M. OPRESCU
M. CHIRICĂ
St. NICULESCU

Solución de los problemas de programación lineal sobre calculadores electrónicos con coma fija

El fin de la presente obra es de poner a disposición de los que programan, el algoritmo y la esquema lógica detallada para solucionar los problemas de programación lineal cuya programación sobre los calculadores electrónicos con coma fija presenta una dificultad especial.

El análisis del desarrollo de los cálculos en el algoritmo simplex ordinario, conduce a la observación de que el conjunto de los valores numéricos, que constituyen el cuadro del simplex, en lugar de ser transformado podrá ser reconstituido por completo a cada iteración, partiendo del conjunto de los datos iniciales, o sea, del primer cuadro simplex. En este sentido, para pasar de una solución a otra, que da un valor mejor a la función-objetivo, basta con inversar la matriz de la base y empezar los cálculos partiendo de dicho revés.

En el cálculo aplicado, las restricciones del problema de programación lineal

$$\begin{cases} Max\ CX \\ AX \leqslant D \\ X \geqslant 0, \end{cases} \tag{1}$$

son transformadas frecuentemente en ecuaciones al añadirse m variaciones de igualación.

Este procedimiento de igualación da las formas

$$AX + I_m U = D \tag{2}$$

$$X \cdot U \geqslant 0 \tag{3}$$

donde $A = (a_{ij})$ es una matriz mxn; $X = (x_j)$ es un vector-columna con n componentes; I_m es la matriz-unidad mxm, D es el vector-columna cuyos componentes son los términos libres de las restricciones, y los m componentes del vector-columna U son las variaciones de igualación.

La principal dificultad de cálculo en la solución de los problemas de programación lineal es encontrar la solución del sistema (2) y evaluar la función-objetivo C.X. El problema de la evaluación de C.X. es tratado corrientemente a través del procedimiento de añadir otra variación V y mediante la consideración del sistema

$$\begin{cases} -CX + 1V + O_m^l U = O \\ AX + O_m^c V + I_m U = D \end{cases} \tag{4}$$

de $m + 1$ ecuaciones con $n + m + 1$ variaciones, donde O_m^l y O_m^c son respectivamente el vector-línea y el vector-columna de m componentes O.

Una solución $\overline{X}$, $\overline{U}$, $\overline{V}$, del sistema (4) satisface también las relaciones $A\overline{X} + I_m\overline{U} = D$, $\overline{V} = C\overline{X}$, de modo que el valor de la forma lineal CX es dato por el valor de V.

Para problemas con restricciones bajo forma de igualdades se pueden encontrar soluciones a través del procedimiento de eliminación, que presentamos a continuación.

Apuntaremos con

$A = (a_{ij})$, $(1 \leqslant i \leqslant p,\ 1 \leqslant j \leqslant q)$ una matriz no singular, y con a_j sus columnas;

$I_p = (e_{ij})_{i,\,j\,=\,1\,\ldots\,p}$ y e_j respectivamente la matriz-unidad y sus columnas;

$Q(i, j)$ el operador que al aplicarse a la matriz $A = (a_{ij})$ da la matriz $A' = (a'_{ij})$, o sea $A' = Q(i, j)\,A$, cuyos elementos se calculan de la siguiente manera:

$$\begin{cases} a'_{il} = \dfrac{a_{ij}}{a_{i1}} & l = 1, 2, \ldots, q \\ a'_{kl} = a_{k1} - a_{kj}\dfrac{a_{i1}}{a_{ij}} & k = 1, 2, \ldots, p; \quad k = \text{i}. \\ & l = 1, 2, \ldots, q; \end{cases} \tag{5}$$

Sea J_i la matriz obtenida de la matriz-unidad I_p mediante el reemplazo de la columna i de esta matriz con una columna $-\dfrac{1}{a_{ij}}\,a'_j$. El vector a'_j difiere de a_j solamente en la posición i donde $a'_{ij} = -1$. Tenemos:

$$\begin{cases} J_i e_k = e_k \text{ para } k \neq i \\ J_i a_j = e_i \end{cases} \tag{6}$$

Se nota el hecho de que para una matriz A y $a_{ij} \neq 0$, la operación $Q(i,j)A$ es equivalente al producto $J_i A$. De las relaciones (6) resulta que el revés de J_i se puede obtener de I_p reemplazando su columna i con la columna a_j.

El resultado precedente puede servir al cálculo de cerca a cerca de la matriz-revés a una matriz no singular A de $m \times m$ elementos. Para esto se parte de la matriz-unidad

$$I_m = (e_1, e_2, \ldots, e_m)$$

y se reemplaza sucesivamente $e_1, \ldots e_m$ con $a_1, \ldots, a_m$, formando de este modo las matrices $A_1, A_2, \ldots, A_m = A$. Tenemos :

$$A_1^{-1} = J_1 I_m = J_1$$

$$A_2^{-1} = J_2 A_1^{-1} = J_2 J_1$$

$$\ldots\ldots\ldots\ldots\ldots$$

$$A_m^{-1} = J_m A_{m-1}^{-1} = J_m J_{m-1} \ldots J_1 \qquad (7)$$

En conclusión, una solución del sistema

$$AX = D \qquad (8)$$

de p ecuaciones con q variaciones, $q \geqslant p$, sin restricciones de X, se puede encontrar mediante una columna de p operaciones de la forma

$$Q(i_p, j_p) \ldots Q(i_2, j_2)\, Q(i_1, j_1)\, (A, D)$$

en $p \cdot (q + 1)$ cuadros sucesivos (A, D) con $i_p \neq i_{p-1} \neq \ldots \neq i_1, y j_1, j_2, \ldots j_p \leqslant q$.

Entonces el resultado de la $(q + 1)$ columna es la expresión de D, como una combinación lineal de los vectores de base. Por consiguiente, los elementos de D dan una solución del sistema (8) y son los valores correspondientes a las variaciones de la descripción de la base (los valores de todas las demás variaciones siendo considerados 0).

Esto corresponde para operar la combinación repetida de las ecuaciones de (8) hasta que de la multitud de las q variaciones, p aparecen solamente en una ecuación, el resto de q-p variaciones siendo iguales a 0.

Si a un paso $k < p$ la aplicación de $Q(i_k j_k)$ no es posible, o sea si no existe a_{ij}, de tal modo que $a_{ij} \neq 0$, $i \neq i_{k-1}, i_{k-2}, \ldots, i_1$, entonces A tiene el rango $k-1$ y sólo existe una solución en el caso en que los componentes que no son nulos de la columna $(q + 1)$ de (A, D) aparecen en las líneas $i_1, i_2, \ldots, i_{k-1}$ y tienen los valores $x_{j_1}, x_{j_2}, \ldots, x_{j_{k-1}}$

En la práctica de cálculo se suele acostumbrar que, despues de normalizar las columnas (o las líneas) de una matriz, sea elegido $Q(i_k j_k)$ para las operaciones de eliminación, de tal modo que i_k sea la línea en que aparezca el menor elemento de la columna j_k. Esta técnica, que persigue reducir los errores de redondear y de corriente, se llama establecimiento de la posición de la medida. Los elementos de la columna de la parte derecha de la matriz deben ser permutados de manera conveniente para dar en orden los componentes de la solución.

Las variantes de cálculo del método simplex son distintas, en primer lugar, debido al modo en que se efectua la eliminación. Mediante el método del producto revés $(A_0 B)^{-1}$, donde B es una descripción de la base, se preparan las matrices de los pasos sucesivos de eliminación para reanudarse en las iteraciones siguientes, mediante la indicación de la columna en que estas matrices son distintas a la matriz-unidad. Así, la primera línea de $(A_0 B)^{-1}$,

$(A_0B)^{-1}e_1$, puede ser calculada multiplicando a e_1 a la izquierda con la transpuesta de la matriz de eliminación en orden inverso de los pasos de la eliminación.

Partiendo de dicho resultado, la primera línea de la matriz transformada, o sea

$$A_0^*(A_0B)^{-1}e_1 = [e_1^*(A_0B)^{-1}A_0]^*$$

. . . . . .

puede ser calculada y se pueden encontrar el índice q y su menor componenta negativa. Entonces la columna q de $(A_0B)^{-1}A_0$, y de $(A_0B)^{-1}D_0$ se pueden obtener mediante la multiplicación simultanea de D_0 y de la columna q de A_0, a la izquierda, mediante el producto de la eliminación de las matrices en el orden de los pasos de eliminación correspondientes. Las correlaciones de las componentas correspondientes de la columna q de $(A_0B)^{-1}A_0y$ de $(A_0B)^{-1}D_0$ pueden ser calculadas, y el número natural p para el paso de eliminación $Q(p, q)$ se puede encontrar conforme a la siguiente

Teorema 1. Dado el sistema

$$(A, D) = (A_0B)^{-1}(A_0, D_0)$$

$D \geqslant O$, donde B corresponde a la descripción de la base $(i_1, i_2, \ldots, i_{m+1})$ con $i_1 = n + 1$, la operación de eliminación $Q(p, q)$ $(2 \leqslant p \leqslant m + 1$, $1 \leqslant q \leqslant n + m + 1)$ es definida y resulta un nuevo sistema

$$(A', D') = (A_0B')^{-1}(A_0, D_0),$$

con $d'_1 \geqslant d_1 y D' \geqslant O$, donde B' corresponde a la descripción de la base obtenida mediante el reemplazo de i_p con q en líneas $(i_1, i_2, \ldots, i_{m+1})$, si

$$(a) \qquad a_{1q} < 0$$

$$(b) \qquad \frac{d_p}{a_{pq}} = \min_k \frac{d_k}{a_{kq}}$$

para cualquier k, para el cual $a_{kq} > 0$.

O sea, la nueva matriz de eliminación, o, practicamente, la columna en que ella es distinta a la matriz-unidad, se puede indicar.

El algoritmo descrito más arriba puede ser seguido con mucha facilidad en un ejemplo.

Consideraremos el siguiente problema con $m = 3$ restricciones y $n = 2$ variaciones.

Que se maximine la forma lineal $2x_1 + 5x_2$ con las restricciones:

$$x_1 + 4x_2 \leqslant 24$$
$$3x_1 + x_2 \leqslant 21$$
$$x_1 + x_2 \leqslant 9$$
$$x_1,\ x_2 \geqslant 0$$

Apuntaremos con $x_3 = 2x_1 + 5x_2$

Introduciendo las variaciones de igualación x_4, x_5, x_6 podemos escribir :

$$\begin{aligned} -2x_1 - 5x_2 + x_3 &= 0 \\ x_1 + 4\,x_2 \quad + x_4 &= 24 \\ 3x_1 + x_2 \quad + x_5 &= 21 \\ x_1 + x_2 \quad + x_6 &= 9 \end{aligned}$$

Arreglamos los cálculos de la siguiente manera :

	$(A_0B)^{-1}$	q	a_i	τ_i	p	Base	A_0	D_0
I Fase	1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1	2	0 24 21 9	—5 4 1 1	2	3 4 5 6	—2 —5 1 0 0 0 1 4 0 1 0 0 3 1 0 0 1 0 1 1 0 0 0 1	0 24 21 9

donde q nos indica el vector que entrará en la base; éste representa el *índice* de la columna A_0 para la cual el producto escalar $b_1 a_q = \min_{1 \leqslant j \leqslant m+n+1} \{b_1 a_j / b_1 a_j < 0\}$;
cuando $q = 0$ el algoritmo se ha terminado;

s_i representa los componentes del vector obtenido como resultado del producto $(A_0B)^{-1}D_0$. Estos componentes forman a cada etapa una solución posible de base;

τ_i representa los componentes del vector obtenido como resultado del producto $(A_0B)^{-1}a_q$;

p nos indica el vector de base que debe ser reemplazado. Representa el índice de la correlación $\frac{s_i}{\tau_i}$ minima positiva, o sea $\frac{s_p}{\tau_p} = \min_{1 \leqslant i \leqslant m+1} \left\{\frac{s_i}{\tau_i} \geqslant 0\right\}$

Los componentes revés a la nueva base se calculan así:

$$\begin{cases} b'_{pj} = \dfrac{b_{pj}}{\tau_p} & j = 1, 2, \ldots, m+1 \\ b'_{ij} = b_{ij} - \tau_i\, b'_{pj} & i \neq p \qquad i, j = 1, 2, \ldots m+1 \end{cases}$$

A continuación, el cálculo puede ser seguido en estos cuadros :

	$(A_0B)^{-1}$				q	s_i	τ_i	p	Base
II Fase	1	5/4	0	0	1	30	—3/4	4	3
	0	1/4	0	0		6	1/4		2
	0	—1/4	1	0		15	11/4		5
	0	—1/4	0	1		3	3/4		6
III Fase	1	1	0	1	0	33			3
	0	1/3	0	—1/3		5			2
	0	2/3	0	—11/3		4			5
	0	—1/3	0	4/3		4			1

Utilizando la descripción de la base de la última fase, constatamos que la solución óptima es :

$$x_1 = 4$$
$$x_2 = 5$$
$$x_3 = 33$$
$$x_5 = 4$$
$$x_4 = x_6 = 0$$

En conclusión, la variante del algoritmo simplex revisado expuesto, exige que en la iteración k se encuentre una línea de índice p y una columna de índice q de acuerdo con las exigencias de la teorema 1 y aplicar una operación de eliminación $Q(p_k, q_k)$ sobre una matriz de orden $(m+1) \cdot (n+m+2)$, que inicialmente es (A_0D_0) Este método es distinto al simplex ordinario debido a que los datos iniciales (A_0D_0) no son modificados, sino solamente el revés de la base $(A_0B_k)^{-1}$ es modificado mediante multiplicación a la izquierda de la forma $(A_0B_{k-1})^{-1}$ con la matriz correspondiente a la eliminación $Q(p_k, q_k)$.

Se puede observar que los datos modificados pueden ser obtenidos a través de la multiplicación $(A_0B_k)^{-1}$. (A_0D_0), puesto que la multiplicación con $(A_0B_k)^{-1}$ suma las primeras k operaciones de eliminación. En todo caso, solamente la primera línea de dicho producto sirve para encontrar a q_{k+1}, o sea solamente las columnas a_{q+1} y p_0 sirven para encontrar a p_{k+1}.

II. DESCRIPCÓN DE LA ESQUEMA LÓGICA DE SOLUCIÓN DE LOS PROBLEMAS DE PROGRAMACIÓN LINEAL CON LOS CALCULADORES ELECTRÓNICOS CON COMA FIJA.

a) Los datos de entrada constan en

— los $(m + 1) \cdot (n + m + 2)$ elementos (A_0D_0) apuntados en columnas;

— la descripción de la base señalada en los puntos l_1, l_2, ..., l_{m+1}, su contenido siendo modificado durante la aplicación del algoritmo;

— I_{m+1} revés a la matriz compuesta por las $m+1$ columnas de A_0 contadas $n+1$, $n+2$, ..., $n+m+1$ de acuerdo con la descripción de la base apuntada en líneas. Los puntos en que son indicados los elementos de la matriz I_m serán ocupados, durante los cálculos, por los elementos revés a la base corriente $(A_0B_k)^{-1}$;

— los constantes del tipo real $(+1)$, (-1), ε, 0, 0,1; los constantes del tipo entero 1, m, n, $m+1$, $n+1$, $m+n+1$, $(m+1)^2$ y los constantes apuntados bajo forma de instrucción, que sirven para iniciar las instrucciones variables de los ciclos del programa.

b) Descripción de la esquema lógica de cálculo (fig. 1)

El 1 bloque atribuye inicialmente $(+1)$ a θ; θ indicará el orden de medida del resultado bajo la forma 10^{-x}.

El 2 bloque atribuye inicialmente O a S; S es un conectador con cuya ayuda se realiza la edición de la solución para la última iteración.

El 3 bloque atribuye inicialmente O a α; α es un conectador que en la última iteración da un salto por encima de los cálculos indicados por los bloques 21—25 que llegan a sobrar.

El 4 bloque atribuye inicialmente O a q; q a contendrá el índice de la columna $(A_0B)^{-1}$ A_0 con el elemento mínimo negativo en primer lugar. Si es que no existe elemento negativo, q queda en 0 y el algoritmo es terminado. q indica el vector que debe ser introducido en la base.

El 5 bloque atribuye O a r. En r será retenido el menor elemento negativo del producto b_1A_0.

El 6 bloque inicia el control de ciclo con vistas a realizar el producto b_1A_0.

El 7 bloque calcula el elemento $L_j = \sum_{k=1}^{m+1} b_{1k}a_{kj}$ de la primera línea de $(A_0B)^{-1}A_0$.

El 8 bloque compara L_j calculado anteriormente con r en que se halla el menor elemento L_k $(k < j)$ encontrado hasta ahora. El máximum de los errores se apunta con ε.

El 9 bloque atribuye el valor corriente de *jaq*.

Bloque 10 — cuando L_j es menor que el contenido de r, a r se le atribuye L_j.

Bloque 11 — Se han comparado todos los elementos L_j $(1 \leqslant j \leqslant m+1)$?

Bloque 12 atribuye $j+1aj$. Se pasa al cálculo de L_{j+1} (bloque 7).

Bloque 13 — ¿q es estrictamente positivo?

Bloque 14 atribuye (-1) a S cuando $q = 0$, o sea en la última iteración, con el fin de editar la solución $(s_1, \ldots s_{m+1})$

Bloque 15 atribuye (-1) a α en la última iteración.

Bloque 16 atribuye inicialmente $(+1)$ a r. En r será retenido $\underset{1 \leqslant i \leqslant m+1}{\text{minimum}} \left(\frac{s_i}{\tau_i} > 0 \right)$

Bloque 17 atribuye inicialmente 1 a i, con el fin de calcular el número p que indica cuál de los vectores de la base debe ser reemplazado.

Bloque 18 atribuby inicialmente 0 a p

Bloque 19 calcula el componente i, $s_i = \sum_{k=1}^{m+1} b_{ik}d_k$ de $(A_0B)^{-1}D_0$.

Bloque 20 — ¿ Es nula la α? Realiza el salto por encima de los bloques 21—25 en la última iteración.

Bloque 21 calcula la componente i, $\tau_i = \sum_{k=1}^{m+1} b_{ik} a_{kq}$, de $(A_0 B)^{-1} a_q$

Bloque 22 —¿ τ_i es estrictamente positivo? ε es un error redondo máximo.

Bloque 23 —¿ s_i/τ_i es la correlación mínima positiva? Esta correlación es positiva porque $s_i \geqslant 0$.

Bloque 24 atribuye a r la nueva correlación mínima.

Bloque 25 atribuye a p el índice i correspondiente a la correlación mínima positiva.

Bloque 26 — ¿ Se han calculado todos los componentes s_i y τ_i?

Bloque 27 atribuye el valor $i + 1$ *ai* y reanuda el calculo del bloque *19*.

Bloque 28 — ¿ $S = 0$? Realiza la entrada en el ciclo de escritura de la solución en la última interación del algoritmo.

Bloque 29 inicialmente atribuye 1 *ai* con el fin de editar los elementos s_i.

Bloque 30 edita s_i.

Bloque 31 [vuelve a trasladar el carro de la máquina de escribir.

Bloque 32.¿ Se ha editado el último s_i?

Bloque 33 atribuye $i + 1$ a i y reanuda las operaciones empezando por el bloque *30* con vistas a editar a s_{i+1}.

Bloque 34.¿ $p = 0$? Indica si el problema tiene o no tiene solución.

Bloque 35. Si $p = 0$, edita a0.

Bloque 36 comenta : no existe solución; se para el proceso de cálculo.

Bloque 37 atribuye inicialmente $(+ 1)$ $a\theta^*$; θ^* indica el orden de medida de los resultados a cada iteración.

Bloque 38 atribuye 0 *aM*. En M será retenido $\max_{1 \leqslant j \leqslant m+1} \{|b_{pj}|\}$

Bloque 39 pone en 1 el índice de columna j de la línea b_p.

Bloque 40¿ $|b_{pj}|$ es el elemento máximo de la línea b_p?

Bloque 41 atribuye $|b_{pj}|$ *aM*

Bloque 42 — ¿ Se han coparado todos los elementos $|b_{pj}|$ de la línea b_p?

Bloque 43 atribuye $j + 1$ *aj* y reanuda las operaciones empezando por el bloque *40*.

Bloque 44¿ La correlación $\frac{M}{\tau_p} < 1$?

Bloque 45 atribuye o, 1. $\theta^* a \theta^*$.

Bloque 46 atribuye 0, 1.M a *M* y reanuda las operaciones empezando por el bloque *44*.

Bloque 47 atribuye $0.\theta^* a \theta$. De este modo θ podrá indicar en la última iteración, el orden de medida del resultado.

Bloque 48 edita $a\theta$.

Bloque 49.¿ $\theta^* = (+ 1)$? Indica el orden de medida que debe tener los elementos de la matriz $(A_0 B_k)^{-1}$ en la iteración k.

Bloque 50 inicia el control del ciclo que realiza la puesta de los elementos de la matriz $(A_0 B_k)^{-1}$ al orden de medida indicado por θ^*.

Bloque 51 atribuye $\theta^* \cdot b_{ik} a b_{ik}$

Bloque 52.¿ Se han multiplicado todos los elementos de la matriz $(A_0 B_k)^{-1}$ con θ^*?

Bloque 53 atribuye $j + 1$ *aj* y el cálculo se reanuda desde el bloque *51*.

Bloque 54 pone el índice de columna j en 1

Bloque 55 atribuye $b_{pj}/\tau_p a b_{pj}$

Bloque 56.¿ Es completa la línea b_p?

Bloque 57 atribuye $j + 1$ *ajy* reanuda el cálculo desde el bloque *55*.

Bloque 58 pone a i en 1. El índice i indica la línea calculada.

Bloque 59.¿ $i = p$? Realiza el salto por encima de la línea b_p que ha sido ya calculada (según los bloques *55—58*)

Bloque 60 indicia el ínice de columna j con vistas a calcular la línea b_i $(i \neq p)$

Bloque 61 calcula el nuevo elemento b'_{ij} según la formula $b'_{ij} = b_{ij} - \tau_i b'_{pj}$

Bloque 62. ¿ La línea i es completa?

Bloque 63 atribuye $j + 1$ *aj* y reanuda el cálculo desde el bloque **61**.

Bloque 64. ¿Se han calculado todas las líneas?

Bloque 65. atribuye $i + 1$ *ai* y reanuda el cálculo desde el bloque 59, o sea pasa a calcular la siguiente línea b_{i+1}.

Bloque 66 da un salto por encima de la línea p

Bloque 67 forma la descripción de la base

Bloque 68. Inicialmente atribuye *la* i con el fin de editar los elementos l_i que forman la descripción de la base.

Bloque 69 edita a l_i

Bloque 70. ¿Se ha editado el último l_i? Si el último l_i ha sido editado, se pasa a la iteración siguiente del algoritmo, reanudando los cálculos desde el bloque *4*.

Bloque 71 atribuye $i + 1$ a i y reanuda las operaciones empezando por el bloque *69*, con vistas a editar a l_{i+1}.

Observacion

1. Despues del bloque *70*, antes de pasar al cálculo de la siguiente iteración, se recomienda introducir un conjunto de bloques que desplacen hacia izquierda los elementos de la matriz $(A_0\ B_k)^{-1}$ para elímínar los ceros insignificantes.

2. En la solución de los problemas de programación lineal con ayuda del calculador electrónico „Economist CIFA-102", cuyas palabras miden 32 cifras binarias, se ha apuntado :

$$
\begin{aligned}
(+1) &= +\ 0{,}999\,999\,999 \\
(-1) &= -\ 0{,}999\,999\,999 \\
0{,}1 &= +\ 0{,}100\,000\,000 \\
\varepsilon &= +\ 0{,}000\,000\,010 \\
1 &= 1 \cdot 2^{-32} \\
m &= \mathrm{m} \cdot 2^{-32} \\
n &= \mathrm{n} \cdot 2^{-32} \\
m + 1 &= (\mathrm{m} + 1) \cdot 2^{-32} \\
n + 1 &= (n + 1) \cdot 2^{-32} \\
m + n + 1 &= (m + n + 1) \cdot 2^{-32} \\
(m + 1)^2 &= (m + 1)^2 \cdot 2^{-32}
\end{aligned}
$$

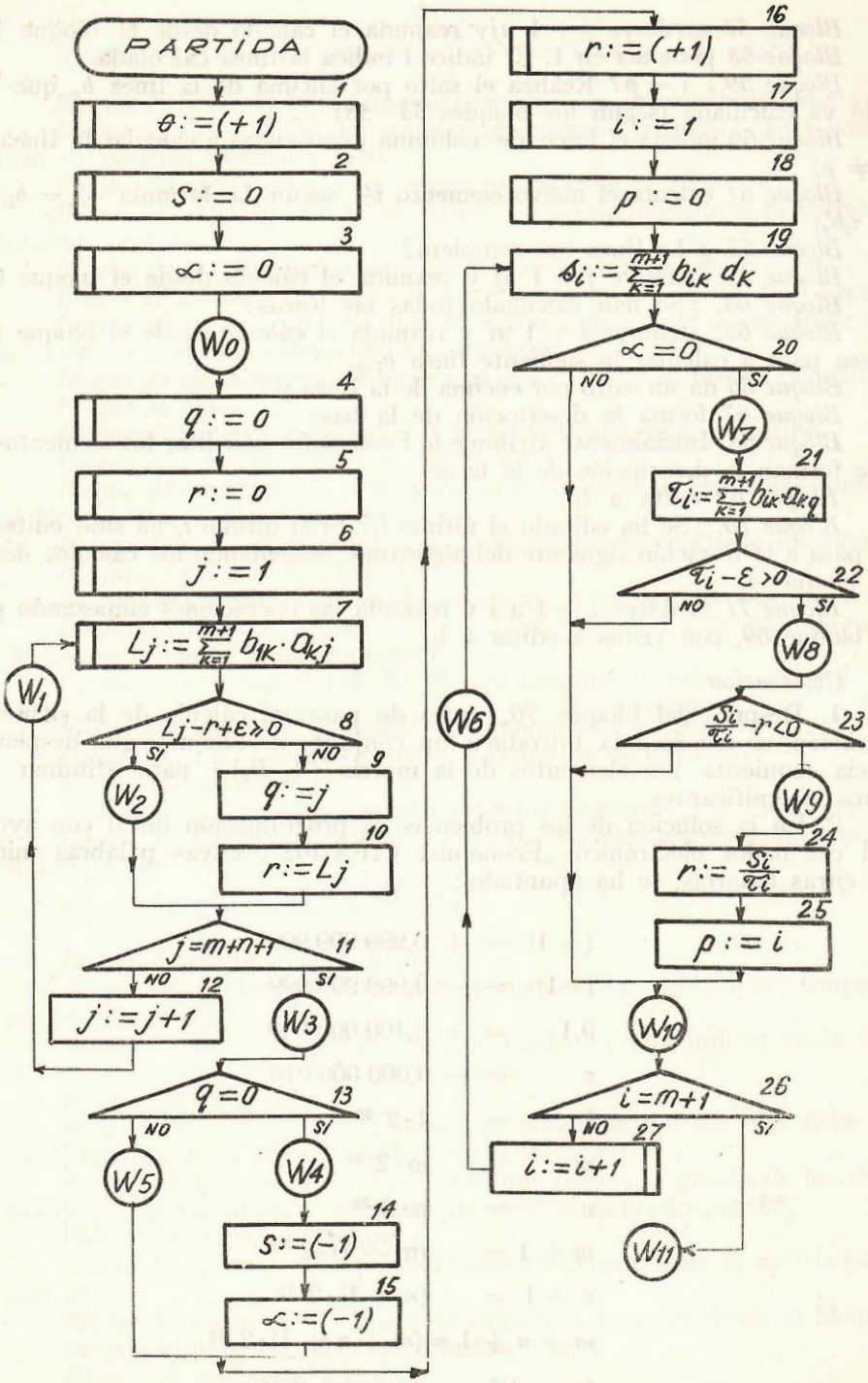
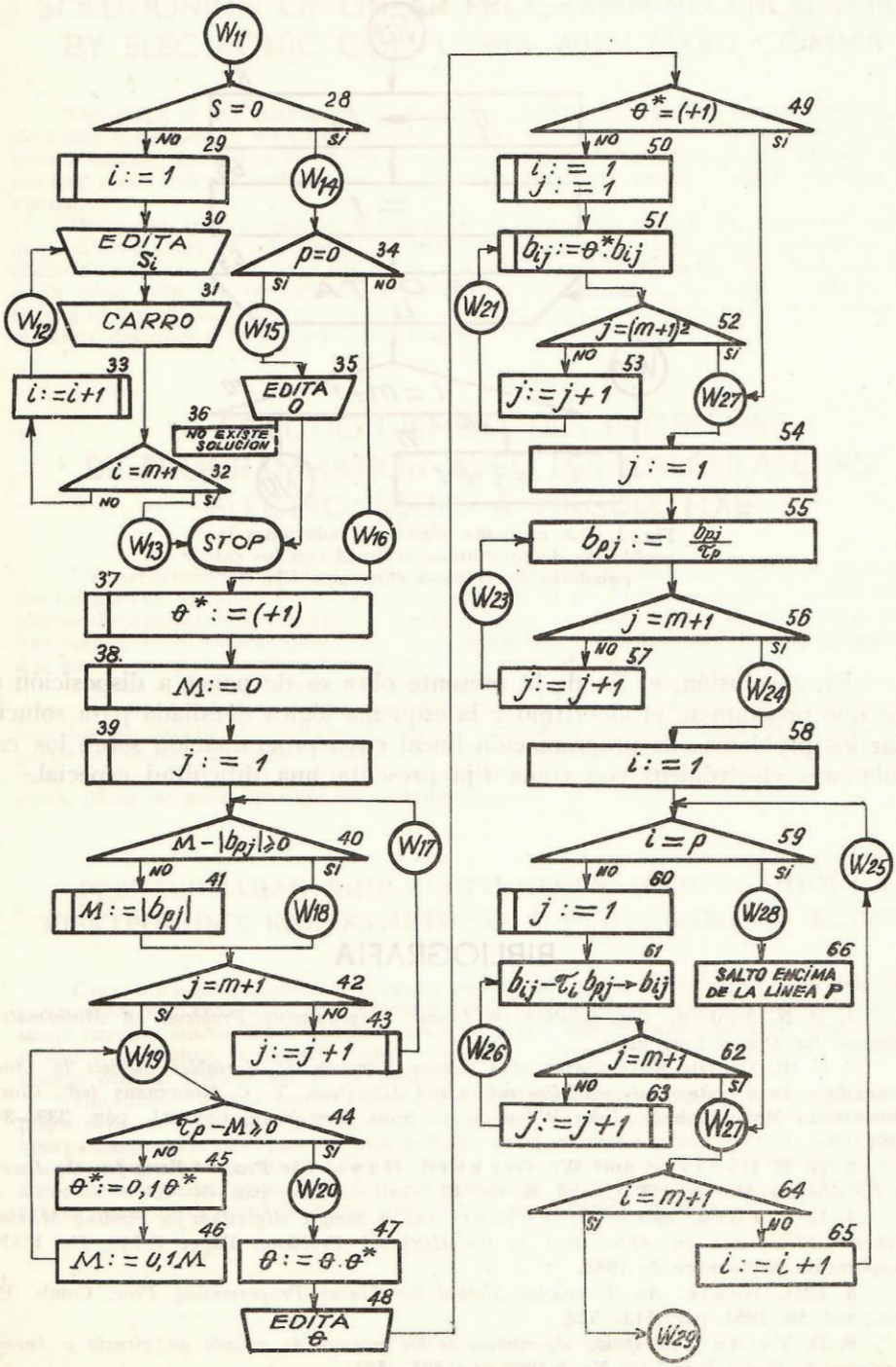


Fig. 1. La esquema lógica de solucionar los problemas de



programación lineal con los calculadores electrónicos con coma fija.

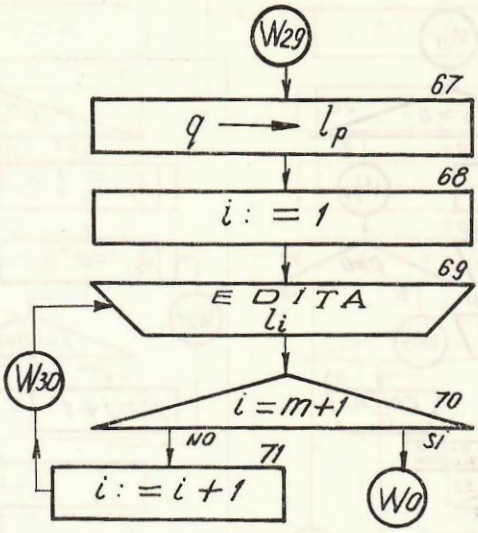


Fig. 1. La esquema lógica de solucionar los problemas de programación lineal con los calculadores electrónicos con coma fija.

En conclusión, el fin de la presente obra es de poner a disposición de los que programan, el algoritmo y la esquema lógica detallada para solucionar los problemas de programación lineal cuya programación sobre los calculadores electrónicos con coma fija presenta una dificultad especial.

BIBLIOGRAFIA

1. D. N. Arden, *The Solution of Linear Programming Problems in Mathematical Methods for Digital Computers.*

2. G. B. Dantzig, *Maximization of a Linear Function of Variables Subject to Linear Inequalities in Activity Analysis of Production and Allocation*, T. C. Koopmans (ed), Cowles Commission Monograph B, John Wiley et co. Sons, New York, Cap. 21, pág. 339—349, 1951

3. G. B. Dantzig and W. Orchard Hays, *The Product Form for the Inverse in the Simplex Method*, MTAC, vol. 8, nr. 46, abril de 1954, pág. 64—67.

4. L. R. Ford, and D. R. Fulkerson, *A Simple Algorithm for Finding Maximal Network Flows and on Application to the Hitchcock Problem.*, Paper P-743, The RAND Corporation, septiembre de 1955.

5. EML Beale, *An Alternative Method for Linear Programming Proc.* Comb. Phil Soc., vol. 50, 1954, pág. 513—523.

6. D. Vaida *Descripción algorítmica de los procesos de cálculo en ciencia e Investigaciones de Física* Tom VIII Nr. 5 1966 pag. 525—584.

SOLUTIONING OF LINEAR PROGRAMMING PROBLEMES BY ELECTRONIC COMPUTERS WITH FIXED COMMA

The stage of our national economy development requires the intensive utilization of electronic computers in working out its problems, amongst which the optimization problems have a leading place. The electronic computers with fixed comma, currently utilized in our country raises delicate problems of programming requiring important efforts for their solutioning.

The paper tries to put at the disposal of programme settlers the algorithm and the detailed flow chart of solutioning of linear programming problems whose programming on electronic computers with fixed comma have a high degree of difficulty. The first part o the paper deals with the theoretical presentation of the algorithm, as well as with the solutioning of a problem which facilitates the understanding of the method and the second part minutely describes a flow chart of calculation.

LE SOLUTIONNEMENT DES PROBLÈMES DE PROGRAMMATION AVEC LES CALCULATEURS ÉLECTRONIQUES À VIRGULE FIXE

L'étape actuelle de dévéloppement de notre économie nationale impose l'utilisation intense des calculateurs électroniques pour le solutionement de ces problèmes, parmi lesquels les problèmes d'optimisation occupent une première place. Les calculateurs électroniques à virgule fixe employés courrament dans notre pays posent des problèmes délicats de programmation qui parfois recquièrent des efforts importants pour leur solutionnement.

L'étude se propose de mettre à la disposition des programmateurs l'algorithme et l'organigrame détaillée de solutionnement des problèmes de programmation linéaire dont la programmation avec les calculateurs électroniques à virgule fixe présente de grandes difficultés. La première partie de l'ouvrage comprend la présentation théorique de l'algorithme, ainsi que le solutionnement d'un problème qui facilite la compréhension de la méthode, tandis que la deuxième partie décrit de facon détaillée une organigrame.

РЕШЕНИЕЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ НА ЭЛЕКТРОННЫХ КАЛЬКУЛЯТОРАХ С НЕПОДВИЖНОЙ ЗАПЯТОЙ

Стадия развития нашего народного хозяйства требует интенсивного использования элекронных калькуляторов при решении задач, среди которых задачи оптимузации занимают главное место. Электронные калькуляторы с неподвижной запятой, приме́няемые у нас с недавнего времени, ставят сложные задачи программирования, требующие иногда особых усилий для их решения.

Настоящая статья ставит перед собой цель предоставить программировщикам алгорифм и подробную логическую схему решения задач линейного программирования, программирование которых на электронных калькуляторах с неподвижной запятой представляет большие трудности. Первая часть статьи охватывает теоретическое изложение алгорифма, а также решение задачи, облегчающей понимание медтода, в во второй части статьи приводится подробное описание долической схемы расчета.