

REVISTA DE

STATISTICA

OCTOMBRIE 1971 **10**

ANUL XX

S U M A R

I. MARCUS: Despre influența productivității muncii și cheltuielilor de timp de muncă asupra sporului producției	3
GH. ȘERBAN, T. MUNTEANU: Variațiile sezoniere și rezervele de producție în activitatea de construcții-montaj	26
E. FERENȚ: Aplicarea metodei simulării la procesele de aprovizionare — stocare	36
P. DUMITRU: Sugestii privind o interpretare cibernetică a procesului de cointerensare materială a oamenilor muncii	41
D. V. ILIESCU, V. GH. VODĂ: Studiul caracteristicilor operative pentru planuri de control statistic de recepție în cazul controlului calității medii	49

PRELUCRAREA ELECTRONICĂ A INFORMAȚIEI

C. DARIE, M. FERNEA, L. KOVACS: Experiențe pseudoaleatoare efectuate pe un calculator electronic	56
--	----

ÎN ÎNTÎMPINAREA ANULUI MONDIAL AL POPULAȚIEI — 1974

VL. TREBICI: Introducere la o încercare de uniformizare a termenilor demografici (I)	63
--	----

CONSULTAȚII

GH. MIHOC, V. URSEANU: Legi de repartiție ale variabilelor aleatoare. Teorema limită centrală. Repartiții Marcoviene	65
--	----

STATISTICA TERITORIALĂ

I. BĂLAN: Dezvoltarea demografică și economică a municipiului Ploiești	76
M. GÎRLĂ: Cercelare statistică privind creșterea animalelor în cooperativele agricole de producție din județul Timiș, 1966—1970	84
E. ROVENȚA: Industria cărbunelui în Valea Jiului, în trecut și astăzi	89

STATISTICA ÎN ÎNTREPRINDERI

E. NEGREA: Durata de recuperare a investițiilor în extracția țițeiului	93
--	----

DIN STATISTICA INTERNAȚIONALĂ

SIMONA PANTAZI: O aplicație a energiei informaționale ca măsură a nivelului de trai în Japonia	96
--	----

NOTE-RECENZII

B. STĂNESCU: Tineretul — obiect al cercetării științifice	100
I. NEGURĂ: Un manual de buzunar indispensabil cadrelor din agricultură: Indicatorii folosiți în activitatea întreprinderilor agricole socialiste	102

DIN PRACTICA MUNCII DE STATISTICĂ

Întrebări — Răspunsuri	103
----------------------------------	-----

INFORMAȚII STATISTICE

Probleme actuale ale statisticii de stat în discuția Colegiului Direcției Centrale de Statistică	105
--	-----

DIN CUPRINSUL REVISTELOR DE SPECIALITATE

COLEGIUL DE REDACȚIE

E. BARAT, M. BIJI, membru corespondent al Academiei Republicii Socialiste România,
A. CSILLAG, I. DUMITRESCU, D. D. HAȘIGAN, D. MIHALACHE, Acad. GH.
MIHOC, R. MOLDOVAN, membru corespondent al Academiei Republicii Socialiste
România, B. STĂNESCU (redactor șef), P. ONICĂ,
I. PĂCURARU, L. RĂDULESCU, VL. TREBICI

REVISTA DE

STATISTICĂ

ORGAN AL DIRECȚIEI CENTRALE DE STATISTICĂ
DE PE LÎNGĂ CONSILIUL DE MINISTRI AL R. S. R.
ȘI AL SOCIETĂȚII DE ȘTIINȚE ECONOMICE DIN R. S. R.

10

EXPERIENȚE PSEUDOALEATOARE EFECTUATE PE UN CALCULATOR ELECTRONIC

C. DARIE M. FERNEA A. KOVÁCS

Institutul de calcul, Cluj

Aplicațiile șirurilor pseudoaleatoare sînt multiple. Ele pot, de exemplu, să modeleze șirurile aleatoare, care — după cum se știe — apar în experiențe probabilistice sau ca rezultate ale unor fenomene naturale economice sau sociale.

O altă aplicație a șirurilor pseudoaleatoare se face în metoda Monte Carlo, al cărei obiect este rezolvarea problemelor de analiza numerică prin exprimarea rezultatelor ca funcții de variabile aleatoare. Realizarea acestor variabile aleatoare se poate face direct prin scheme probabilistice sau indirect prin aproximarea nealeatoare a variabilelor aleatoare, deci prin numere pseudoaleatoare. Astfel, șirurile normale de cifre sînt șiruri pseudoaleatoare pentru schema lui Bernoulli, iar șirurile de distribuție uniforme sînt șiruri pseudoaleatoare pentru schema distribuției probabilistice uniforme. Prin transformări efectuate asupra acestor șiruri pseudoaleatoare se pot obține aproximații ale diferitelor distribuții de probabilitate.

Studiul șirurilor pseudoaleatoare se poate aborda structural [2] sau din punct de vedere al generării acestora. Un colectiv de cercetători de la Institutul de calcul din Cluj a reușit să pună la punct programe pentru calculatorul DACICC-1 de generare a unor șiruri pseudoaleatoare (metoda pătratelor medii a lui J. von Neumann, metoda congruențelor), de analiză a unor șiruri normale de cifre. De asemenea, s-au modelat unele probleme nenumerice cum ar fi: mersul aleator în plan și obținerea invelitorii convexe a unei mulțimi finite de puncte din pătratul unitate (problemă legată de metoda țințelor prin care se pot evalua ariile unor configurații plane neregulate).

Deși experiențele s-au făcut pe calculatorul DACICC-1 schemele logice prezentate mai jos sînt universale.¹

Modelarea pe calculator a generării numerelor pseudoaleatoare și aplicațiile lor

Un procedeu de generare a numerelor pseudoaleatoare a fost propus de J. von Neumann [6], [7].

Definiție: Dacă $g \geq 2$ este un număr natural, atunci vom nota cu

$$\overline{a_{m-1} a_{m-2} \dots a_1 a_0}^{(g)}$$

numărul natural

$$a_0 + a_1 g + \dots + a_{m-1} g^{m-1}$$

unde $m = 2k$ (număr par).

Dacă din pătratul

$$\overline{(a_{m-1} a_{m-2} \dots a_1 a_0)^{(g)}}^2 = \overline{b'_{2m} \dots b'_1 b'_0}^{(g)} \quad (b'_j \in \overline{0, g-1})$$

ștergem primele și ultimele $\frac{m}{2} = k$ cifre, obținem numărul notat cu

$$N(\overline{a_{m-1} \dots a_1 a_0}^{(g)}) = \overline{b'_{k+m-1} \dots b'_{k+1} b'_k}^{(g)} = \overline{a'_{n-1} \dots a'_1 a'_0}^{(g)}$$

¹ Unele idei ale problemelor puse au fost sugerate de dr. S. N. Bertî personal precum și prin lucrarea [2].

În anumite condiții [6] șirul finit de cifre în baza g generat de $\overline{a_{m-1} \dots a_0}^{(g)}$:

$$a_{m-1}, \dots, a_0, \overline{a'_{m-1}}^{(1)}, \dots, \overline{a'_0}^{(1)}, \dots, \overline{a'_{m-1}}^{(l)}, \dots, \overline{a'_0}^{(l)} \quad (1)$$

se consideră ca fiind pseudoaleator, acest șir fiind o aproximație al unui șir normal de cifre [2, 3], în sensul că orice combinație finită de lungime n de cifre apare în n -întinderea lui (1) aproximativ de același număr de ori $\left(\frac{m}{n} \text{ este un număr mare}\right)$.

Experiența arată că în general șirul (1) satisface această proprietate, pentru care motiv generarea acestui șir se folosește în metoda Monte Carlo.

Un număr $\overline{a_{m-1} \dots a_0}^{(g)}$ are perioada l dacă numerele

$$\overline{a_{m-1} \dots a_0}^{(g)}, \dots, \overline{a_{m-1} \dots a_0}^{(l)} \quad (g)$$

sînt diferite și $N(\overline{a_{m-1} \dots a_0}^{(g)})$ este egal cu unul din aceste numere. Această definiție riguroasă a perioadei se cere completată cu o altă definiție cu un caracter stohastic: un număr $\overline{a_{m-1} \dots a_0}^{(g)}$ are perioada pseudoaleatoare l dacă șirul (1) este un șir pseudoaleator în sensul aproximației normale de mai sus, iar șirul

$$a_{m-1}, \dots, a_0, \dots, \overline{a'_{m-1}}^{(s)}, \dots, \overline{a'_0}^{(s)}$$

pentru $s \geq l$ nu este pseudoaleator.

Pornind de la ipoteza că perioada pseudoaleatoare este puțin inferioară perioadei, se pune problema generării unor numere de perioadă mai mare decît un număr dat. În acest fel pentru a obține un număr de perioadă pseudoaleatoare p generăm un număr de perioadă p' și ($p' > p$).

Schema logică, după care s-a efectuat generarea de șiruri de numere pseudoaleatoare de tip Neumann, este dată în fig. 1 (v. anexa 1).

Un alt procedeu de generare a numerelor pseudoaleatoare, propus de Lehmer, este acela al congruențelor [1]. Metoda constă în formarea, printr-un procedeu inductiv a șirului de numere $\{x_n\}$ plecînd de la o valoare inițială arbitrară x_0 . Termenul general al șirului se obține cu ajutorul relației de recurență

$$x_{n+1} = kx_n \pmod{M},$$

unde k, n sînt numere naturale.

Ținînd seama de specificul mașinilor de calcul e indicat să se ia k de forma 3^p sau 5^q , p și q fiind cele mai mari numere prime pentru care 3^p , respectiv 5^q , nu depășește lungimea registrului iar M se ia 2^x , x fiind numărul biților dintr-un cuvînt al calculatorului respectiv.

La mașinile de calcul care lucrează cu virgulă fixă, produsul kx_n se efectuează în general simplu, datorită existenței codurilor de tipuri produs de dublă lungime. Astfel se ia în considerare numai partea ce apare în registrul auxiliar, deci operația $\pmod{M}$ se execută automat.

Sînt și mai avantajoase alegerile lui $k = 2^r + 1$, înmulțirea reducîndu-se la o simplă deplasare a registrului la stînga sau la dreapta.

În cazul mașinii electronice de calcul DACICC-1 s-a ales $k = 5^{13}$, $M = 2^{33}$ și $x_0 = 1$; cu aceste valori pe baza formulei de recurență, se obține următorul șir:

$$1106047162546200632271610575452552131422666167504515405 \dots$$

numerele fiind date în sistemul de numerație octal. Schema logică corespunzătoare este prezentată în fig. 2 (v. anexa 1) unde notațiile sînt cele folosite în schema anterioară.

La mașina de calcul DACICC-1 s-au mai efectuat calcule pentru obținere de numere pseudoaleatoare cu ajutorul șirurilor de numere ale lui Fibonacci. Formula de recurență în acest caz este

$$x_{n+2} = \frac{1}{M} F_{n+2}$$

unde

$$F_{n+2} = a F_{n+1} + b F_n \pmod{M}$$

și

$$F_0 = 0, F_1 = 1.$$

Evident, această metodă nu diferă esențial de cea expusă anterior.

O problemă importantă din punct de vedere statistic este studiul periodicității șirurilor de numere obținute. În acest sens se caută partea neperiodică a șirurilor și apoi lungimea perioadei.

Pentru a fi respectat caracterul pseudoaleator trebuie păstrată și folosită în diferitele probleme numai partea neperiodică a unui șir obținut. Fiindcă, de obicei, calculele în care se folosesc numerele pseudoaleatoare cer un volum mare de numere, ne interesează acele șiruri care au partea neperiodică mare. În acest sens s-a elaborat o schemă logică și un program pentru calculatorul DACICC-1 cu ajutorul cărora se obțin șiruri cu o parte neperiodică mai mare decât un număr dat.

În cazul algoritmului J. von Neumann, pentru obținerea de șiruri cu parte neperiodică mai mare decât un număr dat, schema logică este dată în fig. 3 (v. anexa 1).

Schema logică e general valabilă pentru că „blocul algoritmului lui Neumann”, poate fi înlocuit cu blocul oricărui alt algoritm de generare a numerelor pseudoaleatoare.

Ca rezultat se obține tipărirea limitelor inferioare și superioare între care se caută șirul de numere cu parte neperiodică mai mare decât un număr dat, precum și șirurile pentru care primul termen este x_0 , iar partea neperiodică este mai mare decât numărul cerut.

Ca exemplu pentru algoritmul lui Neumann de șiruri cu parte neperiodică mai mare decât 60 sint șirurile cu termeni inițiali 1647, 2057. Șirurile corespunzătoare sint:

164771267798808841572807873631760869755101760309095491018282

591598724563820938670233054229376259175006253906256859463549

595445012590708114059740867627294474016602750756571566127185

205723123453923222982808884828712426885439334684939842243941

531423858823619097194288871694624690959919656644080860528614

785671673658380950848470740989327806933616085856292756731829

Șirurile de numere pseudoaleatoare obținute prin unul din algoritmele date au fost utilizate la simularea mersului aleator pe o axă și în plan, care are aplicații la rezolvarea problemelor la limită a ecuațiilor diferențiale și cu derivate parțiale.

Algoritmul pentru mersul aleator într-un dreptunghi din plan s-a conceput în felul următor: mobilul fiind într-un punct dat, probabilitățile de deplasare în cele patru direcții și ale rămînerii pe loc sint p, q, r, s, t , unde suma lor este 1.

Într-o celulă este introdus un număr aleator. Deoarece nu avem nevoie de un număr atât de mare de cifre, împărțim celula în patru părți, conținutul fiecăreia fiind considerat ca un număr pseudoaleator de sine stătător. Prin aceasta s-a reușit să se mărească capacitatea mașinii de patru ori.

În cazul mașinii de calcul DACICC-1, lungimea unei celule este de 36 biți, deci a patra parte este de 9 biți, ceea ce corespunde la trei cifre octale. Cel mai mic octal de trei cifre este 000, iar cel mai mare 777, în total o mie de numere octale. Această mie de numere o împărțim în grupe proporțional cu probabilitățile de deplasare date anterior adică fiecărei grupe îi corespunde una din cele patru plus 1 direcții de deplasare. Pentru a decide direcția de deplasare a mobilului, trebuie să verificăm căruia din cele 5 grupe îi aparține numărul pseudoaleator pe care îl luăm în considerare.

În felul acesta s-a modelat legea de deplasare a mobilului. În drumul pe care îl parcurge, mobilul poate ajunge pe oricare din laturile dreptunghiului, care joacă rolul unor bariere absorbante. În momentul atingerii frontierei, se reia studiul mișcării mobilului din punctul inițial. Mișcarea mobilului poate începe din orice punct al dreptunghiului. În felul acesta, fiecărui punct îi corespunde o anumită frecvență de atingere a laturilor și în concluzie se poate studia media de atingere a fiecărei laturi în raport cu toate punctele dreptunghiului și șirul numerelor pseudoaleatoare considerate.

Cu mici modificări algoritmul este valabil și pentru domenii mai generale decât dreptunghiul și pentru spații cu mai multe dimensiuni.

De exemplu:

$$p = \frac{1}{4} \quad q = \frac{1}{8} \quad r = \frac{3}{8} \quad s = \frac{1}{16} \quad t = \frac{3}{16}$$

punctul inițial este 0 de coordonate $x_0 = y_0 = 5$ și domeniul este un pătrat definit prin $x = 0, y = 0, x = 12, y = 12$ (în octal).

În acest caz corespondențele între direcția de deplasare și grupele de numere sint:

100 — 200	— deplasare la dreapta
201 — 300	— deplasare în sus
301 — 600	— deplasare la stînga
601 — 640	— deplasare în jos
641 — 1000	— rămîne pe loc.

Șirul numerelor pseudoaleatoare fiind 106515110711, 611110603311, ..., mobilul se deplasează astfel: (6,5); (5,5); (6,5); (6,4); (7,4); (7,3); ... iar peste 31 de pași ajunge la frontieră. Schema logică este prezentată în fig. 4 (v. anexa 1).

În lucrarea [2] se face un studiu al șirurilor normale de cifre care au aplicații în metoda Monte Carlo; evaluînd exact funcția L definită prin

$$L(a, r, g) = \sum_{j=1}^{r-k+1} j \cdot \mu S(a, r, j) - \frac{r}{g^k}$$

se stabilește formula de recurență pentru unele probleme legate de aceste șiruri. Pentru a putea face acest studiu, se introduce numărul $t(a, b)$ după cum urmează:

Fie g, k, r, n numere naturale, $g \geq 2$ baza sistemului de numerație $1 \leq k \leq r, n \geq 0$, $0, g-1$ mulțimea $\{0, 1, \dots, g-1\}$ și $0, g-1^k$ mulțimea sistemelor de formă $(a_1, \dots, a_k)$ unde $a_i \in 0, g-1, b = (b_1, \dots, b_r) \in 0, g-1^r, b^{(j)} = (b_j, \dots, b_{j+k-1})$. Fie

$$t(a, b) = \sum_{j=1}^{r-k+1} I(a, b^{(j)})$$

unde $I(a, b^{(j)}) = \begin{cases} 1 & \text{dacă } a = b^{(j)}, \\ 0 & \text{altfel,} \end{cases}$

Astfel $t(a, b)$ este numărul acelor sisteme din $(b_1, \dots, b_k), (b_2, \dots, b_{k+1}), \dots, (b_{r-k+1}, \dots, b_r)$ care sint egale cu a . Pentru determinarea numărului $t(a, b)$ s-a elaborat un program, a cărui schemă logică este prezentată în fig. 5 a.b.c.d (v. anexa 2). S-a ales $g = 2, r$ este arbitrar iar k satisface condiția $k \leq 36$. Fixînd o mulțime $b \in 0, g-1^r$ și un k , programul determină pentru orice combinație de lungime k a lui a numerele $t(a, b)$. Menționăm că șirul b de lungime r poate fi generat cu oricare din metodele de generare a numerelor pseudoaleatoare date mai sus. Cu ajutorul acestui program se poate face un studiu statistic al mulțimii $S(a, r, n)$, unde $S(a, r, n)$ este mulțimea acelor elemente $b \in 0, g-1^r$ pentru care $t(a, b) = n$.

De exemplu, dacă $r = 720$ șirul fiind $b =$

65150756615566434565123324377570261603334250224555375164116

741664216351136733117266453070060765224717105342672400212311

031672573000674162752144506152455242433346335710041045427775

776773752627256664472573137525162766042521340317751204504756

pentru $k = 2$ obținem

00—569
01—630
10—529
11—529

Numerele din șirul b sint transformate automat în baza 2, conform tabelului următor:

0	000	5	101
1	001	6	110
2	010	7	111.
3	011		
4	100		

Algoritm pentru determinarea învelitorii convexe a unei mulțimi de puncte

Se pune problema de a se determina un algoritm pentru determinarea învelitorii convexe a unei mulțimi de puncte date, cu ajutorul unui calculator.

Fie M această mulțime $M \subset [0, 1] \times [0, 1]$. Notăm cu $\tilde{T}_{ij}$ triunghiul format de punctele X_0, X_i, X_j unde punctul X_0 este un punct cu abscisa minimă, $X_0 \in M, X_i \in M, X_j \in M$. Triunghiul $\tilde{T}_{ij}$ se consideră ca fiind interiorul și frontiera triunghiului. Notăm $T_{ij} = M \cap \tilde{T}_{ij}$. Construim mulțimea:

$$C = \bigcap_{i=1}^{n-1} \bigcap_{j=1}^n \{M - \{T_{ij} - \{X_0, X_i, X_j\}\}\}$$

Teoremă. Mulțimea C este învelitoarea convexă a mulțimii M .

Demonstrație. Vom nota cu τ învelitoarea convexă a mulțimii M . Vom demonstra teorema arătând dubla incluziune $C \subset \tau$ și $\tau \subset C$.

Mulțimea C prin construcție este mulțimea vîrfurilor tuturor triunghiurilor care se pot construi cu puncte din mulțimea M pentru care nu există un alt triunghi care să le conțină.

Punctele învelitorii convexe sînt vîrfurile unui poligon convex cu n laturi. Aceasta se poate descompune în triunghi plecînd de la punctul X_0 și unindu-le cu celelalte vîrfuri ale poligonului. Așadar punctele mulțimii τ sînt vîrfurile unor triunghiuri pentru care nu există alte triunghiuri care să le conțină. Rezultă $\tau \subset C$ (fig. a). Să demonstrăm că și $C \subset \tau$.

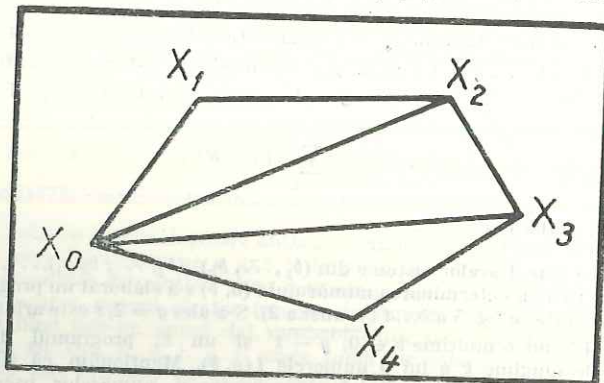


Fig. a

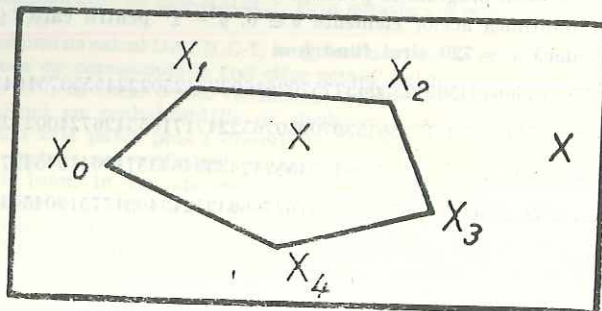


Fig. b

Fie $X \in C, X \in \tau$ căi altfel X este sau în interiorul sau în exteriorul poligonului convex. Dacă se află în interiorul poligonului convex atunci există cel puțin un triunghi care-l conține, rezultă deci $X \notin C$ ceea ce este o contradicție. Dacă X este în exterior, atunci τ nu este învelitoarea convexă (fig. b).

Pe baza acestei teoreme se poate construi un algoritm de determinare a învelitorii convexe cu ajutorul unei mașini electronice de calcul.

Descrierea algoritmului: Fie M o mulțime de puncte pe care le notăm cu $X_1, X_2, \dots, X_n$. Notăm cu X_0 punctul care se caracterizează prin abscisa minimă. Acest punct aparține

învelitorii convexe. Luăm două puncte oarecare X_i și X_j . Cele trei puncte X_0, X_i și X_j formează un triunghi. Vom elimina toate punctele X_s ($s = 1, \dots, n$) care aparțin triunghiului X_0, X_i, X_j . Un punct aparține triunghiului dacă:

$$[\text{sign } D_{X_0 X_j}(X_i) = \text{sign } D_{X_0 X_j}(X_s)] \& [\text{sign } D_{X_0 X_i}(X_j) = \text{sign } D_{X_0 X_i}(X_s)] \& [\text{sign } D_{X_i X_j}(X_0) = \text{sign } D_{X_i X_j}(X_s)]$$

este adevărată,

unde $D_{X_i X_k}$ este ecuația dreptei ce trece prin punctele X_i și X_k , iar $D_{X_i X_k}(X_p)$ este puterea punctului X_p față de dreapta $D_{X_i X_k}$. Dacă $D_{X_i X_k}(X_s) = 0$ atunci punctul se elimină dacă:

$$x_k < x_s < x_i$$

Aceste calcule se fac pentru $s = 1, 2, \dots, n; j = 2, 3, \dots, n; i = 1, 2, \dots, n - 1$.

Cînd $s = n$, se reia calculul de la $s = 1$ pentru triunghiul X_0, X_{j+1}, X_i pînă cînd $j = n$. Cînd $j = n$, se reia calculul de la $s = 1, j = 1$ pentru triunghiul X_0, X_j, X_{i+1} pînă la $i = n - 1$. Cînd $i = n - 1$ punctele care nu au fost eliminate pînă acum formează învelitoarea convexă. Schema logică a acestui algoritm este prezentată în figurile 6, 7, 8, 9 (anexa 3 și 4).

Organizarea memoriei: Coordonatele punctelor formează două masive, unul pentru abscise, altul pentru ordinate. Pentru calculatorul DACICC-1 programul ocupă memoria de la 0 la 276. Datele se introduc începînd cu adresa 301 pentru abscise și cu adresa 1141 pentru ordinate, în total se pot analiza 500 de puncte. În celula 234 se înscrie numărul n (numărul punctelor de analizat). Programul se pornește de la adresa zero.

Evaluarea numărului de operații: Pentru o valoare l a lui s se fac trei comparații. Pentru o valoare a lui i se fac $3 \cdot n^2$ comparații. În total $3 \cdot n^2(n - 1)$ comparații. Pentru o comparație se fac 10 scăderi și 5 înmulțiri.

În total: $30 \cdot n^2(n - 1)$ adunări și $15 \cdot n^2(n - 1)$ înmulțiri.

Observație: la calculatorul DACICC-1, pentru $n = 50$, rezultă 14 minute de lucru; această durată se micșorează însă în fapt, deoarece pe parcursul desfășurării programului n scade.

Programul în limbajul ALGOL-60 este următorul:

Procedure convex

value n ; **integer** n ; **array** X, Y ;

begin **integer** i, j, s, r ;

begin **own**:

$x[0] := x[1]$;

begin **for** $i = 1$ **step** 1 **until** n **do**

if $x[0] > x[i]$ **then** E

$x[0] := x[i]$;

$j := i$;

end

begin **integer** r ;

$y[0] := y[j]$;

for $r = j$ **step** 1 **until** n **do**

$y[r] := y[r + 1]$;

end

end

begin **for** $i = 1$ **step** 1 **until** n **do**

begin **for** $j = 2$ **step** 1 **until** n **do**

begin **for** $s = 1$ **step** 1 **until** n **do**

if $((x[s] - x[0])(y[i] - y[0]) - (y[s] - y[0])(x[i] - x[0])) > 0$

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

then E1 **else** E3

```

E4:      y[r] := y[r + 1];
        n := n - 1;
        end

```

```

        end
    end
begin for i = 1 step 1 until n do
    print x[i];
    print y[i];
end

```

Exemplu: Fie M mulțimea punctelor:

$X_1 (0,2; 0,25); X_2 (0,2; 0,3); X_3 (0,2; 0,4); X_4 (0,2; 0,5);$
 $X_5 (0,2; 0,6); X_6 (0,2; 0,7); X_7 (0,1; 0,4); X_8 (0,3; 0,7);$
 $X_9 (0,3; 0,6); X_{10} (0,3; 0,5); X_{11} (0,3; 0,4); X_{12} (0,3; 0,3);$
 $X_{13} (0,4; 0,2); X_{14} (0,4; 0,3); X_{15} (0,4; 0,4); X_{16} (0,4; 0,5);$
 $X_{17} (0,4; 0,6); X_{18} (0,4; 0,7); X_{19} (0,4; 0,8); X_{20} (0,5; 0,9);$
 $X_{21} (0,5; 0,8); X_{22} (0,5; 0,7); X_{23} (0,5; 0,6); X_{24} (0,5; 0,5);$
 $X_{25} (0,5; 0,4); X_{26} (0,5; 0,3); X_{27} (0,6; 0,3); X_{28} (0,6; 0,4);$
 $X_{29} (0,6; 0,5); X_{30} (0,6; 0,6); X_{31} (0,6; 0,7); X_{32} (0,6; 0,8);$
 $X_{33} (0,7; 0,8); X_{34} (0,7; 0,7); X_{35} (0,7; 0,6); X_{36} (0,7; 0,5);$
 $X_{37} (0,7; 0,4); X_{38} (0,7; 0,3); X_{39} (0,8; 0,5); X_{40} (0,8; 0,6);$
 $X_{41} (0,8; 0,7); X_{42} (0,8; 0,8); X_{43} (0,9; 0,3);$

Învelitoarea convexă din exemplul dat (fig. 10) este formată din punctele:

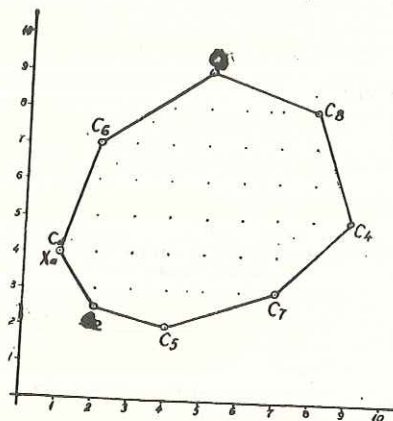


Fig. 10

$C_1 (0,1; 0,4); C_2 (0,2; 0,25); C_3 (0,5; 0,9); C_4 (0,9; 0,5);$
 $C_5 (0,4; 0,2); C_6 (0,2; 0,7); C_7 (0,7; 0,3); C_8 (0,8; 0,8)$

Bibliografie

- [1] D. H. Lehmer, *Mathematical methods in large-scale computing units*, Proc. sympos. on large-scale digital calcul. machinery, 141–146, Harvard Univ. Press, 1949.
- [2] S. N. Berti, *Probleme legate de studiul șirurilor normale de cifre*, St. cerc. mat., 17 (1965), 1429–1441.
- [3] E. Borel, *Leçons sur la théorie des fonctions*, Paris, 1914.
- [4] B. Jansson, *Random number generators*, Stockholm, 1966.
- [5] N. Metropolis, *Phase shifts-middle squares - wave equations*, Symposium on Monte Carlo Methods, March 16–17, 1954.
- [6] J. von Neumann, *Various techniques used in connection with random digits*, Collected Works, vol. 5, Oxford, 1963.
- [7] C. Darie, *Generarea prin metoda lui J. von Neumann a șirurilor de numere pseudoaleatoare*, St. cerc. mat., 21 (1969), 1279–1285.

METODA LUI J. VON NEUMANN

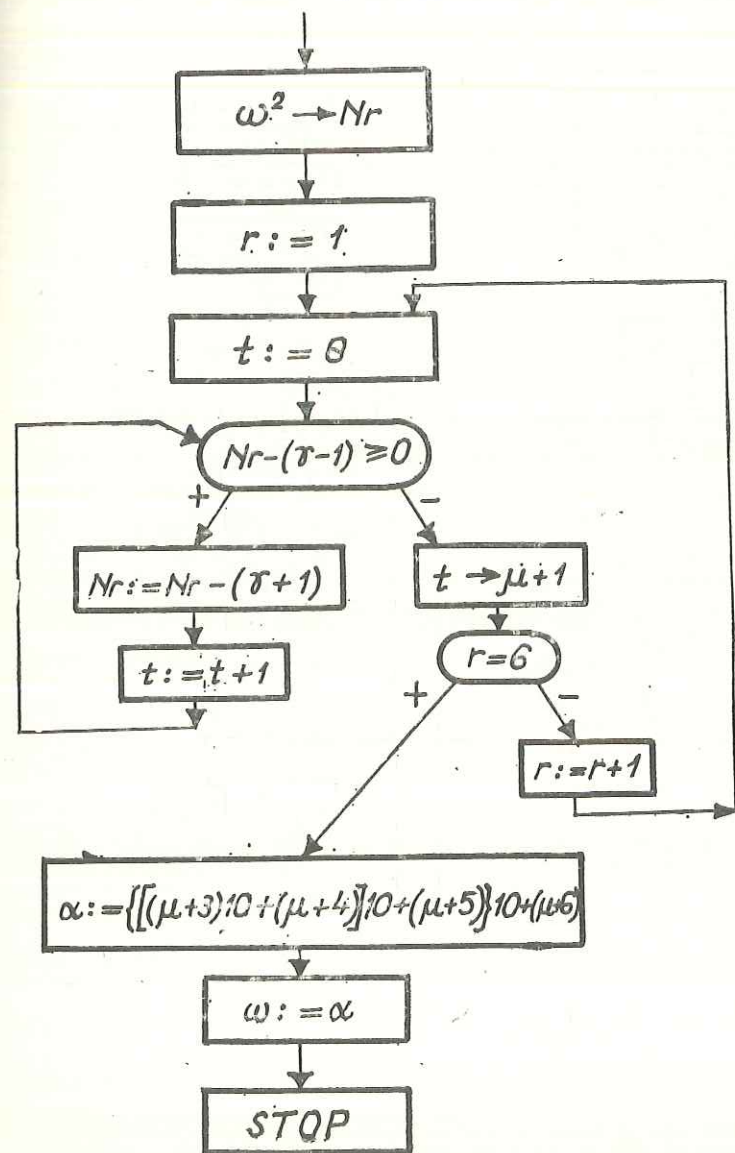


Fig. 1

METODA CONGRUENȚELOR

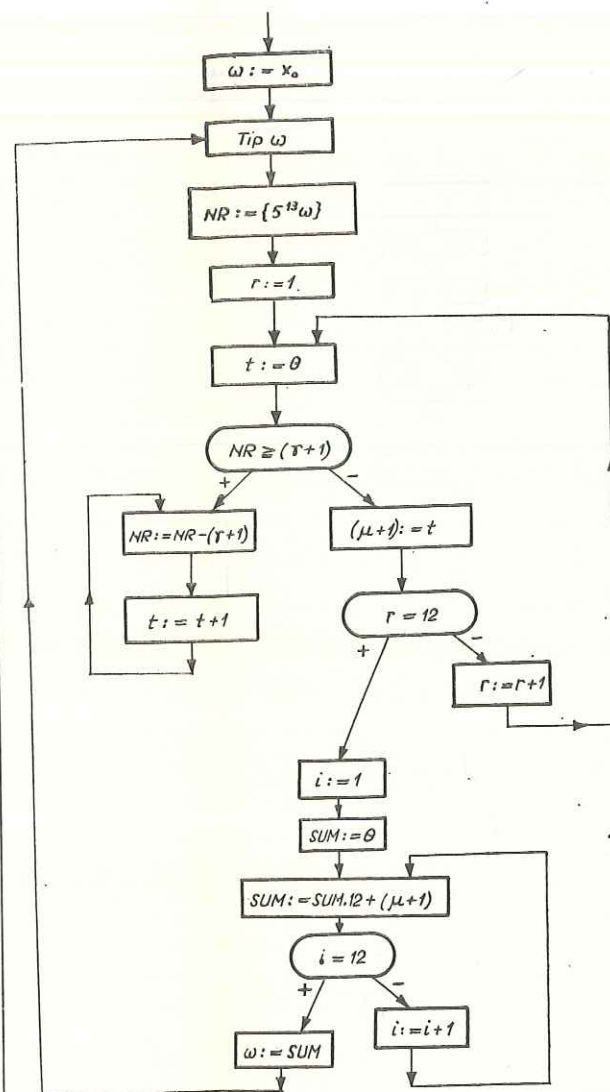


Fig. 2

STUDIUL PERIOADEI

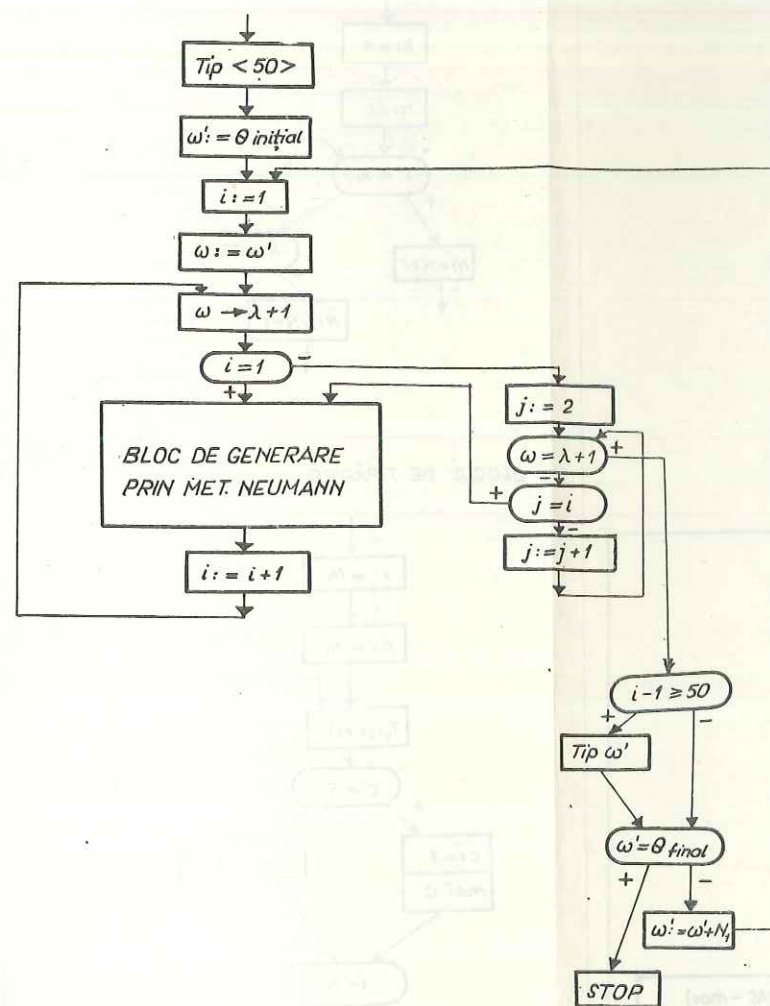


Fig. 3

MERSUL ALEATOR

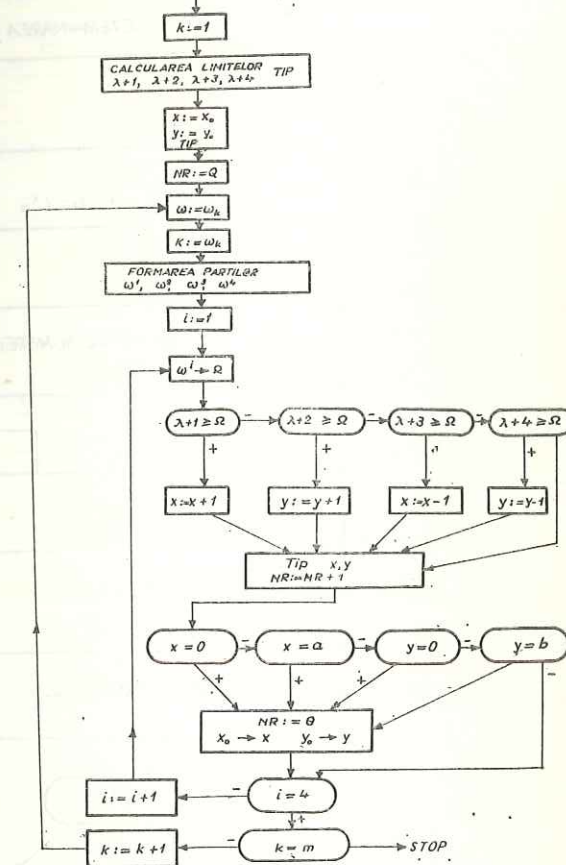


Fig. 4

PROBLEMA INTINDERILOR
PROGRAMUL DE CONDUCERE AL CALCULULUI

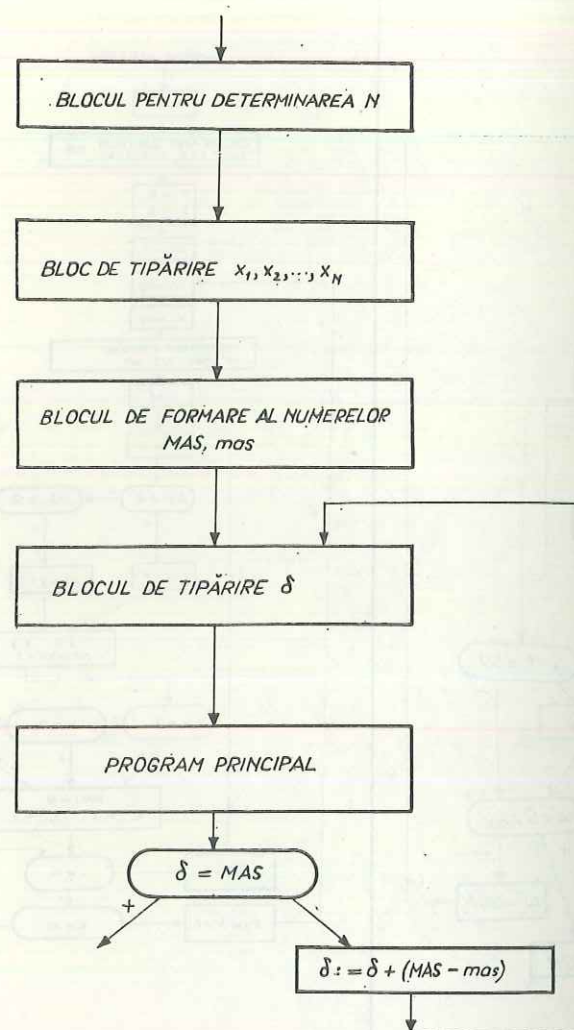
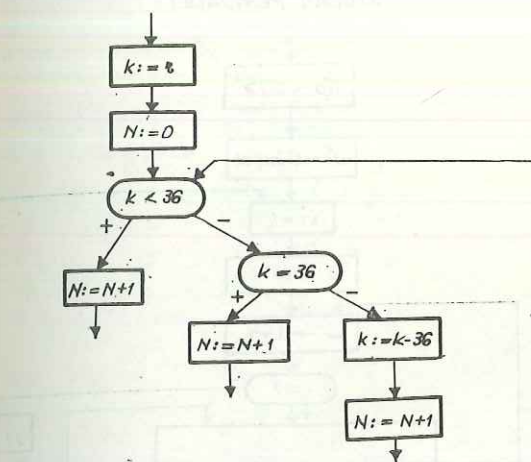


Fig. 5a

BLOCUL PENTRU DETERMINAREA LUI N
(N= numărul celulelor ocupate)



BLOCUL DE TIPĂRIRE γ

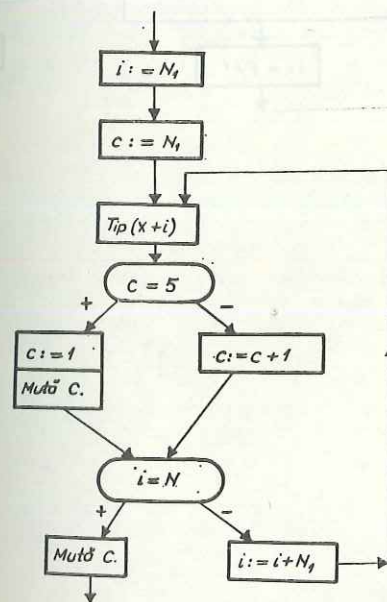
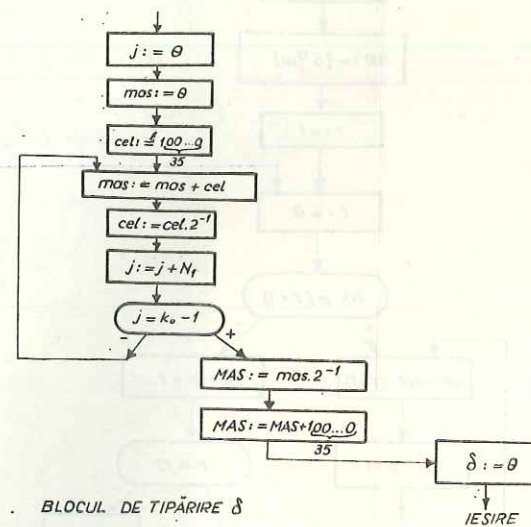


Fig. 5b

BLOCUL DE FORMARE AL NUMERELOR
MAS și mas



BLOCUL DE TIPĂRIRE δ

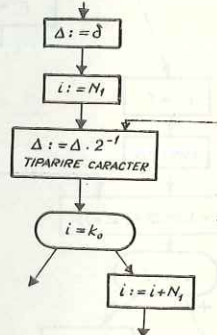


Fig. 5c

PROGRAM PRINCIPAL

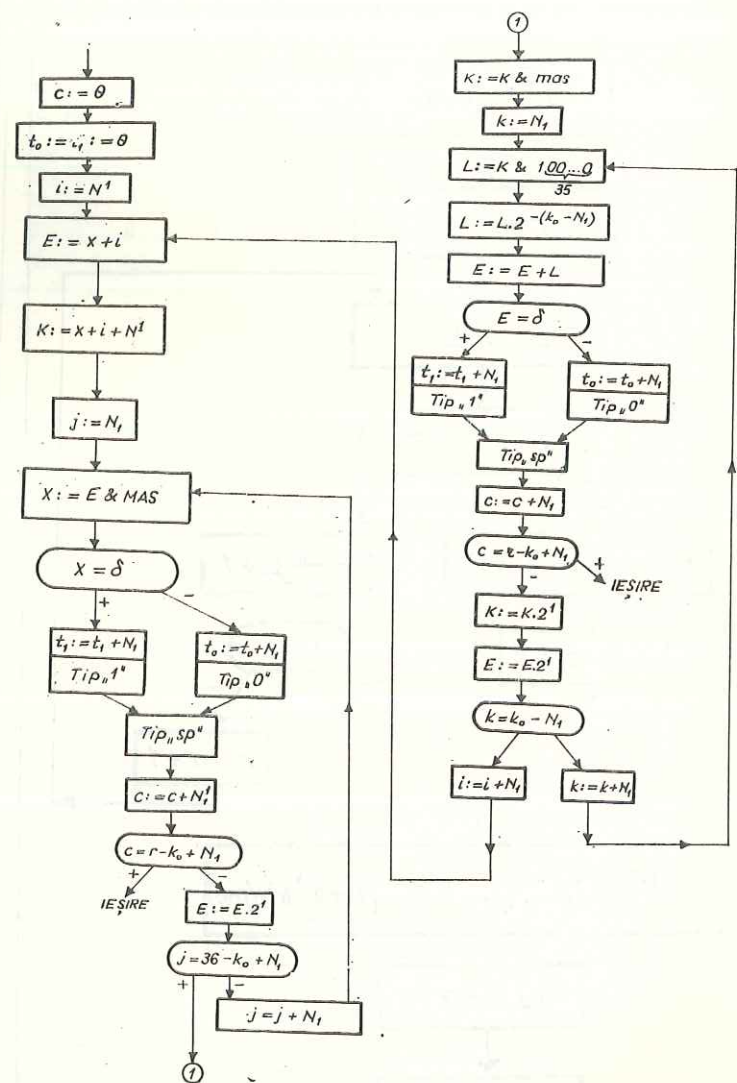


Fig. 5d

schema logică a programului principal

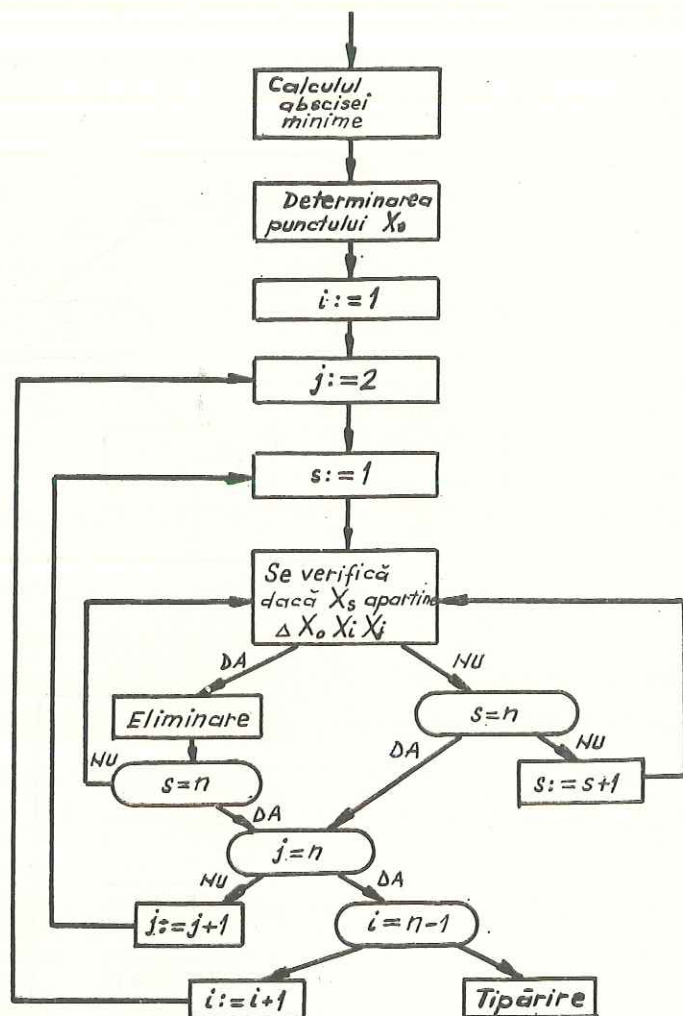


Fig. 6

Schema logică pentru calculul abscisei minime

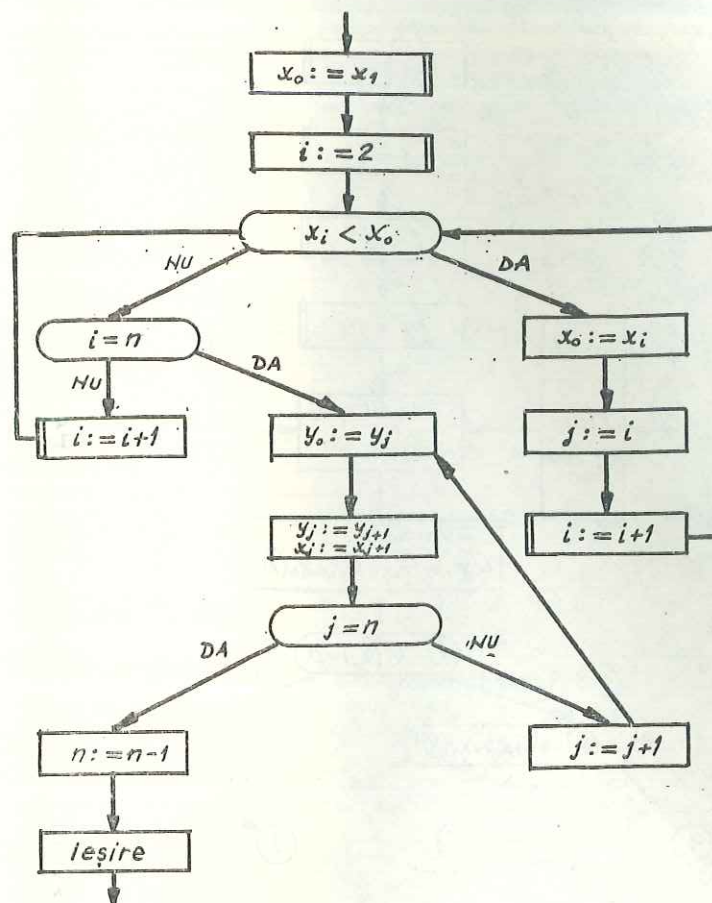


Fig. 7

Schema logică a programului
care verifică dacă

$$X_s \in \Delta X_m X_i X_j$$

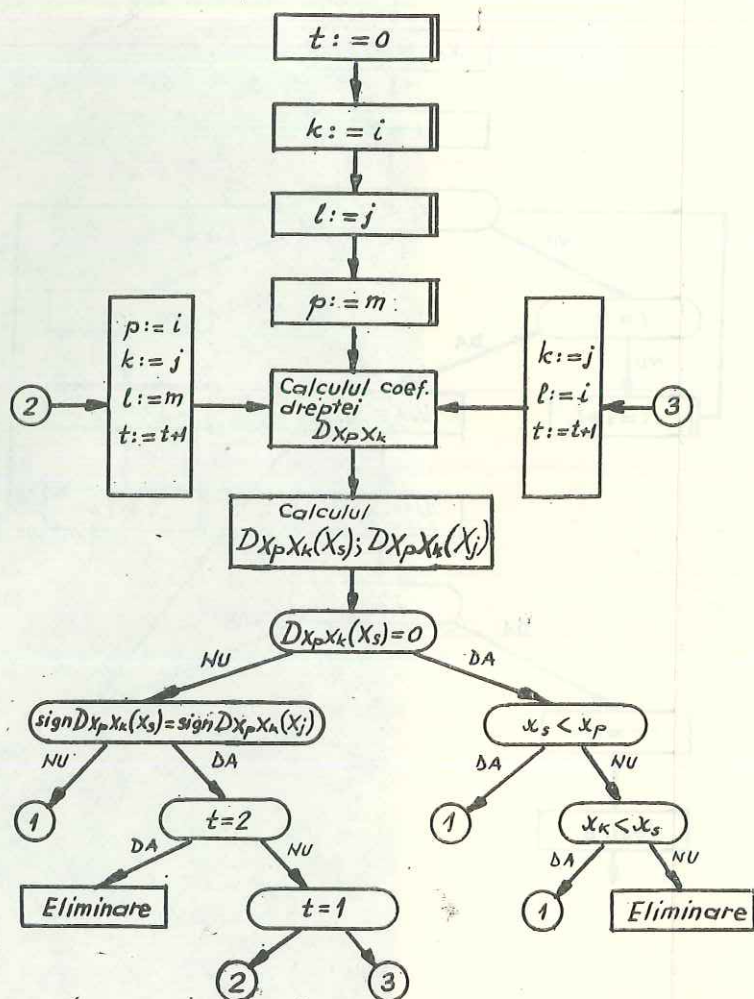


Fig. 8

Schema logică a programului
de eliminare a unui punct

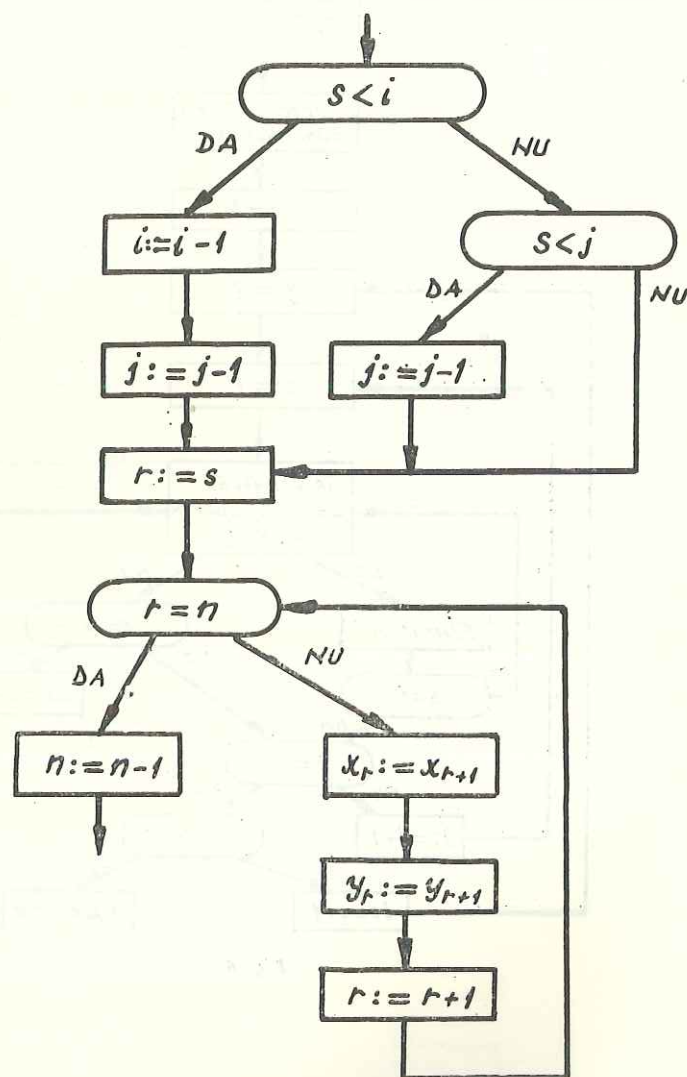


Fig. 9