_3$	1		$-^2/_3$	
R_2	t	5			$^5/_3$	-2	1	$^4/_3$	
R_3	p	3	1		$^2/_3$			$^1/_3$	
R_4	v	2			$^2/_3$	-1		$^1/_3$	1
Control		24			$^4/_3$	2		$^2/_3$	

The solution matrix being the same as the previous simplex solution matrix has been produced by 1 2/7 iterations compared with the 2 iterations with the simplex method. The advantages of the polyplex method are not fully utilized with such a small problem but are far more apparent with larger problems. For ease of description small problems have been described rather than a larger one which would be encountered in practise.

Relating this second problem to the data processing apparatus shown in Figure 2, using the simplex method, the problem is solved in almost identical manner to the first example. The initial matrix is set up in a first tape unit 26 and the quantity column is stored in the working store 21.

The initial matrix is scanned and the column having the most negative value in the control row is transferred to working store 21. The pivot point in the pivot column is determined in arithmetic unit 23 as described hereinbefore. Each of the columns of the initial matrix are then transferred to the working store 21 in turn and operated on in arithmetic unit 23 with the pivot column. The resultant columns are then stored in a second tape unit 26. When a complete iteration has been executed the second matrix in the second tape unit 26 is scanned, and the column having the most negative value in the control row is transferred to working store 21. This process is continued until there are no negative values in the control row, at which time the solution matrix has been obtained and the processing completed.

To solve the problem using the polyplex method the problem is set up in the same way as for simplex. Instead of scanning the initial matrix the first column is read from the first tape unit 26 into working store 21. If the column has a negative value in the control row position, the pivot point is determined and the quantity column is updated by operating on it with the pivot column in the arithmetic unit 23. The pivot column is stored in a pivot column location in drum store 24, and transferred to a second tape unit 26.

With the pivot column stored in the drum store 24 there is also stored an indication of how many times the pivot column must be utilised to operate on columns of the matrix. Each pivot column must operate on every column of the matrix and therefore the indication has a value representative of the number of columns in the matrix. Each time a pivot column is utilised to operate on a column the indication is reduced so that when the pivot column has operated on all the columns of the matrix it is erased from the drum store 24.

The process is executed as described for the first example with the difference that the quantity column is stored in the working store 21 and is updated each time a new pivot column is obtained. The process continues until there are no pivot columns in the drum store, at this point the problem has been solved. The solution matrix may be partly on the first tape unit 26 and partly on the second tape unit 26 and so part of it is transferred from one of the tape units to the other so that it is all on one tape.

In practise for solving linear programming problems the initial matrix is stored on more than one reel of tape. This is so that the processing need not be held up while tapes are being rewound. The tape units being able to read from, or write onto tape only whilst the tape is moving from the file reel to the machine reel. The number of tape reels and hence the number of tape units required to ensure this is two more than is required to hold the matrix.

Although it has been stated previously that for solving a problem involving a large number of linear functions by the polyplex method, all the pivot columns are automatically stored in the drum store 24, this may not always be possible. The number of pivot columns stored in the drum store 24 is determined by two main criteria, one is the storage space available for storing pivot columns on the drum, and the other is the number of operations the arithmetic unit 23 can deal with in the time taken to read a column from tape unit 26. For most efficient operation of the data processing apparatus the programme is adapted to ensure that the number of pivot columns stored in the drum store 24 is such that columns are being read from, and written onto the tape units 26 continuously. This ensures that the tapes keep moving past the read/write heads, so that no time is required for the tapes to be started or stopped when reading and writing and so that the speed of operation of the tape units is co-ordinated with the speed of operation of the faster arithmetic unit.

Although in both of the previous examples the solution has been produced using columns of the matrix, it should be understood that the problems can also be solved using rows of the matrix. By working with the row containing the pivot point and another row, the previously stated general rules can be used to solve a problem in similar manner to the working described above using columns.

Referring now to Figure 3 there is shown a program flow diagram from which the actual linear programming problem is compiled. With the problem set up with the initial matrix stored on tape the process begins at arrow 31.

The first instruction, in box 32, causes the first column to be read from tape. The next instruction, in box 33 causes any pivot columns in a pivot column store, to be used to operate on the column which has just entered the computing unit. A copy of the resultant column of the operation is then caused to be transferred to tape by the next instruction, in box 34.

The next instruction, in box 35, causes a branch operation, if the pivot column store has just emptied indicating the end of processing, the next instruction will be the one in box 36, which causes the full result matrix to be transferred to one tape, and the result to be printed out. If the pivot column store has not just emptied the other branch is followed and a second branch operation is performed as denoted by box 37. At this point if the pivot column store is full, then the next instruction will be back at the beginning, in box 32, to read the next column from tape. If there is space available in the pivot column store then a third branch instruction in box 38 causes the resultant column to be examined to see if it can be used as a pivot column. The sequence of operations reverts to instruction 32 if the column is not suitable as a pivot column. If the resultant column is suitable as a pivot column then the other branch is followed and the instruction in box 39 is executed. This instruction causes the resultant column to be transferred to the first free position in the pivot column store. The next operation shown in box 40 causes the quantity column which is stored in the working store to be updated by using the new pivot column and operating on the quantity column.

The cycle of operations then begins again at instruction 32. The cycling of these operations continues until, as previously explained above, the pivot column store becomes empty, denoting that the problem has been solved.

For solving a set of simultaneous equations such as shown in the previous first example, branch instruction 38 would not be required as each column is to be used as a pivot column, also instruction 40 would not be required.

Branch instruction 38 can be expanded to keep a running check on the pivot column store as to how full it is and the rate at which pivot columns are being found during processing. To operate this check the increment in the profit to be achieved by a potential pivot is calculated (before the pivot-operation is started) and if this is less than a "nominal ascent" the pivot is rejected. Taking the two extremes, if, on the one hand, only a small number of potential pivot columns are found, the set ascent value is low and all potential pivot columns are transferred into the pivot column store and utilized, however, if, on the other hand, an extremely large number of potential pivot columns are found then the set ascent value is high and only those potential pivot columns producing an ascent above the set value are transferred into the pivot column store and utilized. This arrangement produces a still further improvement in the efficiency of operation of the data processing apparatus.

In the program flow diagram shown the important operation steps have been illustrated, in practise however, a far more complicated flow diagram is used.

From the flow diagram a detailed programme is written which is then reproduced, for example, on punched paper tape. Each type of data processing apparatus has its own set of operation codes which cause the data processing apparatus to perform operations.

For example with the Ferranti Mercury digital computer some of the operation codes are as follows:—

42 — Add
50 — Multiply
68 — Read from Drum
69 — Write to Drum
82 — Print a Line
86 — Read Tape

The Mercury computer operates with instruction words which are twenty digits long, the first seven digit positions containing the function or operation code, the next three digit positions containing a modifying code, and the last ten digit positions containing an address. The modifying code is used when it is required to modify the address portion of the instruction word. The address portion of the instruction word comprises the address of a location in storage to or from which data is to be transferred depending upon the operation code. When the operation code is "multiply" then the address portion is the address of the data by which other data is to be multiplied. When the operation code is "read tape" then the address portion denotes the address of the location in storage to which the data read from tape is to be transferred.

Referring now to Figure 4 there is shown a portion of punched paper tape 41 suitable for use with the Mercury computer. The tape 41 comprises a row of guide holes 42 and a plurality of data punchings 43. Each column across the tape 41 comprises one guide hole 42 and up to five data punchings 43, two above the guide hole and three below. Each column across the tape 41 represents one alpha-numeric character. When the data punchings are preceded by a blank column then the following columns represent figures or special characters. The data punchings 43 in Figure 4 show the positions of punching to represent the numerals one to nine and zero, as indicated on the tape above the columns. When the data punchings 43 are preceded by a column as shown by column 44 then the following columns represent alphabetic characters. In Figure 4 the alphabetic characters which would be represented by data punchings 43 when preceded by column 44, are shown beneath the columns on tape 41.

The program on punched tape is built up by writing a set of instruction words which will cause the computer to operate as shown by the flow diagram, and then punching the instruction words in code form onto punched paper tape using a key operated machine. The program on the punched tape can then be fed into the computer, via an input device, to storage, when the computer is being set up prior to calculating.

The program is written from the flow diagram using the instruction word format and instruction codes of the computer in which it is to be utilized. The program need not be represented on punched paper tape but can be on any medium that can be read by an input device of the computer in which it is to be utilized, for example punched cards or magnetic tape.

Although in the above particular description the working store was described as comprising an array of magnetic cores and the pivot column store was in the form of a magnetic drum, the present invention can be utilized with any digital processing apparatus, suitable for solving linear programming problems.

Claim 1 of our copending patent application 19464/62 (published with the final number 1,039,142) reads:—

A programming means which is adapted to control a computer which comprises quick-access storage, slow-access storage and an arithmetic unit in such a manner that the computer is caused to solve a linear programming problem by means of an iterative algorithm, the programming means being formed so as to cause the computer to

(a) transfer data representations from slow-access to quick-access storage,
(b) transfer as it becomes available after transfer as specified in (a), a suitable portion of data representations back and forth, without intermediate transfer to slow-access storage, between the quick-access storage and the arithmetic unit and to process it therein in accordance with at least two iterations of the iterative algorithm during at least one performance of process (b), and
(c) return the processed portion to slow-access storage.

WHAT WE CLAIM IS:—

1. A computer which comprises quick-access storage, slow-access storage and an arithmetic unit and which is arranged to solve automatically a linear programming problem by means of an iterative algorithm, the computer being arranged so that it automatically

(a) transfers data representations from slow-access to quick-access storage,
(b) transfers a suitable portion of data representations, as it becomes available by transference as specified in (a), back and forth, without intermediate transfer to slow-access storage, between the quick-access storage and the arithmetic unit where the portion is processed in accordance with at least two iterations of the iterative algorithm during at least one performance of process (b), and
(c) returns the processed portion to slow-access storage.

2. A computer according to claim 1, which is arranged to provide quick-access storage locations for pivot data representations and to continue every performance of process (b) until the back and forth transference has been repeated as many times as are necessary to enable all the available pivot data representations stored in the quick-access storage to be used.

3. A computer as claimed in claim 2, which is arranged to that it examines the processed portion of data representations after every performance of process (b) and, if the portion is suitable for use as pivot data and provided there is room in the quick-access storage, it retains a copy of the said portion in quick-access storage.

4. A computer which comprises quick-access

storage, tape storage and an arithmetic unit and which is arranged to solve, by means of an iterative algorithm, a linear programming problem which is recorded vector by vector on tape, the computer being arranged so that it automatically

(i) transfers data representations from tape storage to quick-access storage,

(ii) transfers each vector in turn, as it becomes available after transfer of specified in (i), back and forth between the quick-access storage and the arithmetic unit where it is processed in accordance with the iterative algorithm in conjunction with pivot data representations stored in quick-access storage and continues said transference until it has been repeated as many times as are necessary to enable all the available pivot data to be used,

(iii) examines each processed vector and, if it is suitable for use as pivot data and if there is sufficient room in quick-access storage, retains a copy in quick-access storage for future use in process (ii), and

(iv) returns the processed rector to tape storage.

5. A computer according to claim 4, which is arranged so that it automatically counts the number of times each section of pivot data which corresponds to a rector selected in process (iv) is used in process (ii) and makes the portion non-available for further use when it has been used a number of times equal to the number of rectors stored on the tape.

6. A computer which is arranged so that it automatically continues to re-cycle the data representations as specified in any one of the preceding claims until no pivot data representations are contained in quick-access storage and there are no data representations suitable for selection as pivot data recorded in the slow access storage.

7. A computer according to any one of claims 2—6 which is arranged so that it limits the maximum amount of pivot data representations stored in quick access storage so as to balance as far as possible the time taken in transfer processes with the time taken in computation processes.

8. A computer according to claim 1 arranged to operate substantially as hereinbefore described with reference to the accompanying drawings.

J. WOOLARD,
Agent for the Applicants,
Chartered Patent Agent.

Leamington Spa: Printed for Her Majesty's Stationery Office, by the Courier Press (Leamington) Ltd.—1966. Published by The Patent Office, 25 Southampton Buildings, London. W.C.2, from which copies may be obtained.

FIG.1.

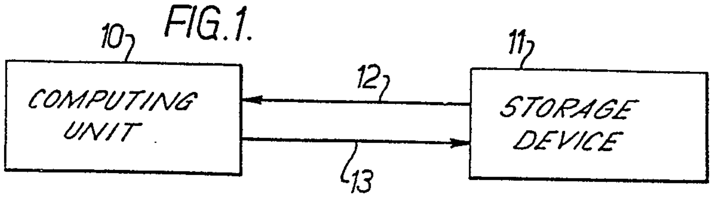


FIG.2.

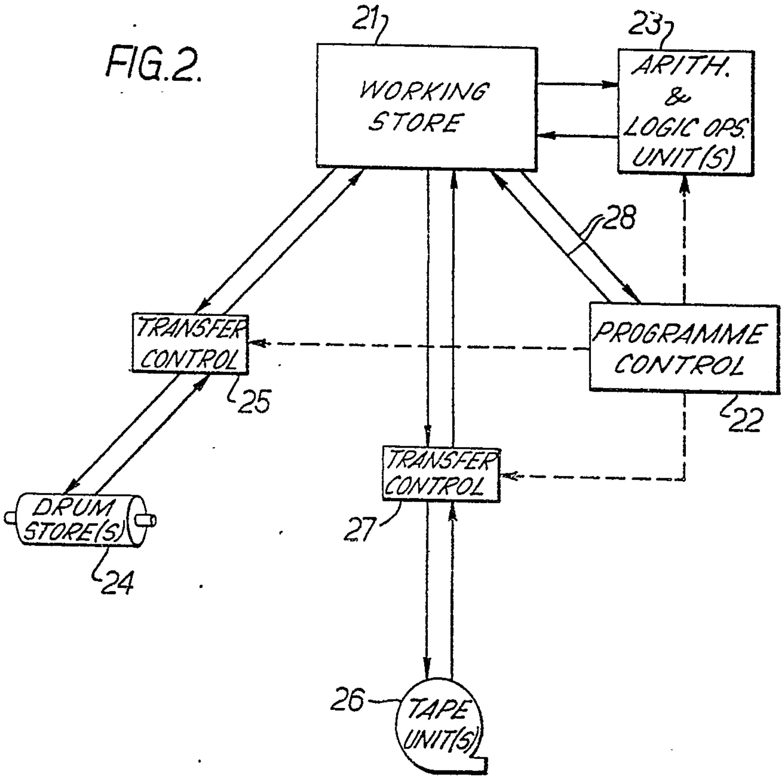
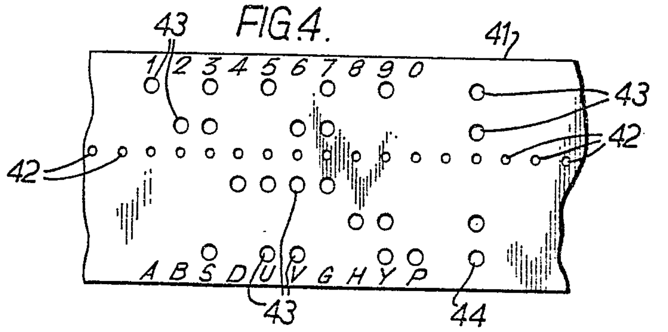
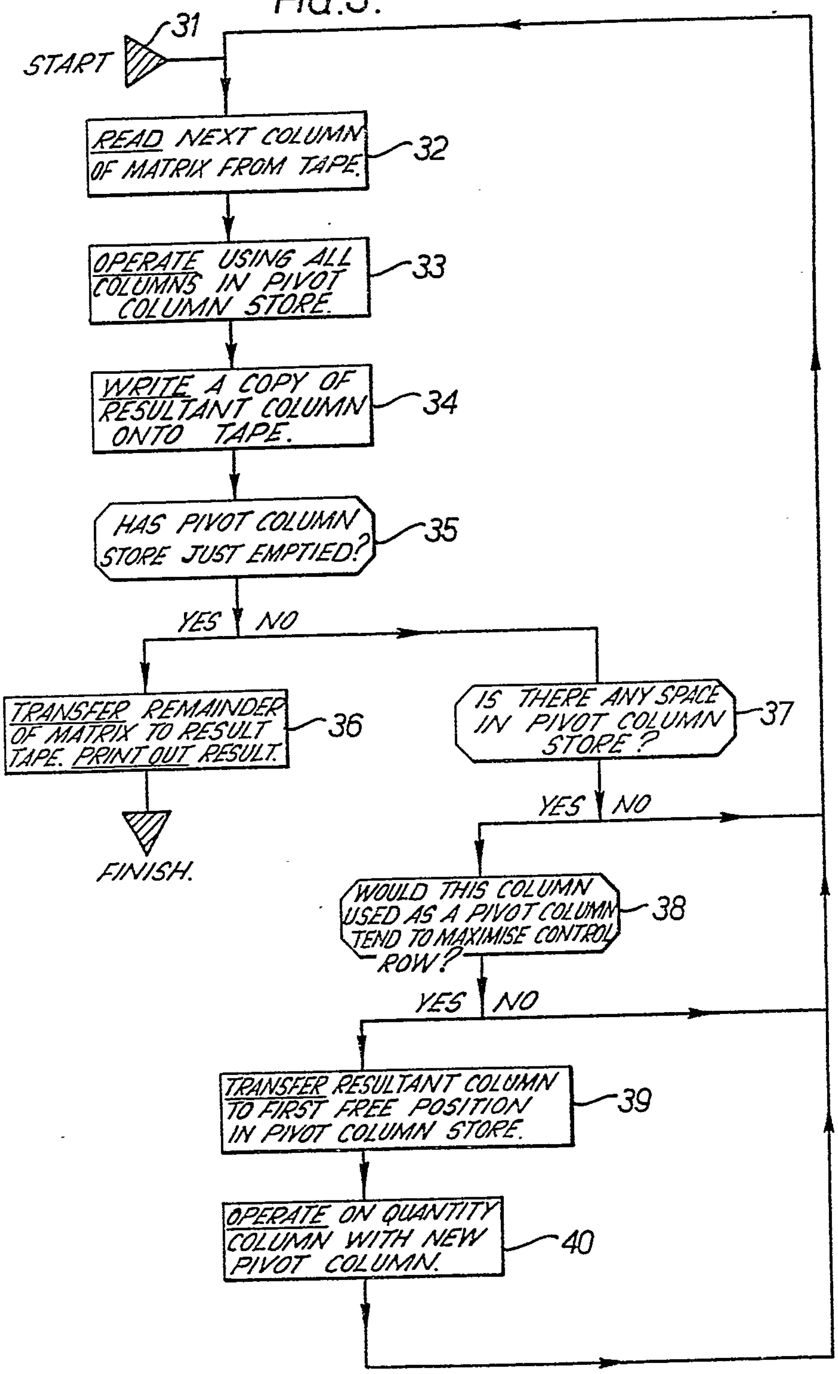


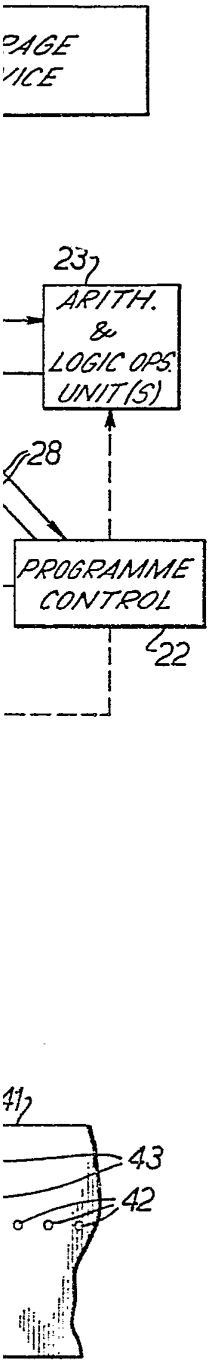
FIG.4.



1039141 PROVISIONAL SPECIFICATION
2 SHEETS This drawing is a reproduction of the Original on a reduced scale
Sheets 1 & 2

FIG.3.





1039141 PROVISIONAL SPECIFICATION
2 SHEETS *This drawing is a reproduction of the Original on a reduced scale*
Sheets 1 & 2

FIG.1.

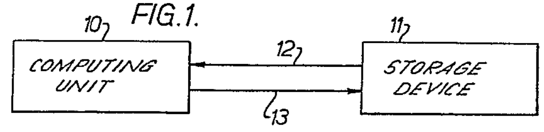


FIG.2.

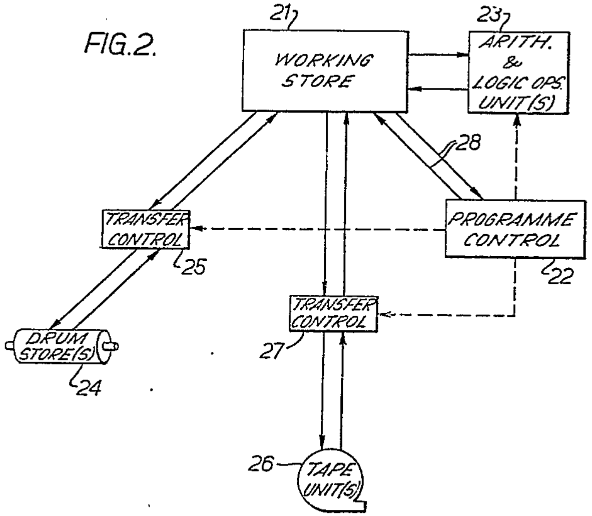


FIG.4.

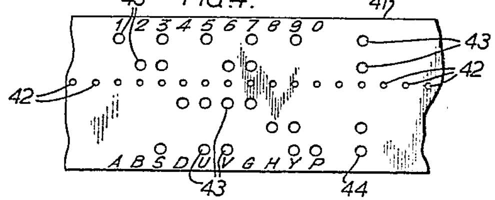


FIG.3.

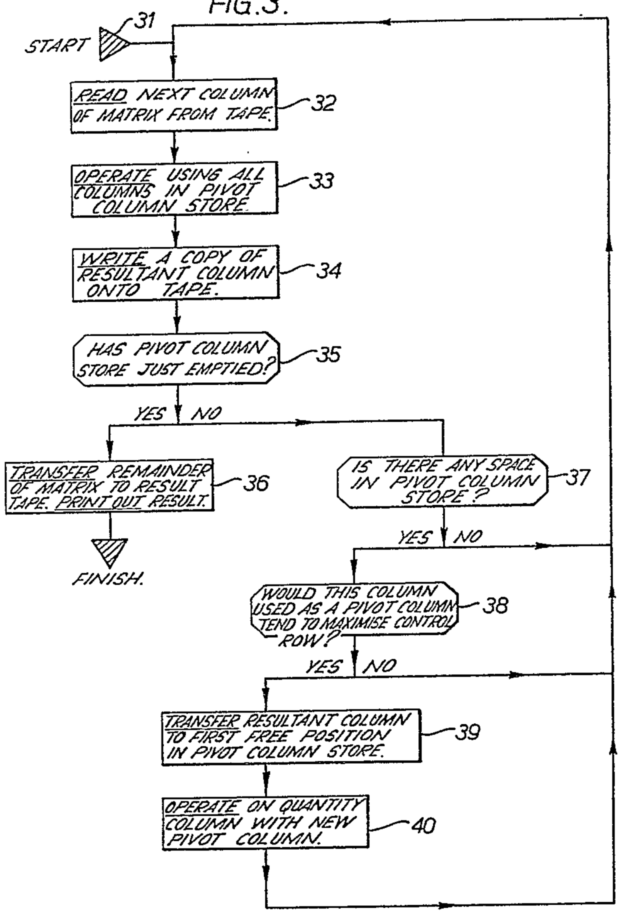


FIG.1.

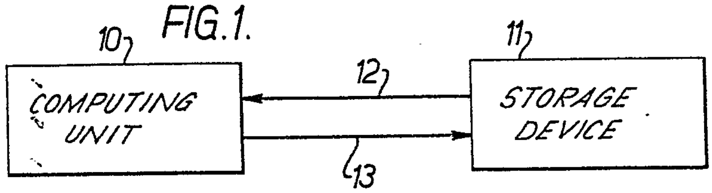


FIG.2.

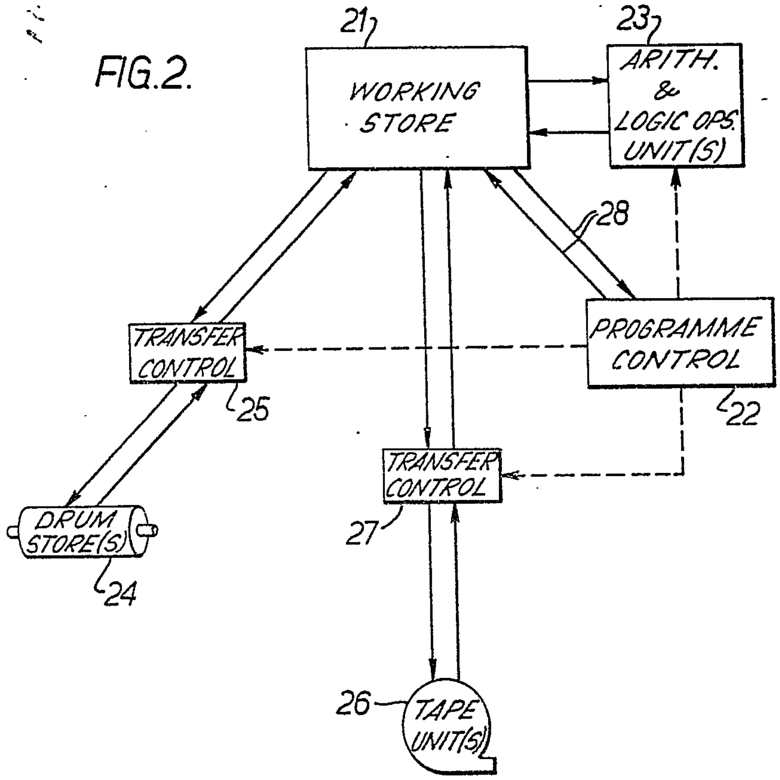
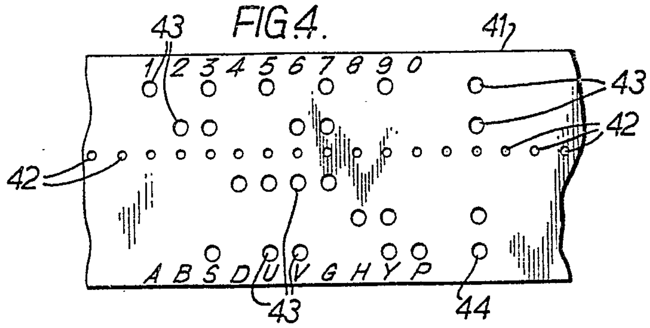


FIG.4.



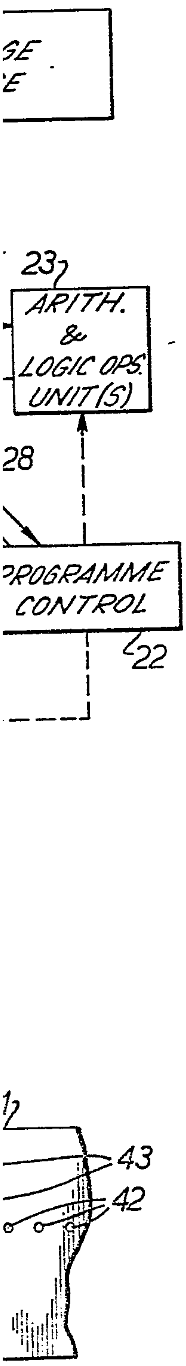
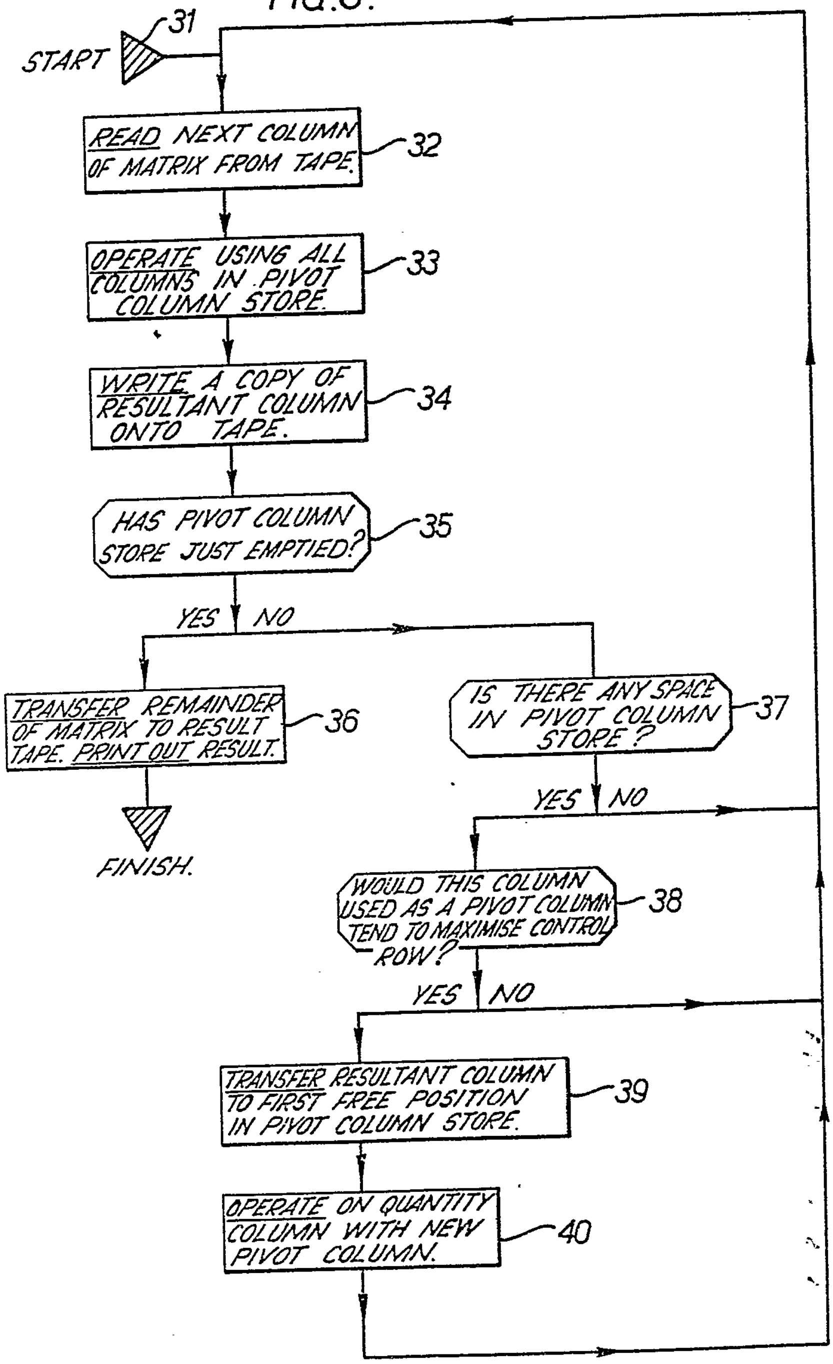

GE
E
23
ARITH. & LOGIC OPS. UNIT(S)
28
PROGRAMME CONTROL
22
1
43
42

FIG.3.


START
31
READ NEXT COLUMN OF MATRIX FROM TAPE.
32
OPERATE USING ALL COLUMNS IN PIVOT COLUMN STORE.
33
WRITE A COPY OF RESULTANT COLUMN ONTO TAPE.
34
HAS PIVOT COLUMN STORE JUST EMPTIED?
35
YES
NO
TRANSFER REMAINDER OF MATRIX TO RESULT TAPE. PRINT OUT RESULT.
36
FINISH.
IS THERE ANY SPACE IN PIVOT COLUMN STORE?
37
YES
NO
WOULD THIS COLUMN USED AS A PIVOT COLUMN TEND TO MAXIMISE CONTROL ROW?
38
YES
NO
TRANSFER RESULTANT COLUMN TO FIRST FREE POSITION IN PIVOT COLUMN STORE.
39
OPERATE ON QUANTITY COLUMN WITH NEW PIVOT COLUMN.
40

1039141 COMPLETE SPECIFICATION
2 SHEETS This drawing is a reproduction of the Original on a reduced scale
Sheets 1 & 2

FIG.1.

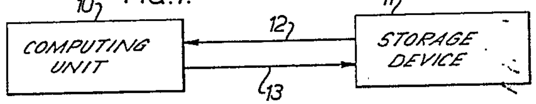


FIG.2.

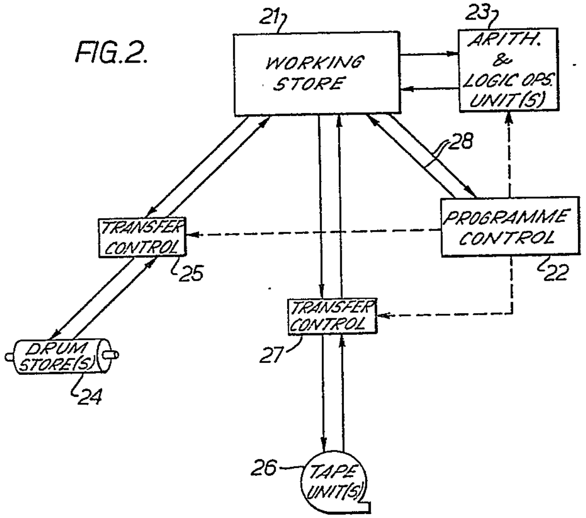


FIG.4.

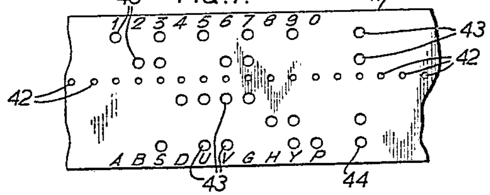


FIG.3.