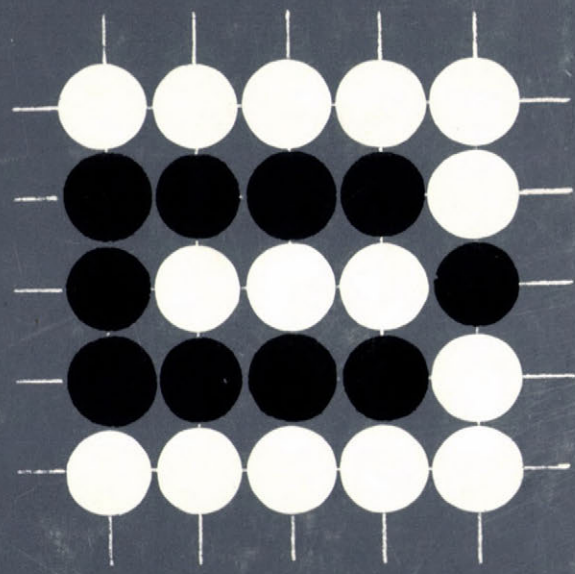
közlemények 19/1978

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

KÖZLEMÉNYEK

1978. JANUÁR

Felelős kiadó:

DR VÁMOS TIBOR
igazgató

ISBN 963 311 054 8
ISSN 0133-7459

Technikai szerkesztő:
Solt Jánosné

TARTALOMJEGYZÉK

CONTENTS

Proceedings of the Computer and Automation Institute
Hungarian Academy of Sciences
Vol. 19.

СОДЕРЖАНИЕ

Труды Исследовательского Института
Вычислительной Техники и Автоматизации
Венгерской Академии Наук
Выпуск 19.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 19/1978.

THE LATTICE OF SYMMETRIC LANGUAGES INVARIANT UNDER INNER LINEAR TRANSFORMATIONS *

Bagyinszki János – Demetrovics János
MTA KFKI – MTA SzTAKI

Let $V = \{1,2, \ldots, k-1\}$ for $k \geqslant 2$, $V_0 = V \cup \{0\}$ a finite alphabet and L a language over $V_0 : L \subseteq V_0^*$. Notations: $V_0^* = V_0^+ \cup \{\varepsilon\}$, $V_0^+ = \bigcup_{n=1}^{\infty} V_0^n$, $V_1 \cdot V_2 = \{v_1 v_2 \mid v_1 \in V_1, v_2 \in V_2\}$, and "$\varepsilon$" is the empty sentence. A language L is termed symmetric, if it contains the word $a_0 a_{\pi(1)} \ldots a_{\pi(n)}$ for any $a_0 a_1 \ldots a_n \in L \subseteq V_0^+$ and for any permutation $\pi \in S_n$ over the index-set $N = \{1,2, \ldots, n\}$. The class of symmetric languages (S-languages) over the alphabet V_0:

$$\mathcal{S} = \{L \mid (a_0 a_1 \ldots a_n \in L \subseteq V_0^+) \text{ iff } (V\pi \in S_n)(a_0 a_{\pi(1)} \ldots a_{\pi(n)} \in L)\}.$$

A language $L \subseteq V_0 V^*$ is said to be invariant under inner linear transformations (IL-language) if it is closed under the following two operations 0_1 and 0_2 $(a_0 a_1 \ldots a_n, b_0 b_1 \ldots b_m \in L)$:

1.) $0_1(a_0) = a_0$, $0_1(a_0 a_1) = a_0 a_1$,

$$0_1(a_0 a_1 \ldots a_n) = a_0 \ldots a_{n-2} a' \text{ for } n \geqslant 2, \quad a' = \begin{cases} a_{n-1} + a_n, & \text{if } a_{n-1} + a_n \neq 0; \\ \varepsilon, & \text{if } a_{n-1} + a_n = 0. \end{cases}$$

2.) $0_2(a_0, b_0 b_1 \ldots b_m) = a_0$, $0_2(a_0 a_1, b_0 b_1 \ldots b_m) = (a_0 + c_{10}) c_{11} c_{12} \ldots c_{1m}$,

$$0_2(a_0 a_1 \ldots a_n, b_0 b_1 \ldots b_m) = (a_0 + c_{n0}) a_1 \ldots a_{n-1} c_{n1} \ldots c_{nm}, \quad \text{for } n \geqslant 2,$$

$$c_{ij} = \begin{cases} a_i \cdot b_j, & \text{if } j = 0 \text{ or } a_i \cdot b_j \neq 0; \\ \varepsilon, & \text{if } j \neq 0 \text{ and } a_i \cdot b_j = 0. \end{cases}$$

(the addition "+" and multiplication "·" are carried out mod k in this lecture.)

The class of IL-languages over the alphabet V_0 is

$$\mathcal{I} = \{L \mid \text{if } \underline{a}, \underline{b} \in L \subseteq V_0 V^*, \text{ then } 0_1(\underline{a}), \ 0_2(\underline{a}, \underline{b}) \in L\}.$$

The main purpose of this lecture is to investigate the class of symmetric languages invariant under inner linear transformation (SIL-languages):

$$\mathcal{L} = \mathcal{S} \cap \mathcal{I}$$

* Presented at the Internat. Colloquium on Automata and Formal Languages, Szeged, (Hungary), Aug 30 – Sept. 2, 1977.

Results:

a.) The complete lattice-structure of $\mathscr{L}$ and therefore the exact (finite) cardinality of $\mathscr{L}$ are presented for $k = p$ prime number.

b.) The base and the rank of each languages $L \in \mathscr{L}$ are given.

c.) The elements of $\mathscr{L}$ are generable by regular grammars.

d.) The correspondence between $\mathscr{L}$ and the class of linear functions on the set V_0 is presented.

Remark: A. Salomaa presented in [6] very impressive results concerning closed sets of sequences over the set V_0. Still, he defined the closedness of a set in a way according to Malcev-algebras and thus a little different from ours. Besides, his essential results concern the case of infinite cardinalities. Moreover, according to his definition, it is not languages what he deals with. In the case $k = p$ with p being a prime number he presents the sets corresponding to the sets $L, L_\alpha, L_\Delta, L_{\Delta 0}, L^{(1)}$ with the remark that because of the finiteness of $L^{(1)}$ the cardinality in question must be finite as well.

A word $a_0 a_1 \ldots a_n$ can be interpreted as a linear polynominal over $GF(p)$; by the correspondence $a_0 a_1 \ldots a_n \longleftrightarrow a_0 + a_1 x_1 + \ldots + a_n x_n$ a connection between many-valued logics and our results is presented. Some significant results on many-valued logics are tabelled as follows:

Structure of 2-valued logics (Post-lattice)	– E. Post, 1921. [4]
All precomplete subsets in P_3	– Sz. V. Jablonszkij 1953. [7]
Closed and precomplete subsets in P_k	– Sz. V. Jablonszkij, 1958. [7]
All precomplete subsets in P_k	– I. Rosenberg, 1965. [5]
Closed subsets { infinitely generated in P_k $(k \geqslant 3)$ / without bases	– Ju. I. Janov, A.A. Muchnik, 1959. [8]
Maximal and precomplete sets in $L(k)$	– A. Salomaa, 1964. [6]
Lattice of $SIL(p)$-languages	– J. Bagyinszki, J. Demetrovics, 1976. [1], [2].

Table 1.

It can be checked, that the following sets are *SIL*-languages (notations: $a_0 \epsilon V_0$, $a_i \epsilon V$ for $i \geqslant 1$, $\sum_{i=1}^{n} a_i = a$, $\alpha \epsilon V_0$):

$$L(k) = \{ a_0 a_1 \ldots a_n \mid n = 0,1,2, \ldots \} = V_0 \cdot V^*$$

$$L_\Delta = \{ a_0 a_1 \ldots a_n \mid a = 1 \}$$

$$L_\alpha = \{ a_0 a_1 \ldots a_n \mid a_0 = \alpha(1-a),\ n \geqslant 1 \} \cup \{ \alpha \}$$

$$L^{(1)} = V_0 \cup V_0 V$$

$$L^{(0)} = V_0$$

$$L^{(1)} \setminus L^{(0)} (= V_0 V)$$

$$L_{\Delta\alpha} = L_\Delta \cap L_\alpha = L_{\Delta 0}$$

$$L_\Delta^{(1)} = L_\Delta \cap L^{(1)}$$

$$L_\alpha^{(1)} = L_\alpha \cap L^{(1)} = \{ a_0 a_1 \mid a_0 = \alpha(1-a_1) \} \cup \{ \alpha \}$$

$$L_\alpha^{(1)} \setminus \{ \alpha \} (= L_\alpha \cap (L^{(1)} \setminus L^{(0)}))$$

$$L_\alpha^{(0)} = L_\alpha \cap L^{(0)} = \{ \alpha \}$$

$$L_\alpha^{(1)} \cup L^{(0)}.$$

Theorem 1: *If* $k = p$ *(prime), then* $< \mathcal{L} \cup \{\emptyset\}, \subseteq >$ *is a finite lattice, with the unit element* $L(p)$ *and zero element* $\emptyset$ *(empty set).*

The linear sublattice of the known Post-lattice ($k = 2$) is given for an example on Fig. 1.

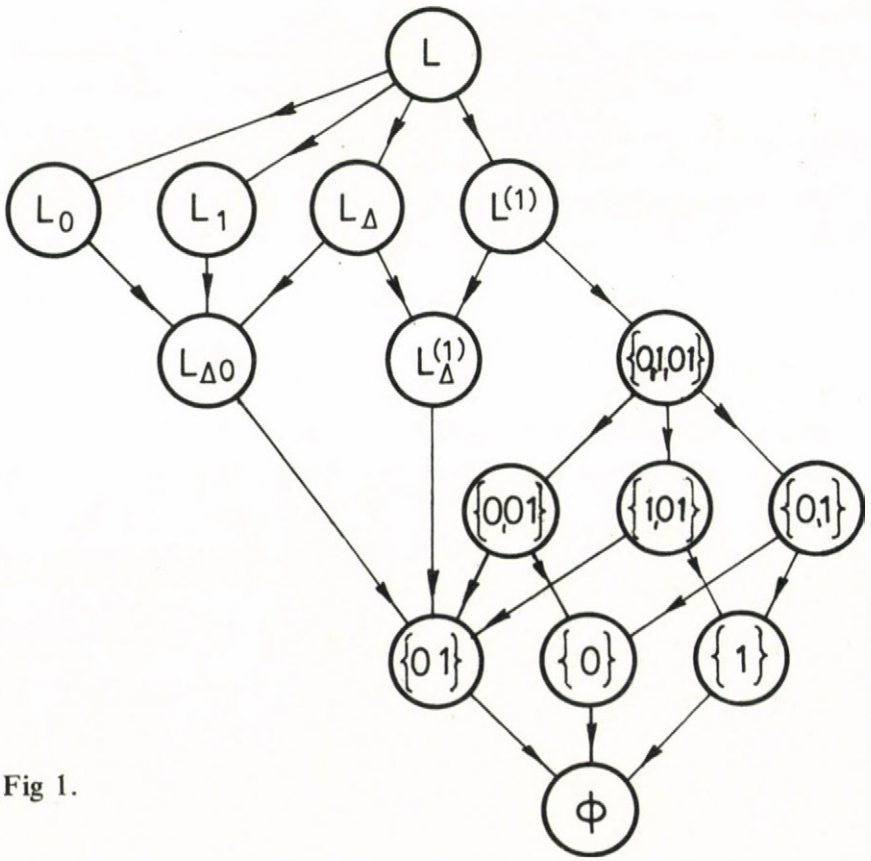


Fig 1.

In the rest of the paper k is supposed to be a prime number $p \geqslant 3$.

Let $[A] \in \mathscr{L}$ be the least language in $\mathscr{L}$ containing A (thus, $A \subseteq \overline{A} \subseteq [A]$, $\overline{A}, [A] \in \mathscr{L}$ implies $\overline{A} = [A]$.). The set $B \subseteq L' \in \mathscr{L}$ is a base of L', if $[B] = L'$ and $B' \subseteq B$, $[B'] = L'$ implies $B' = B$.

"A" is called to be complete in $L' \in \mathscr{L}$, if $[A] = L'$.

Let $L', L'' \in \mathscr{L}$. L'' will be called precomplete in L', if $L'' \subset A \subseteq L'$ implies $[A] = L'$.

Let L' be a *SIL*-language. To prove that all the precomplete *SIL*-languages in L' are given, we need bases of the *SIL*-languages. Bases of $L(p)$, precomplete languages in $L(p)$ and their bases are given in theorems 2 – 6. (without proofs). This is the first level in the lattice $< \mathscr{L} \cup \{\emptyset\}, \subseteq >$.

Theorem 2. *The following sets are bases in* $L(p)$:

(a) $\{011, 11\}$.

(b) $\{a_0 a_1 a_2, b_0, c_0\}$; $a = 1$, $a_0 = 0$, $b_0 \neq c_0$.

(c) $\{a_0 a_1 a_2, b_0\}$; $a = 1$, $a_0 \neq 0$.

(d) $\{a_0 a_1 a_2, b_0\}$; $a \neq 1$, $b_0 \neq (p - a_0)(a - 1)^{p-2}$.

Theorem 3.

(1) *Languages* L_α *are precomplete in* $L(p)$, $\alpha = 0, 1, \ldots, p - 1$.

(2) *The language* L_Δ *is a precomplete in* $L(p)$.

(3) *The language* $L^{(1)}$ *is a precomplete in* $L(p)$.

Theorem 4.

(a) *The set of base-functions (bases with one element each) in the language* L_α *is*: $L_\alpha \setminus (L_\Delta \cup L^{(1)})$.

(b) *The set of base functions in the set* L_Δ *is*: $L_\Delta \setminus (L_{\Delta 0} \cup L^{(1)})$.

(c) *The set of base functions in the set* $L_{\Delta 0}$ *is*: $L_{\Delta 0} \setminus L^{(1)}$.

Let $c_0 \in V_0$, $B = \{a_{10} a_{11}, a_{20} a_{21}, \ldots, a_{s0} a_{s1}\}$, *and* $r_i = (a_{i1})$ *be the multiplicative order of* $a_{i1} \in V$.

Theorem 5.

A.) *The following statements are equivalent*:

(1) *The set* B *is a base in the language* $L^{(1)} \setminus L^{(0)}$.

(2) *The set* $B_0 = B \cup \{c_0\}$ *is a base in the language* $L^{(1)}$.

(3) *For elements of the set* B *are valid*:

(a) *l.c.m.* $\{r_1, \ldots, r_s\} = p - 1$.

(b) $B \setminus L_\alpha^{(1)} \neq \emptyset \quad \alpha = 0,1, \ldots, p-1$.

(c) *if* $B' \subset B$, $B' \neq B$, *then* (a) *and* (b) *cannot hold for* B' *at the same time.*

B.) *If* B *is a base of* $L^{(1)} \setminus L^{(0)}$, *then* $|B| \geqslant 2$, $|B_0| \geqslant 3$.

Theorem 6.

Every language $L \in \mathscr{L}$ *different from* $L(p)$ *is a subset at least in one of the precopmlete languages* $L_0, L_1, \ldots, L_{p-1}, L_\Delta, L^{(1)}$.

Languages of the next level can be determined in a similar way. Results are presented only in a more compact form on Fig. 2. giving the structure of the lattice $<\mathscr{L} \cup \{\emptyset\}, \subset>$. If the language L'' is precomplete in $L' \in \mathscr{L}$, then L'' is of the next level and there is an edge connecting it with L'.

It can be seen that the set $L^{(1)} \setminus L^{(0)}$ constitutes a group of order $p(p-1)$ with respect to the operation 0_2. Let $p - 1 = q_1^{\kappa_1} \ldots q_u^{\kappa_u}$ be the canonical decomposition of $p - 1$, where $2 = q_1 < q_2 < \ldots < q_u$ are prime numbers, $\kappa_i \geqslant 1$, $p_i = \frac{p-1}{q_i}$ and $L^{(1,i)} = \{a_0 a | r(a) \text{ divides } p_i\}_{i=1,2,\ldots u}$ ($r(a)$ is the order of "a" in the group $< V, \cdot >$).

To complete the structure of $L^{(1)}$ it needs, for example, the following statements:

(1) The group $L_\Delta^{(1)}$ is contained in a subgroup G of the group $L^{(1)} \setminus L^{(0)}$, if and only if the order of G, $|G| \geqslant p$.

(2) The subgroup $G \subseteq L^{(1)} \setminus L^{(0)}$ of order $|G| \leqslant p - 1$ is cyclic, G is a subgroup of $L_\alpha^{(1)} \setminus \{\alpha\}$ for some suitable α.

The next theorem involves the result on cardinality cited in theorem 1.

Theorem 7.

(1) $\mathscr{L}$ has the cardinality

$$|\mathscr{L}| = p + 2 - (p-2)2^p + 2d(p-1) + 2p \cdot \sum_{e|p-1} 2^e.$$

(2) $\mathscr{L}$ has maximal and minimal chains of lenght $p + 2 + \sum_{i=1}^{u} \kappa_i$ and 3, respectively.

At last, we shall show that $\mathscr{L}$ is a subclass in the class of regular languages . Thus we shall describe the regular grammars which generate the elements of $\mathscr{L}$, and finite accepting automata. The grammars G for languages L, L_Δ, $L_{\Delta 0}$ and L_α are given as follows. Let $G = (K, V_0, P, A_0)$ be with non-terminals K, terminals V_0, productions P.

$$L: K_L = \{A_0, A_1\} \quad , P_L = \bigcup_{\substack{i \in V \\ j \in V_0}} \{A_0 \to j,\ A_0 \to jA_1, A_1 \to i,\ A_1 \to iA_1\}.$$

$$L_\Delta - L_{\Delta 0}: K_0 = \{A_0, A, A_1, \ldots, A_{p-1}\},$$

$$P_\Delta = \bigcup_{\substack{i,j \in V \\ j_0 \in V_0}} \{A_0 \to j_0 A,\ A \to 1, A \to iA_i, \qquad A_i \to p - i + 1, A_i \to jA_{i+j}\}.$$

$$P_{\Delta 0} = P_\Delta \setminus \{A_0 \to jA \mid j \in V\}.$$

$$L_\alpha: K = \{A_0, A_1, \ldots, A_{p-1}, \quad B_0, B_1, \ldots, B_{p-1}\}$$

$$P_\alpha = \bigcup_{\substack{i,j,m \in V \\ j_0 \in V_0}} \{A_0 \to \alpha, A_0 \to jB_{j \cdot \beta}, B_m \to p - m + 1, B_m \to i \cdot A_{m+i}, A_i \to p - i + 1, A_i \to jA$$

$$\beta = \alpha^{p-2}.$$

It is clear, that for finite languages there are accepting finite automata.

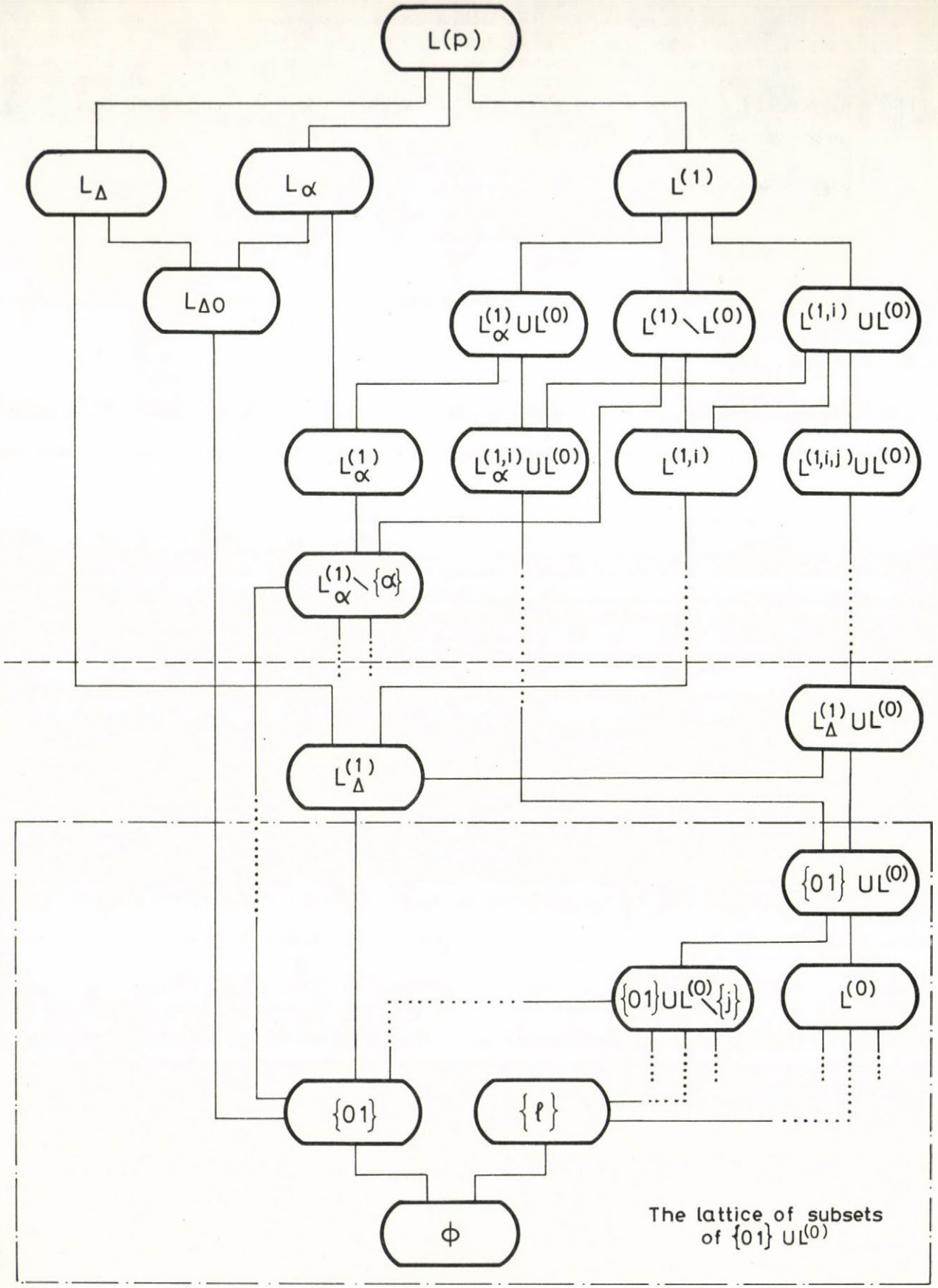

L(p)
L_Δ
L_α
$L^{(1)}$
$L_{\Delta 0}$
$L^{(1)}_\alpha \cup L^{(0)}$
$L^{(1)} \setminus L^{(0)}$
$L^{(1,i)} \cup L^{(0)}$
$L^{(1)}_\alpha$
$L^{(1,i)}_\alpha \cup L^{(0)}$
$L^{(1,i)}$
$L^{(1,i,j)} \cup L^{(0)}$
$L^{(1)}_\alpha \setminus \{\alpha\}$
$L^{(1)}_\Delta \cup L^{(0)}$
$L^{(1)}_\Delta$
$\{01\} \cup L^{(0)}$
$\{01\} \cup L^{(0)} \setminus \{j\}$
$L^{(0)}$
$\{01\}$
$\{\ell\}$
Φ
The lattice of subsets of $\{01\} \cup L^{(0)}$

Fig. 2.

REFERENCES

[1] Bagyinszki J. – Demetrovics J. : The structure of linear classes in prime valued logics (in Hungarian) MTA SzTAKI Közlemények, 16/1976, 25-52.

[2] Bagyinszki J. – Demetrovics J. : The structure of the class of symmetric languages invariant for inner linear transformations, in Proc.s of second Hung. Comp. Sci. Conf. 100-130. (1977, Budapest)

[3] Hopcroft J.B. – Ullman J.D. : Formal languages and their relation to automata, 1969. Addison – Wesley

[4] Post, E. : The two-valued iterative systems of mathematical logic, Annals of Math. Studes 5 (1941).

[5] Rosenberg, I. : La structure des fonktions de plusieurs variables sur un ensemle fini. C.R. Acad. Sci. Paris Ser A.B 260 (1965) 3817-19.

[6] Salomaa, A.; On infinitely generated sets of operations in finite algebras. Ann. Univ. Turkuensis, ser. A.I. 74. (1964) 1-13.

[7] Jablonszkij, S.V. : Functional constructions in k-valued logics (Russian) Trudy Mat. - Inst. Steklov 51 (1958) 5-142.

[8] Janov, Ju.I., Muchnik, A.A.: Existence of k-valued closed classes having no finite basis (Russian), Dokl.Akad.Nauk. SSSR. 127 (1959) 44-46.

Összefoglaló

A belső lineáris transzformációra invariáns szimmetrikus nyelvek hálója

Bagyinszki János – Demetrovics János

A V_0 alaphalmazon definiált L nyelvet szimmetrikusnak nevezzük, ha $a_0a_1a_2 \ldots a_n \in L$ esetén $a_0a_{\pi(1)}a_{\pi(2)} \ldots a_{\pi(n)} \in L$ is igaz minden $\pi(x)$ permutációra. Egy nyelv invariáns a belső lineáris transzformációra nézve, ha zárt a dolgozatban definiált 0_1 és 0_2 operációkra. A szimmetrikus és a belső lineáris transzformációra invariáns nyelveket SIL-nyelveknek nevezzük. A jelen dolgozatban azokat SIL nyelveket tanulmányozzuk, amelyekben a V_0 alaphalmaz számossága primszám.

Dolgozatunkban leírjuk a SIL(p) nyelvek osztályának teljes szerkezetét, megadjuk a tartalmazási reláció által indukált hálót, a minimális bázisokat és a pontos elemszámot. Megmutatjuk, hogy a SIL(p)- nyelvek mindegyike reguláris és utalunk a többértékü logika lineáris függvény-osztályaival való kapcsolatra.

Р Е З Ю М Е

Структура симметрических языков, инвариантных по отношению к внутренним линейным трансформациям.

Янош Бадински - Янош Деметрович

Назовем язык L симметрическим над алфавитом $V_c=\{0,1,2,\ldots,k-1\}$, если для каждой перестановки $\pi(x)$ из $a_0a_1a_2\ldots a_n \in L$ следует, что $a_0a_{\pi(1)}\ldots a_{\pi(n)} \in L$. . Язык L является инвариантным поотношению к внутренним линейным трансформациям, если он замкнут относительно операций O_1, O_2 определенных в данной работе. Язык L называется SIL-языком, если он симметрический и инвариантен по отношению к внутренним линейным трансформациям. В настоящей работе мы исследовали SIL-язык, в случае когда мощность алфавита V_o равна простому числу.

В работе описана полная структура SIL-языков относительно операции включения и охарактерезован каждый класс / SIL-языков/ в этой структуре. Кроме того, для каждого SIL-языка задан минимальный базис и приведена точная арифметическая формула для мощности SIL-языков /элементов в структуре/. Доказывается, что каждый SIL-язык является регулярным и, кроме того, указывается связь между линейными замкнутыми классами k-замкнутой логики и SIL-языками.

MTA Számítástechnikai és Automatizálási Kutató Intézete, **Közlemények** 19/1978.

A SUMT MÓDSZER ALKALMAZÁSA LOGARITMIKUSAN KONKÁV FELTÉTELI FÜGGVÉNYEKET TARTALMAZÓ NEM-LINEÁRIS PROGRAMOZÁSI FELADAT MEGOLDÁSÁRA

Rapcsák Tamás

A SUMT (Sequential Unconstrained Minimization Techniques) olyan nem-lineáris programozási algoritmus, amely a feladat megoldását feltétel nélküli minimalizálások sorozatára vezeti vissza. Ha a célfüggvény konvex és a feltételi függvények konkávak, akkor az algoritmus globális minimumot határoz meg. Ebben a cikkben megmutatjuk, hogy a SUMT belső pont algoritmusai logaritmikusan konkáv feltételi függvények mellett nem korlátos tartomány esetén is globális optimumot határoznak meg. Ilyen tipusu feladatok Prékopa András munkáiban fordulnak elő. [9], [10], [11]. Az itt található konvergencia bizonyitások is eltérnek a korábbiaktól. A dolgozat végén számitástechnikai tapasztalatokat közlünk.

Definició. Egy R^n-beli $g(\underline{x})$ függvény logaritmikusan konkáv, ha $\ln g(\underline{x})$ konkáv függvény. A következő feladattal foglalkozunk:

$$
\begin{aligned}
&\min f(\underline{x}),\\
(1)\qquad &g_i(\underline{x}) \geqslant p_i, \quad i = 1, \ldots, r,\\
&g_i(\underline{x}) \geqslant 0, \quad i = r+1, \ldots, m
\end{aligned}
$$

ahol $f(\underline{x})$, $-g_i(\underline{x})$, $i = r+1, \ldots, m$ R^n-beli konvex függvények, a $g_i(\underline{x})-k$, $i = 1, \ldots, r$ pedig R^n-beli logaritmikusan konkáv függvények, $p_i > 0$, $i = 1, \ldots, r$.

1. Lemma. *Ha $g(\underline{x})$ logaritmikusan konkáv, akkor $p > 0$ esetén $g(x) - p$ is logaritmikusan konkáv* $(\underline{x} \mid g(\underline{x}) > p\} - n$.

Bizonyitás. A logaritmikus konkávitás definiciójából következik, hogy $g(\underline{x})$ az egész téren folytonos..Ezért elegendő megmutatni, hogy

$$g\left(\frac{\underline{x}_1 + \underline{x}_2}{2}\right) - p \geqslant \sqrt{(g(\underline{x}_1) - p)(g(\underline{x}_2) - p)} \quad \text{ha} \quad \underline{x}_1, \underline{x}_2 \in \{\underline{x} \mid g(\underline{x}) > p\} \quad .$$

De $g(\underline{x})$ logaritmikusan konkáv, igy

$$g\left(\frac{\underline{x}_1 + \underline{x}_2}{2}\right) \geqslant \sqrt{g(\underline{x}_1)g(\underline{x}_2)}, \qquad \text{illetve}$$

$$g\left(\frac{\underline{x}_1 + \underline{x}_2}{2}\right) - p \geqslant \sqrt{g(\underline{x}_1)g(\underline{x}_2)} - p.$$

Tehát elegendő belátni, hogy

$$\sqrt{g(\underline{x}_1)g(\underline{x}_2)} - p \geqslant \sqrt{(g(\underline{x}_1) - p)(g(x_2) - p)}, \quad \text{ha} \quad \underline{x}_1, \underline{x}_2 \in \{\underline{x} \mid g(\underline{x}) > p\} \; .$$

A geometriai egyenlőtlenség miatt

$$2\sqrt{g(\underline{x}_1)g(\underline{x}_2)} \leqslant g(\underline{x}_1) + g(\underline{x}_2).$$

Mivel $p > 0$, igy

$$-2p\sqrt{g(\underline{x}_1)g(\underline{x}_2)} \geqslant -p(g(x_1) + g(\underline{x}_2)).$$

Mindkét oldalhoz $g(\underline{x}_1)g(\underline{x}_2) + p^2 - t$ adva

$$g(\underline{x}_1)g(\underline{x}_2) - 2p\sqrt{g(\underline{x}_1)g(\underline{x}_2)} + p^2 \geqslant g(\underline{x}_1)g(\underline{x}_2) - p(g(\underline{x}_1) + g(\underline{x}_2)) + p^2$$

azaz

$$(\sqrt{g(\underline{x}_1)g(\underline{x}_2)} - p)^2 \geqslant (g(\underline{x}_1) - p)(g(\underline{x}_2) - p).$$

Ha $\underline{x}_1, \underline{x}_2 \in \{\underline{x} | g(\underline{x}) > p\}$, akkor

$$\sqrt{g(\underline{x}_1)g(\underline{x}_2)} - p \geqslant \sqrt{(g(\underline{x}_1) - p)(g(\underline{x}_2) - p)},$$ ami éppen az állitás.

2. **Lemma.** *Ha* $g(\underline{x})$ *logaritmikusan konkáv, akkor* $p > 0$ *esetén* $1/g(\underline{x}) - p$ *konvex* $\{\underline{x}/g(\underline{x}) > p\} - n$.

Bizonyitás. Az 1. Lemma miatt $\ln(g(\underline{x}) - p)$ konkáv $\{\underline{x}/g(\underline{x}) > p\} - n$.
Ebből következik, hogy

$$e^{-\ln(g(\underline{x})-p)} = \frac{1}{g(\underline{x}) - p} \quad \text{konvex} \quad \{x | g(\underline{x}) > p\} - n.$$

3. **Lemma.** *Ha a* $g_i(\underline{x}) - k$, $i = 1, \ldots, r$ *logaritmikusan konkávak,*

$$P = \{x | g_i(x) \geqslant p_i,\ i = 1, \ldots, r\},\ P_0 = \{\underline{x} | g_i(\underline{x}) > p_i,\ i = 1, \ldots, r\}$$

nem üres, akkor P_0 *megegyezik* P *relativ belső pontjainak halmazával.*

Bizonyitás. A $g_i(\underline{x}) - k$ $i = 1, \ldots, r$ folytonosságából következik, hogy P relativ belső pontjainak halmaza tartalmazza P_0-t. A forditott állitást indirekt uton látjuk be. Tegyük fel, hogy P_0 nem tartalmazza P relativ belső pontjainak halmazát. Akkor található $\underline{x}_1$ relativ belső pont, amelyre $g_i(x_1) = p_i$ valamely i indexre. Mivel x_1 relativ belső pont, ezért létezik olyan környezete, amely P-beli pontokból áll. Legyen $x_0 \in P_0$. (Ilyen pont található, mert P_0 nem üres.) Kössük össze az x_0-t az x_1-el és hosszabbitsuk meg ugy ezt a szakaszt hogy a végpontja x_2 az x_1 környezetén belül maradjon. Akkor létezik $0 < \lambda < 1$ ugy, hogy $x_1 = \lambda x_0 + (1 - \lambda)x_2$ és $g_i(\underline{x}_0) > p_i$, $g_i(x_1) = p_i$, $g_i(x_2) \geqslant p_i$.
A $g_i(\underline{x})$ logaritmikus konkávitása miatt

$$p_i = g_i(\underline{x}_1) \geqslant g_i(\underline{x}_0)^{\lambda} \cdot g_i(\underline{x}_2)^{1-\lambda} > p_i^{\lambda} \cdot p_i^{1-\lambda} = p_i.$$

Ez ellentmondás, igy bebizonyitottuk az állitást. Ebből következik, hogy P_0 lezártja megegyezik P-vel.. Legyenek $G_i(\underline{x}) = g_i(\underline{x}) - p_i$, $i = 1, \ldots, r$; $G_i(\underline{x}) = g_i(\underline{x})$, $i = r + 1, \ldots, m$.

Irjuk át (1)-t az alábbi formába

$$(2) \qquad \begin{aligned} &\min f(\underline{x}) \\ &G_i(\underline{x}) \geqslant 0, \quad i = 1, \ldots, m, \end{aligned}$$

ahol $G_i(x) - k \quad i = 1, \ldots, m$ *logaritmikusan konkávak* R^n valamilyen részhalmazán.

A (2) feladatban a megengedett tartomány konvex. Ugyanis ha $G(\underline{x})$ valamilyen tartományon logaritmikusan konkáv, akkor ott kvázikonkáv. Ebből következik, hogy (2)-ben bármely lokális optimum globális. (A nem-lineáris programozási algoritmusok lokális optimumot határoznak meg.)

A (2) feladatot "belső pont" algoritmussal oldjuk meg. Az $\{ \underline{x} | G_i(\underline{x}) > 0, \quad i = 1, \ldots, m \}$ -n értelmezünk egy függvénysorozatot,ugy, hogy a függvénysorozat tagjainak feltétel nélküli minimumai a (2) feladat megoldásához konvergáljanak.
Legyen

$$P_1(\underline{x},r_k) = f(\underline{x}) + r_k I_1(\underline{x}), \text{ ahol } I_1(\underline{x}) = \sum_{i=1}^{m} \frac{1}{G_i(\underline{x})},$$

$$P_2(\underline{x},r_k) = f(\underline{x}) + r_k I_2(\underline{x}), \text{ ahol } I_2(\underline{x}) = \sum_{i=1}^{m} - \ln G_i(\underline{x}).$$

A függvénysorozatokban $r_{k-1} > r_k > 0$ és $\lim_{k \to \infty} r_k = 0,$

$I_1(\underline{x})$, $I_2(\underline{x})$-re pedig teljesülnek:

1, $I_1(\underline{x})$, $I_2(\underline{x})$ folytonosak R_0-on, ahol $R_0 = \{ \underline{x} | G_i(\underline{x}) > 0, \quad i = 1, \ldots, m \}$.

2, ha $\underline{x}_k$ tetszőleges sorozata R_0-nak és $\underline{x}_k \to \underline{x}_0$ ugy, hogy

$G_i(\underline{x}_0) = 0$ legalább egy i indexre, akkor

$\lim_{k \to \infty} I_1(\underline{x}_k) = + \infty,$ illetve

$\lim_{k \to \infty} I_2(\underline{x}_k) = + \infty.$

A 2. tulajdonság miatt a függvénysorozat minden tagjának minimum helye a megengedett tartomány "belsejébe" van kényszeritve, igy a feltétel nélküli optimalizálás során elvileg soha nem lépünk ki R_0-ból. Látható, hogy $P_1(\underline{x}, r_k)$, $P_2(\underline{x},r_k)$ tetszőleges k mellett konvexek. A konvergencia tételekhez szükséges az alábbi lemma.

4. **Lemma.** *Ha* $u(\underline{x})$ R^n*-beli konkáv függvény, a* $G_i(\underline{x})$*-k* $i = 1, \ldots, m$ *logaritmikusan konkávak* R^n *valamilyen részhalmazán és* $Q = \{ \underline{x} | u(\underline{x}) \geqslant 0, \; G_i(\underline{x}) \geqslant 0, \quad i = 1, \ldots, m \}$ *nem üres, korlátos, akkor tetszőleges* $k \geqslant 0$ *értékre*

$$Q_K = \{ \underline{x} | u(\underline{x}) \geqslant - k, \; G_i(\underline{x}) \geqslant 0, \quad i = 1, \ldots, m \} \text{ korlátos.}$$

Bizonyitás. Tegyük fel, hogy Q_K nem korlátos. $Q \subset Q_K$ és Q konvex, zárt. Ha $x_1 \epsilon Q$, akkor x_1-ből indítható olyan sugár, amely metszi Q határát és Q_K-ban halad. Legyen x_2 olyan pontja a sugárnak, hogy $u(\underline{x}_2) = -\delta < 0$, $G_i(\underline{x}_2) \geqslant 0$, $i = 1, \ldots, m$. Az $u(x)$ konkáv, tehát

$$-\delta = u(\underline{x}_2) \geqslant \lambda u \left\{ \underline{x}_1 + \frac{1}{\lambda}(\underline{x}_2 - \underline{x}_1) \right\} + (1-\lambda)u(\underline{x}_1), \quad \text{ahol} \quad 0 < \lambda < 1.$$

Igy

$$u(\underline{x}_1 + \frac{1}{\lambda}(\underline{x}_2 - \underline{x}_1)) \leqslant \frac{-\delta - (1-\lambda)u(\underline{x}_1)}{\lambda} \leqslant -\frac{\delta}{\lambda}, \quad \text{mivel } u(\underline{x}_1) \geqslant 0, \; 1-\lambda > 0.$$

Ha $\lambda < \delta/k$, akkor $u(\underline{x}_1 + \frac{1}{\lambda}(\underline{x}_2 - \underline{x}_1)) < -k$. Ez ellentmondás.

Következmény. Ha a (2)-ben a minimum pontok halmaza nem üres, korlátos, akkor $\{ \underline{x} | f(\underline{x}) \leqslant k, \; G_i(\underline{x}) \geqslant 0, \quad i = 1, \ldots, m \}$ korlátos.

Bizonyitás. Ha $u(\underline{x}) = -f(\underline{x}) + v^x$ (ahol $v^x = \min_{x \epsilon R} f(\underline{x})$), akkor a 3. Lemmát alkalmazva kapjuk az állitást. Ez a lemma biztositja az alábbi feltételek ekvivalenciáját.

(3): a (2) optimum pontjainak halmaza nem üres, korlátos, (4): bármely véges k-ra $\{ \underline{x} | f(\underline{x}) \leqslant k; \; G_i(\underline{x}) \geqslant 0, \quad i = 1, \ldots, n \}$ korlátos.

A következőkben a konvergencia tételeket fogjuk bebizonyitani.

1. **Tétel.** *Ha a* (2)-t *tekintjük és* (3) *teljesül, valamint* $R_0 = \{ \underline{x} | G_i(\underline{x}) > 0, \; i = 1, \ldots, m \}$ *nem üres, akkor bármely* k *értékre* $P_1(\underline{x}, r_k)$ *felveszi a minimumát* R_0 *felett és* $\lim_{k \to \infty} P_1(\underline{x}(r_k), r_k) = v^*$, *ahol* $\underline{x}(r_k) P_1(\underline{x}, r_k)$ *minimum helye* R_0*-on.*

Bizonyitás. 1, Legyen $x_0 \epsilon R_0$ (ilyen pont létezik, mivel R_0 nem üres). Ha $R = \{ \underline{x} | G_i(\underline{x}) \geqslant 0, \; i = 1, \ldots, m \}$, akkor $\{ \underline{x} | P_1(\underline{x}, r_k) \leqslant P_1(\underline{x}_0, r_k), \; \underline{x} \epsilon R \}$ zárt. (Nem üres, mert x_0 eleme). Válasszunk ugyanis ebből a halmazból egy tetszőleges sorozatot, amelyre $\underline{x}_j \to \hat{\underline{x}}$. Megmutatjuk, hogy $\hat{\underline{x}}$ is eleme ennek a halmaznak. A $P_1(\underline{x}, r_k)$-k konvexitásából következik, hogy folytonosak R_0–on, igy ha $\hat{\underline{x}} \epsilon R_0$, akkor igaz az állitás. Ha $\hat{\underline{x}} \notin R_0$, de $\hat{\underline{x}} \epsilon R$, akkor a $P_1(\underline{x}, r_k)$-k folytonossága, illetve $P_1(\underline{x}_j, r_k) \leqslant P_1(\underline{x}_0, r_k)$ miatt ellentmondáshoz jutunk. Elegendő tehát azt belátni, hogy ez a halmaz korlátos bármely k értékre. Mivel $P_1(\underline{x}, r_k) > f(\underline{x})$ tetszőleges k-ra, ezért

$$\{ \underline{x} | P_1(\underline{x}, r_k) \leqslant P_1(\underline{x}_0, r_k); \underline{x} \epsilon R \} \subset \{ \underline{x} | f(\underline{x}) \leqslant P_1(\underline{x}_0, r_k); \underline{x} \epsilon R \}.$$

A jobb oldali halmaz azonban (4) miatt korlátos, igy adódik az állitás.

2. Mivel $f(\underline{x}) < P_1(\underline{x},r_k) < P_1(\underline{x},r_{K-1}) < \ldots < P_1(\underline{x},r_1)$, $\underline{x}\epsilon R_0$ és tetszőleges k esetén, igy

$$\min_{\underline{x}\epsilon R} f(x) = v^* < P_1(\underline{x}(r_k),r_k) < P_1(\underline{x}(r_{k-1}),r_{k-1}) < \ldots < P_1(\underline{x}(r_1),r_1).$$

Tehát $P_1(\underline{x}(r_k),r_k)$ egy szigoruan monoton csökkenő, alulról korlátos végtelen sorozat, igy konvergens.

Állitás. $\lim_{k\to\infty} P_1(\underline{x}(r_k),r_k) = v^*$

Legyen $\varepsilon > 0$. Ekkor $f(\underline{x})$ konvexitása, a $G_i(\underline{x})$-k $i = 1, \ldots, m$ logaritmikus konkávitása miatt létezik $\underline{x}^* \epsilon R_0$, ugy, hogy $f(\underline{x}^*) < v^* + \varepsilon$. Mivel

$$v^* \leqslant P_1(\underline{x}(r_k),r_k) \leqslant P_1(\underline{x}^*,r_k) \leqslant v^* + \varepsilon + r_k I_1(\underline{x}^*), \text{ minden } k\text{-ra igy}$$

$$v^* \leqslant \lim_{k\to\infty} P_1(\underline{x}(r_k),r_k) \leqslant v^* + \varepsilon.$$

ε tetszőlegesen kicsiny lehet, tehát

$$\lim_{k\to\infty} P_1(\underline{x}(r_k),r_k) = v^* \quad , \text{ ami az állitás.}$$

Következmény. 1, $\lim_{k\to\infty} r_k \cdot \sum_{i=1}^{m} \frac{1}{G_i(\underline{x}(r_k))} = 0.$

Ez igaz a tétel állitása, illetve az összegezendő tagok pozitivitása miatt.

2. $\lim_{k\to\infty} f[\underline{x}(r_k)] = v^*.$

3. az $\underline{x}[r_k]$ sorozat bármely konvergens részsorozatának határértéke a (2) probléma optimum helye.

Bizonyitás. Könnyen be lehet látni, hogy a $P_1(\underline{x},r_k)$-k minimum helyeire teljesül az, hogy $f[\underline{x}(r_{k+1})] \leqslant f[\underline{x}(r_k)]$. Ez azt jelenti, hogy $\{\underline{x} | f(\underline{x}) \leqslant f(\underline{x}(r_1)), \underline{x}\epsilon R\}$ tartalmazza az összes $\underline{x}(r_k)$ minimum helyet. De a (4) feltétel miatt a halmaz korlátos, zárt. Tehát a minimum pontok sorozatának van konvergens részsorozata. Ezen konvergens sorozat határértéke $\underline{x}^+$ megoldja a problémát. Ugyanis a $G_i(\underline{x})$-k, $i = 1, \ldots, m$ folytonossága miatt x^+ megengedett pont lesz. Az 1, következmény, illetve az $f(\underline{x})$ folytonossága miatt pedig minimumot ad. A második konvergencia tételhez szükséges az alábbi lemma. A lemma állitását abban az esetben bizonyitjuk, mikor az (1) feladatban az első r feltétel valószinüségi jellegü.

5. **Lemma.** *Ha R_0 nem üres és (2) optimum pontjainak halmaza nem üres, korlátos, akkor a $P_2(\underline{x}, r_k)$ bármely $r_k > 0$ esetén R_0-on alulról korlátos.*

Bizonyitás. Tekintsünk egy R_0-beli elemekből alkotott nem korlátos $y_1, y_2, \ldots$ sorozatot. Be fogjuk bizonyitani, hogy ennek van olyan nem korlátos $y_{l_1}, y_{l_2}, \ldots$ részsorozata, amelyre fennáll az alábbi reláció, tetszőleges $r_k > 0$ esetén.

$$\lim_{s\to\infty} P_2(y_{l_s}, r_k) = +\infty.$$

Elegendő ezt bizonyitani. Tegyük fel, hogy ebből nem következik a $P_2(\underline{x}, r_k)$-k alulról való korlátossága. Ekkor van olyan q_l, $l = 1,2, \ldots$ korlátos sorozat, amelyre $\lim_{l\to\infty} P_2(q_l, r_k) = -\infty$. Legyen $\underline{x}\epsilon R_0$ és vegyük a q_l-t ($l = 1,2, \ldots$) magába foglaló zárt gömb és a $\{\underline{x} | P_2(\underline{x}, r_k) \leqslant P_2(\underline{x}_0, r_k), \underline{x}\epsilon R\}$ metszetét. Ez a metszet korlátos, zárt, R_0-beli halmaz, amely véges sok elem kivételével tartalmazza a q_l-t, $l = 1,2, \ldots$. A $P_2(\underline{x}, r_k)$-k folytonosak, igy a metszet halmazon felveszik a minimumukat. Ez ellentmondás.

Legyen tehát $r_k > 0$, $y_1 y_2, \ldots R_0$-beli elemekből alkotott sorozat. Legyen

$$\underline{x}_0\epsilon R_0 \quad \text{és} \quad D = \{\underline{x} | f(\underline{x}) \leqslant f(\underline{x}_0) + \varepsilon, \underline{x}\epsilon R\},$$

ahol $\varepsilon > 0$. A (3) és (4) miatt D korlátos, zárt. Legyen B a D halmaz határpontjaiból álló halmaz. B szintén korlátos, zárt. Mivel az $y_1, y_2, \ldots$ nem korlátos sorozat, ezért valamely y_{l_s} ($s = 1,2, \ldots$) részsorozatra, $y_{l_s} \notin D$, $s = 1,2, \ldots$. Tekintsük az $\underline{x}_0$-ból induló, ilyen tulajdonságu $y_{l_1}, y_{l_2}, \ldots$ -n áthaladó sugarakat. Ezek metszik a B-t $\underline{z}_{l_1}, \underline{z}_{l_2}, \ldots$ pontokban és $\underline{z}_{l_s} \neq \underline{x}_0$, $s = 1,2, \ldots$. Legyenek $0 < \lambda_{l_s} < 1$ és

$$\underline{z}_{l_s} = \lambda_{l_s} y_{l_s} + (1 - \lambda_{l_s})\underline{x}_0, \qquad s = 1,2, \ldots$$

Ha $l_s \to \infty$, akkor $\lambda_{l_s} \to 0$. A $G_i(\underline{x})$ -k, ($i = 1, \ldots, r$) logaritmikusan konkávok, igy

$$G_i(\underline{z}_{l_s}) \geqslant G_i(y_{l_s})^{\lambda_{l_s}} \cdot G(\underline{x}_0)^{1-\lambda_{l_s}} > 0.$$

Tehát $\underline{z}_{l_s}\epsilon R_0$, $s = 1,2, \ldots$. Másrészt $\underline{z}_{l_s}\epsilon B$, $s = 1,2, \ldots$ igy

$$f(\underline{z}_{l_s}) = f(\underline{x}_0) + \varepsilon, \qquad s = 1,2, \ldots$$

Az $f[\underline{x}]$ konvexitása miatt

$$f(y_{l_s}) \geqslant \frac{f(\underline{z}_{l_s}) - (1 - \lambda_{l_s})f(\underline{x}_0)}{\lambda_{l_s}} = f(\underline{x}_0) + \frac{\varepsilon}{\lambda_{l_s}}. \quad (s = 1,2, \ldots)$$

$\varepsilon > 0$, igy $f(y_{l_s}) \to +\infty$ ha $\lambda_{l_s} \to 0$, (azaz $l_s \to \infty$, $s = 1,2, \ldots$). Az előbbiek miatt

$$0 < G_i(y_{l_s})^{\lambda_{l_s}} \leqslant \frac{G_i(\underline{z}_{l_s})}{G_i(\underline{x}_0)^{1-\lambda_{l_s}}} \leqslant \frac{d_i^0}{G_i(\underline{x}_0)^{1-\lambda_{l_s}}},$$

ahol $d_i^0 = \max_{\underline{x}\epsilon B} G_i(\underline{x})$, $i = 1, \ldots, r$ $s = 1,2, \ldots$,

De

$$0 < G_i(y_{l_s}) < G_i(y_{l_s})^{\lambda_{l_s}} \leqslant \frac{d_i^0}{G_i(x_0)^{1-\lambda_{l_s}}} \quad i = 1, \ldots, r; \quad s = 1,2, \ldots$$

Legyen

$$P_2'(\underline{x}, r_k) = f(\underline{x}) + r_k \sum_{i=1}^{r} - \ln G_i(x),$$

igy

$$P_2'(y_{l_s}, r_k) = f(y_{l_s}) + r_k \sum_{i=1}^{r} - \ln G_i(y_{l_s}) \geqslant$$

$$\geqslant f(\underline{x}_0) + \frac{\epsilon}{\lambda_{l_s}} + r_k \cdot \sum_{i=1}^{r} - \ln\left(\frac{d_i^0}{G_i(x_0)^{1-\lambda_{l_s}}}\right) =$$

$$= f(\underline{x}_0) + \frac{\epsilon + r_k \cdot \lambda_{l_s} \cdot \sum_{i=1}^{r} - \ln\left(\frac{d_i^0}{G_i(\underline{x}_0)^{1-\lambda_{l_s}}}\right)}{\lambda_{l_s}}.$$

Minthogy fennállnak az alábbi egyenlőségek

$$\lim_{l_s \to \infty} \lambda_{l_s}(-\ln[d_i^0]) + \lim_{l_s \to \infty} \lambda_{l_s} \cdot \ln[G_i(\underline{x}_0)^{1-\lambda_{l_s}}] = 0, \qquad i = 1, \ldots, r$$

következik, hogy

$$\lim_{l_s \to \infty} P_2'(y_{l_s}, r_k) = +\infty.$$

Az eredetileg konkáv feltételekre hasonló becsléseket végezve adódik az állitás.

2. Tétel. *Ha a* (2)*-t tekintjük és* (3) *teljesül, valamint* R_0 *nem üres, akkor bármely* k *értékre* $P_2(\underline{x}, r_k)$ *felveszi a minimumát* R_0 *felett és* $\lim_{k \to \infty} P_2(\underline{x}(r_k), r_k) = \nu^*$ *ahol* $\underline{x}(r_k)$ $P_2(\underline{x}(r_k))$ *minimum helye* R_0*-on.*

Bizonyitás. 1, Legyen $\underline{x}_0 \epsilon R_0$ (ilyen pont létezik, mivel R_0 nem üres). Az $\{\underline{x} | P_2(\underline{x}, r_k) \leqslant P_2(\underline{x}_0, r_k), \underline{x} \epsilon R\}$ nem tartalmaz R-beli határpontot és zárt halmaz, amelynek $\underline{x}_0$ eleme. Elegendő belátni, hogy ez a halmaz korlátos bármely k érték esetén. Tegyük fel, hogy

$$\{\underline{x} | P_2(\underline{x}, r_k) \leqslant P_2(\underline{x}_0, r_k), \underline{x} \epsilon R\} = H_k$$

nem korlátos.

Ekkor található egy R_0-beli elemekből alkotott $\underline{y}_1, \underline{y}_2, \ldots$ nem korlátos sorozat. Az 5. Lemma bizonyitásában felhasznált állitás miatt, az előbbi sorozat egy $\underline{y}_{l_1}, \underline{y}_{l_2}, \ldots$ - részsorozatára

$$\lim_{l_s \to \infty} P_2(\underline{y}_{l_s}, r_k) = +\infty.$$

Ez ellentmond annak, hogy

$$P_2(\underline{y}_{l_s}, r_k) \leqslant P_2(\underline{x}_0, r_k), \qquad s = 1,2, \ldots$$

2. Mivel $P_2(\underline{x}(r_k), r_k)$ jelenti $P_2(\underline{x}, r_k)$-nak a minimum értékét R_0-on, ezért

$$f(\underline{x}(r_k)) + r_k I_2(\underline{x}(r_k)) \leqslant f(\underline{x}(r_{k+1})) + r_k I_2(\underline{x}(r_{k+1})),$$

$$f(\underline{x}(r_{k+1})) + r_{k+1} I_2(\underline{x}(r_{k+1})) \leqslant f(\underline{x}(r_k)) + r_{K+1} I_2(\underline{x}(r_K))$$

Szorozzuk be az első egyenlőtlenséget r_{k+1}/r_k -val és adjuk hozzá a másodikhoz. Átrendezés után kapjuk

$$(1 - \frac{r_{k+1}}{r_k}) f(\underline{x}(r_{k+1})) \leqslant (1 - \frac{r_{k+1}}{r_k}) f(\underline{x}(r_k)), \qquad \text{azaz}$$

$$f(\underline{x}(r_{k+1})) \leqslant f(\underline{x}, r_k)).$$

Ezt felhasználva, az első egyenlőtlenségből kapjuk

$$I_2(\underline{x}(r_k)) \leqslant I_2(\underline{x}(r_{k+1})).$$

Az $I_2(\underline{x}(r_k))$, $k = 1,2, \ldots$ sorozat tehát alulról korlátos, alsó korlátja $I_2(\underline{x}(r_1))$. Legyen $\epsilon > 0$. Ekkor az $f(\underline{x})$ konvexitása, a $G_i(\underline{x})$-k $(i = 1, \ldots, m)$ logaritmikus konkávitása miatt létezik $\underline{x}^* \epsilon R_0$ ugy, hogy $f(x^*) < v^* + \epsilon$.
De $I_2(\underline{x}(r_1)) \leqslant I_2(\underline{x}(r_k))$, $P_2(\underline{x}(r_k), r_k)$ minimum érték $k = 1,2, \ldots$ igy

$$v^* + r_k I_2(\underline{x}(r_1)) \leqslant P_2(\underline{x}(r_k), r_k) \leqslant f(\underline{x}^*) + r_k I_2(\underline{x}^*) \text{ minden } k\text{-ra.}$$

Tehát

$$\lim_{k \to \infty} (v^* + r_k \cdot I_2(\underline{x}(r_1))) = v^* + \lim_{k \to \infty} r_k \cdot I_2(\underline{x}(r_1)) = v^* \leqslant$$

$$\leqslant \lim_{k \to \infty} P_2(\underline{x}(r_K), r_K) \leqslant \overline{\lim_{k \to \infty}} P_2(\underline{x}(r_k), r_k) \leqslant \lim_{k \to \infty} (f(\underline{x}^*) + r_k I_2(\underline{x}^*)) =$$

$$= f(\underline{x}^*) + \lim_{k \to \infty} r_k I_2(\underline{x}^*) = f(\underline{x}^*) < v^* + \epsilon.$$

Mivel ϵ tetszőlegesen kicsiny lehet, ez azt jelenti, hogy a $P_2(\underline{x}(r_k),r_k)$ limes superiorja és limes inferioja megegyezik, tehát a $P_2(\underline{x}(r_k),r_k)$, $k = 1,2,\ldots$ konvergens és

$$\lim_{k\to\infty} P_2(\underline{x}(r_k),r_k) = v^* \ , \quad \text{ami éppen az állitás.}$$

Ebben az esetben is teljesülnek:

1. $$\lim_{k\to\infty} r_k \cdot \sum_{i=1}^{m} - \ln G_i(\underline{x}(r_k)) = 0,$$

2. $$\lim_{k\to\infty} f(\underline{x}(r_k)) = v^* \ ,$$

3. az $\underline{x}(r_k)$ sorozat bármely konvergens részsorozatának határértéke a (2) optimum helye.

Bizonyitás. Az $I_2(\underline{x}(r_k))$, $k = 1,2,\ldots$ alulról korlátos sorozat, ezért 1, igaz. Az 1-ből következik a 2. állitás. A 3. állitás bizonyitása teljesen megegyezik a korábbi ilyen állitás bizonyitásával.

A most következő részben a számitástechnikai tapasztalatokat ismertetjük. Egy matematikai programozási feladatot megoldó algoritmusról ugyanis csak ugy dönthetjük el, hogy jó-e vagy sem, ha elkészitjük a gépi programot és segitségével kisérleti feladatokat oldunk meg. Az elméleti algoritmus ismerete általában nem ad felvilágositást a konvergencia sebességéről, a számitáshoz szükséges időről, másrészt a gépi reprezentálásnál adódó ötletek lényegesen hatékonyabbá tehetik az eljárást.

Itt az alábbi feladat megoldását ismertetjük.

$$\min (x_1 + x_2),$$

$$P\left\{\begin{array}{l} 3x_1 + x_2 - 6 \geqslant \beta_1 \\ x_1 + 8x_2 - 8 \geqslant \beta_2 \end{array}\right\} \geqslant 0.8$$

$$x_1 + 4x_2 \geqslant 4,$$
$$3x_1 + x_2 \geqslant 3,$$
$$x_1 \geqslant 0, \quad x_2 \geqslant 0.$$

A (3)-ban β_1 és β_2 együttes eloszlása normális, nulla a várható értékük, 1 a szórásuk és 0.2 a korrelációs együtthatójuk. A valószinüségi feltétel (x_1,x_2)-nek logaritmikusan konkáv függvénye. Ilyen tipusu feladatok Prékopa A. munkáiban fordulnak elő. [9], [10], [11].

A SUMT algoritmust a következőképpen hajtjuk végre:

Képezzük a $P_1(\underline{x},r)$, $P_2(\underline{x},r)$ függvényeket, s ezután egy olyan $\underline{x}_0$-ból kiindulva, ahol minden feltétel éles egyenlőtlenséggel teljesül, valamilyen r_1 érték mellett minimalizáljuk a $P_1(\underline{x},r_1)$-t vagy $P_2(\underline{x},r_2)$-t. Az igy kapott minimum pontból kiindulva valamilyen $r_2 < r_1$ mellett ujra minimalizáljuk a $P_1(\underline{x},r_2)$-t vagy $P_2(\underline{x},r_2)$-t. Az eljárást addig folytatjuk, amig

az optimumhoz elég közel nem kerülünk..

A konvex programozási feladatoknál az eljárás pontossága elsősorban a feltétel nélkül minimalizáló rutin pontosságtól függ, ugyanis igazak az alábbi becslések.

A $P_1(\underline{x},r_k)$ függvényeknél

$$f(\underline{x}_i) - v^* \leqslant r_i \sum_{i=1}^{m} \frac{1}{g_i(\underline{x}_i)} ,$$

a $P_2(\underline{x},r_k)$ függvényeknél pedig

$$f(\underline{x}_i) - v^* \leqslant r_i \cdot m,$$

ahol v^* a konvex programozási probléma optimuma, m a feltételek száma, $\underline{x}_i$ pedig az i-edik feltétel nélküli minimalizálásnál kapott minimum hely

A feltétel nélküli minimalizálásnál gradiensmentes módszert – Hooke és Jeaves módszerét [7] – választottuk, mivel ilyen tipusu feladatokban nem mindig tudunk gradienst számolni. A két dimenziós normális eloszlás értékét a Deák I. [1] által irt szubrutin számolta.

A számitási eredmények azt mutatták, hogy a legjobb eredmény akkor adódik, ha az algoritmusban a logaritmikus büntetőfüggvénnyel számoltunk. Megvizsgáltuk, hogy ennél a problémánál az eljárás hogyan függött az r_1, illetve az r_k, $k = 2,3, \ldots$ paraméterek választásától. Az r_1 érték választása azért problematikus, mert ha az optimum a megengedett tartomány határán van (ami gyakran előfordul) és az r_1 érték nagy, akkor a minimalizáló eljárás során csak nagyon lassan jutunk az optimumig. Ennek oka az, hogy már egészen messze a határtól elkezdjük büntetni a célfüggvényt. Ha viszont az r_1 érték kicsi, akkor a büntető tag hatástalanná válik, a minimalizáló eljárás során gyakran kilépünk a megengedett tartományból és nem találjuk meg az optimumot. Ha $r_1 = 1$ volt és $r_{k+1} = r_k/5$, $k = 2,3, \ldots$ akkor helyes eredményt kaptunk. Ha $r_1 = 0.01$ vagy $r_1 = 0.001$ volt, akkor az algoritmus nem találta meg az optimumot. Amennyiben $r_{k+1} = r_{k/2}$, $k = 2,3, \ldots$ volt, az eljárás lassubb és pontatlanabb lett.

Az algoritmussal a kisérleti feladatot 40 mp alatt oldottuk meg. Az optimum értéke 2.934, az optimum hely koordinátái 1.984 és 0.95 voltak.

A feladatot a megengedett irányok módszerével is megoldották. [1]. Az itt közölt optimum valamivel pontosabb és a számolás lényegesen rövidebb ideig tartott. (Ott a feladat megoldása kb. 5 percet vett igénybe)

A feladatban a nem lineáris feltétel által meghatározott térrész benne van a lineáris feltételek által adott poliéderben. Ha a lineáris feltételeket nem büntettük, akkor is helyes eredményt kaptunk, de az eljárás több időt vett igénybe.

Irodalom

[1] Deák I., Egy sztochasztikus programozási modell számítógépes kiértékelése, MTA Számítástechnikai Központja Közlemények, 1972 dec.

[2] Fiacco, A.V. and G.P. McCormick, Computational algorithm for the sequential unconstrained minimization technique for nonlinear programming. Management Science 10, (4), 1964.

[3] Fiacco, A.V. and G.P. McCormick, The sequential unconstrained minimization technique for nonlinear programming. A primal-dual method Management Science, 10, (2), 1964.

[4] Fiacco, A,V. and G.P. McCormick, Extensions of SUMT for nonlinear programming equality constrains and extrapolition, Managemant Science, 12 (11), 1966.

[5] Fiacco, A.V. and G.P. McCormick, Nonlinear programming: Sequential unconstrained minimization technique, Wiley, New York. London, 1968.

[6] Himmelblau, D.M. A uniform evaluation of unconstrained optimization techniques, Numerical methods for nonlinear optimization, ed. by Lootsma, F.A. London, New York, Academie Press, 1972.

[7] Kovalik, J. and M.R. Osborne, Methods for unconstrained optimization problems, Elsevier, New York, 1968.

[8] Krekó B., Optimumszámitás, Közgazdasági és jogi könyvkiadó, Budapest, 1972.

[9] Prékopa A., Logarithmic concave measures with application to stochastic programming, Acta Math. Szeged, 32, 1971.

[10] Prékopa A., Stochastic programming models for inventory control and water storage problems. Proc. of the Conf. on Inv. Cont. and Water Storage Probl. Győr, 1971. Sept. Bolyai János Math. Soc. North Holland Publ. Co. 1973.

[11] Prékopa A., A class of stochastic programming decision problems, Math. Op. Forsch. und Stat. 3, 1972, 349-354.

Summary

Application of the SUMT method for mathematical programming problems containing logarithmically concave constraint functions

Tamás Rapcsák

It is shown in this paper that the algorithms of the SUMT interior point define, under logarithmically concave constraint functions, even in case of non-bounded domain a global optimum. Tasks of such type can be found in the works of A. Prékopa. Also the proofs of the convergence theorems discussed in the paper are different from the earlier ones. At the end of the study experiences in computing are published.

РЕЗЮМЕ

Применение метода штрафных функций для решения задач нелинейного программирования содержащих среди условий логаритмически вогнутые функции.

Тамаш Рапчак

В этой статье рассматриваются алгоритмы внутреннего пункта СУМТ при наличии логарифмически конкавных условных функций, а также в случае неограниченной области, определяется глобальный оптимум. Задачи такого типа встречаются в работах А. Прекопа. Доказательства теорем сходимости фигурирующих в статье тоже отличаются от известных до сих пор. В конце статьи приведены решения практических задач на ЭВМ.

Ez a cikk beérkezett: 1976. januárjában.

MTA Számítástechnikai és Automatizálási Kutató Intézete, **Közlemények** 19/1978.

EGY TÁBLÁZATKITÖLTÉSI KOMBINATÓRIKAI PROBLÉMA ÉS KÜLÖNFÉLE TÉMÁKHOZ KAPCSOLÓDÓ EKVIVALENS VÁLTOZATAI

Kéri Gerzson

Bevezetés

Nagyméretű lineáris programozási feladatok megoldására szolgáló programrendszerek legkritikusabb része az ujrainvertáló algoritmus. Elsősorban ezen múlik, hogy egy lineáris programozási rendszer mennyire használható a gyakorlatban akár a számítások pontossága, akár a futási idő hossza szempontjából. Az idevágó nemzetközi számítástechnikai kutatások és tapasztalatok szerint a lineáris programozás számára legalkalmasabb ujrainvertálási eljárás a mátrixinverz eliminációs alakjának előállítása valamely jó pivot választási szabállyal. Azt a kérdést, hogy melyik a legjobb pivot választási szabály, meggyőzően csak további számitógépes tapasztalatok dönthetik el. Ezenkivül bármikor felbukkanhatnak ujabb pivot választási szabályra vonatkozó ötletek. E dolgozat szerzőjét egy ilyen ötlet vezette a dolgozat tárgyát képező kombinatórikai problémák vizsgálatára. A [4] előadás szándéka volt ismertetni és több oldalról megvilágitani a problémák eredetét, e dolgozat szándéka pedig a problémák vizsgálata során kapott eddigi eredmények részletes leirása.

1. Egy táblázatkitöltési probléma megfogalmazása

Legyen adott egy m sorból, n oszlopból és $m \cdot n$ cellából álló téglalap alakú táblázat, továbbá az $a_i (i = 1,2, \ldots, m)$ és $b_j (j = 1,2, \ldots, n)$ egész számok, melyekre teljesül, hogy

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j. \tag{1.1}$$

Feltesszük a következő kérdést: Hány féleképpen tölthető ki ez a táblázat két fajta elemmel, mondjuk A-elemmel és B-elemmel, oly módon, hogy a táblázat minden cellájába vagy egy A-elemet vagy egy B-elemet helyezünk ("kizáró vagy"értelemben), a táblázat i-edik sorának celláiba összesen a_i számú A-elemet és $n - a_i$ számú B-elemet helyezünk, a táblázat j-edik oszlopának celláiba összesen b_j számú A-elemet és $m - b_j$ számú B-elemet helyezünk minden $i = 1,2, \ldots, m$, illetve $j = 1,2, \ldots, n$ esetén.

Vezessük be a fenti követelményeknek eleget tevő kitöltések számának jelölésére az

$$\alpha^{(m,n)}(a_1, a_2, \ldots a_m; b_1, b_2, \ldots b_n)$$

függvényt.

A következő három szakaszban megmutatjuk, hogy a fent megfogalmazott kombinatórikai probléma (a továbbiakban alapprobléma) a matematika több ágával is kapcsolatban van.

2. A szállítási probléma 0 – 1 megoldásainak (a Ryser-féle piknik-modell megoldásainak) a száma

Rendeljük hozzá az 1. szakaszban szereplő táblázat (i,j) cellájához az $x_{i,j}$ változót, melynek értéke legyen 1, ha az (i,j) cellába A-elemet helyezünk, és legyen 0, ha az (i,j) cellába B-elemet helyezünk. Azt a követelményt, hogy a táblázat i-edik sorának celláiba a_i számú A-elemet helyezünk, a

$$\sum_{j=1}^{n} x_{ij} = a_i$$

egyenlőség fejezi ki. Hasonlóan azt a követelményt, hogy a j-edik oszlop celláiba b_j számú A-elemet helyezünk, a

$$\sum_{i=1}^{m} x_{ij} = b_j$$

egyenlőség fejezi ki. Az $\alpha^{(m,n)}(a_1, a_2, \ldots a_m; b_1, b_2, \ldots, b_n)$ függvény értéke tehát nem más, mint a

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (i = 1,2, \ldots, m) \tag{2.1}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad (j = 1,2, \ldots, n) \tag{2.2}$$

$$x_{ij} = 0 \text{ vagy } 1 \qquad (i = 1,2, \ldots, m; \quad j = 1,2, \ldots, n) \tag{2.3}$$

feltételrendszer különböző megoldásainak a száma. Ha (2.3) helyett csak azt kötnénk ki, hogy $x_{ij} \geqslant 0$ legyen minden (i,j) párra, akkor itt a szállítási probléma feltételrendszere állna. Az $\alpha^{(m,n)}$ függvény értéke tehát kifejezi a szállítási probléma 0 – 1 megoldásainak a számát.

Nemnegativ a_i, b_j - k esetén a (2.1) – (2.3) rendszer azonos a Ryser-féle piknik-modell feltételrendszerével. Az $\alpha^{(m,n)}$ függvényt is értelmezhetjük a piknik-modell alapján a következőképpen.

Egy helyiségben, amelyben m számú asztal van, sorozatos összejöveteleket akar megvalósitani n számú család, a következő feltételekkel: Az i-edik asztal férőhelyeinek száma $a_i (i = 1,2, \ldots, m)$, a j-edik család tagjainak száma $b_j (j = 1,2, \ldots, n)$, és teljesül (1.1). Minden összejövetel esetén a családok tagjai úgy helyezkedjenek el, hogy egy asztalnál egy családnak legfeljebb egy tagja üljön. További követelmény, hogy minden összejövetelnél az elhelyezés más-más ülésrend szerint történjék. Két összejövetel ülésrendjét csak akkor tekintjük különbözőnek, ha van olyan asztal és család, hogy az egyik összejövetelnél e család képviselve van valamelyik tagjával a szóbanforgó asztalnál, a másik összejövetelnél pedig nem.

A probléma az, hogy legfeljebb hány összejövetel valósulhat meg a fenti követelmények figyelembevételével. Ez pedig ekvivalens azzal, hogy hány különböző megoldása van a (2.1) – (2.3 rendszernek.

3. Adott fokú csúcsokkal rendelkező páros gráfok száma

Rendeljünk hozzá az 1. szakaszban szereplő táblázatkitöltési probléma (alapprobléma) minden megoldásához egy páros gráfot, melynek csúcsai a táblázat sorai $(R_1, R_2, \ldots, R_m)$ és oszlopai $(C_1, C_2, \ldots, C_n)$, melynek csak olyan élei vannak, amelyek valamelyik R_i- t valamelyik C_j-vel kötik össze, és melynek az R_i és C_j csúcsa között akkor és csak akkor halad él, ha a táblázat (i,j) cellája A-elemet tartalmaz.

Az a követelmény, hogy a táblázat i-edik sorának celláiba a_i számú A-elemet, illetve a táblázat j-edik oszlopának celláiba b_j számú A-elemet helyezünk, azt jelenti, hogy a megoldáshoz rendelt gráf R_i csúcsának a foka a_i, illetve C_j csúcsának a foka b_j. Az $\alpha^{(m,n)}(a_1, a_2, \ldots, a_m;\ b_1, b_2, \ldots b_n)$ függvényérték tehát megadja azoknak a páros gráfoknak a számát, melynek csúcsai $R_1, R_2, \ldots, R_m$, $C_1, C_2, \ldots C_n$, minden élük az $R_1, R_2, \ldots R_m$ csúcsok valamelyikét a $C_1, C_2, \ldots C_n$ csúcsok valamelyikével köti össze, az R_i csúcs foka $a_i (i = 1,2, \ldots, m)$, a C_j csúcs foka pedig $b_j (j = 1,2, \ldots, n)$.

4. Szimmetrikus polinomok együtthatói

Itt azt a kérdést vizsgáljuk, hogy az $y_1, y_2, \ldots, y_n$ változók elemi szimmetrikus polinomjai szorzataként adódó polinomokban mi az

$$y_1^{b_1} y_2^{b_2} \ldots y_n^{b_n}$$ tag együtthatója.

Az $y_1, y_2, \ldots, y_n$ változók elemi szimmetrikus polinomjai alatt a

$$\sigma_0 = 1$$

$$\sigma_1 = y_1 + y_2 + \ldots + y_n$$

$$\sigma_2 = y_1 y_2 + y_1 y_3 + \ldots + y_{n-1} y_n$$

$$\vdots$$

$$\sigma_k = \sum y_{i_1} y_{i_2} \ldots y_{i_k}$$

$$\vdots$$

$$\sigma_n = y_1 y_2 \ldots y_n$$

kifejezéseket értjük.

Azt állitjuk, hogy ha (1.1) teljesül, ezenkivül $a_1, a_2, \ldots a_m$ és $b_1, b_2, \ldots b_n$ mind nemnegativ egészek, akkor a

$$\prod_{i=1}^{m} \sigma_{\alpha_i} = \sigma_{a_1} \sigma_{a_2} \ldots \sigma_{a_m}$$

polinomban $y_1^{b_1} y_2^{b_2} \dots y_n^{b_n}$ együtthatója éppen $\alpha^{(m,n)} \quad (a_1, a_2, \dots a_m ; b_1, b_2, \dots b_n)$. Mivel ez az állitás nem teljesen nyilvánvaló, kimondunk és bizonyitunk egy tételt, melyből az állitásunk már egyszerüen következik.

Jelöljük a $\prod_{i=1}^{m} \sigma_{a_i}$ szorzat i-edik tényezőjeként szereplő σ_{a_i} polinom tagjait z_{ik}-val $(i = 1,2, \dots, m; \quad k = 1,2, \dots, \binom{n}{a_i})$. Ekkor irhatjuk, hogy

$$(4.1) \qquad \prod_{i=1}^{m} \sigma_{a_i} = \sum_{k_1=1}^{\binom{n}{a_1}} \sum_{k_2=1}^{\binom{n}{a_2}} \dots \sum_{k_m=1}^{\binom{n}{a_m}} \left(\prod_{i=1}^{m} z_{ik_i} \right).$$

4.1. **Tétel.** *Tegyük fel, hogy* $a_1, a_2, \dots, a_m$, $b_1, b_2, \dots, b_n$ *nemnegativ egészek, melyekre* (1.1) *teljesül. Ekkor a* (4.1) *jobboldalán álló* $\prod_{i=1}^{m} z_{ik_i}$ *tagok között az* $y_1^{b_1} y_2^{b_2} \dots y_n^{b_n}$ *kifejezés* $\alpha^{(m,n)}(a_1, a_2, \dots, a_m ; b_1, b_2, \dots, b_n)$ *multiplicitással szerepel.*

Bizonyitás. Mivel a σ_k polinom minden tagja k-adfokú, ezért a (4.1) jobboldalán álló polinom minden tagja $\sum_{i=1}^{m} a_i$ fokú, tehát minden egyes $\prod_{i=1}^{m} z_{ik_i}$ tag $y_1^{b_1} y_2^{b_2} \dots y_n^{b_n}$ alakú, ahol $\sum_{j=1}^{n} b_j = \sum_{i=1}^{m} a_i$.

A (2.1), (2.3) feltételrendszer megoldásaihoz kölcsönösen egyértelmüen hozzárendelhetők a (4.1) jobboldalán álló összeg tagjai a következőképpen.

Minden $i = 1,2, \dots, m$-re az $x_{i1}, x_{i2}, \dots, x_{im}$ változók közül a_i számú változónak 1, a többinek pedig 0 az értéke. Legyenek $r_{i1}, r_{i2}, \dots, r_{ia_i}$ az i első indexhez tartozó 1 értékü x_{ij}-k második indexei, és képezzük az $y_{r_{i1}} y_{r_{i2}} \dots y_{r_{ia_i}}$ szorzatot, amely a σ_{a_i} polinom egy tagja, és a (2.1), (2.3) feltételrendszer szóbanforgó megoldásához rendeljük hozzá a

$$(4.2) \qquad \prod_{i=1}^{m} (y_{r_{i1}} y_{r_{i2}} \dots y_{r_{ia_i}})$$

szorzatot, amely tehát a (4.1) jobboldalán álló összeg egy tagja. A hozzárendelés egyértelmü, mert a (2.1), (2.3) rendszer két különböző megoldása esetén legalább egy i indexre eltérő az $x_{i1}, x_{i2}, \dots, x_{im}$ változók értékrendszere, tehát az ezek segitségével képzett $y_{r_{i1}} y_{r_{i2}} \dots y_{r_{ia_i}}$ szorzat a két különböző megoldás esetén a σ_{a_i} polinom más-más tagja, és igy a két különböző megoldáshoz a (4.1) jobboldalán álló összeg két különböző tagját rendeljük.

Fordítva: (4.1) jobboldalának tetszőleges $\prod_{i=1}^{m} z_{ik_i}$ tagjában szereplő mindcn z_{ik_i} egyértelműen irható $y_{r_{i1}} y_{r_{i2}} \ldots y_{r_{ia_i}}$ alakban. Ilymódon minden egyes $\prod_{i=1}^{m} z_{ik_i}$ taghoz egyértelműen rendeljük az

$$
\begin{aligned}
&r_{11}, r_{12}, \ldots, r_{1a_1},\\
&r_{21}, r_{22}, \ldots, r_{2a_2},\\
&\cdot\\
&\cdot\\
&\cdot\\
&r_{m1}, r_{m2}, \ldots, r_{ma_m}
\end{aligned}
$$

indextáblázatot, ehhez pedig a (2.1), (2.3) feltételrendszernek azt a megoldását, amelyre minden i esetén

$$
x_{ij} = \begin{cases} 1 \text{ ha } j \text{ az } r_{i1}, r_{i2}, \ldots, r_{ia_i} \text{ indexek közül való} \\ 0 \text{ egyébként.} \end{cases}
$$

A (2.1), (2.3) feltételrendszer bármely megoldásához rendelt (4.2) szorzatban valamely y_j annyiszor szerepel tényezőként, ahány különböző i indexre teljesül $x_{ij} = 1$, tehát a (4.2) szorzatban y_j előfordulásainak a száma

$\sum_{i=1}^{m} x_{ij}$ minden $j = 1,2, \ldots, n$ esetén. Ez azt jelenti, hogy a (2.1), (2.3) feltételrendszer valamely $x_{ij} (i = 1,2, \ldots, m; j = 1,2, \ldots, n)$ megoldása akkor és csak akkor elégiti ki (2.2)-t is, ha a (4.1) jobboldalán álló összegnek a szóbanforgó megoldáshoz rendelt tagjában, azaz a (4.2) szorzatban y_j előfordulásainak a száma b_j minden $j = 1,2, \ldots, n$ esetén. Mivel a hozzárendelés kölcsönösen egyértelmü, az eddigiekből következik, hogy (4.1) jobboldalának tagjai között annyiszor szerepel az

$$
y_1^{b_1} y_2^{b_2} \ldots y_n^{b_n}
$$

kifejezés, ahány különböző megoldása van a (2.1) – (2.3) feltételrendszernek. A 2. szakaszban megmutattuk, hogy az utóbbiak száma $\alpha^{(m,n)}(a_1, a_2, \ldots, a_m; b_1, b_2, \ldots, b_n)$, és igy a tétel bizonyitásával készen vagyunk.

5. Az alapproblémával kapcsolatos két további függvény

Az $\alpha^{(m,n)}$ függvény értéke nyilvánvalóan 0 az alábbi esetekben:

(5.1) $a_i < 0$ valamely i-re,

(5.2) $a_i > n$ valamely i-re

(5.3) $b_j < 0$ valamely j-re,

(5.4) $b_j > m$ valamely j-re.

Igy lényegében nem szükitettük a problémát azzal, hogy a 4. szakaszban nemnegativ a_i és b_j értékekre szoritkoztunk.

Az is nyilvánvaló, hogy az $\alpha^{(m,n)}$ függvény értéke nem változik, ha akár az $a_1, a_2, \ldots, a_m$ argumentumok értékét, akár a $b_1, b_2, \ldots, b_n$ argumentumok értékét tetszőlegesen permutáljuk. Amennyiben az (5.1) – (5.4) eseteket kizárjuk, $\alpha^{(m,n)}$ értékét egyértelmüen meghatározza az az információ, ha tudjuk, hogy az $a_1, a_2, \ldots, a_m$ argumentumok értékei között hány $0, 1, 2, \ldots, n$ érték szerepel, a $b_1, b_2, \ldots, b_n$ argumentumok értékei között hány $0, 1, 2, \ldots, m$ érték szerepel. Jelöljük s_i-vel az $a_1, a_2, \ldots, a_m$ értékek között található i-vel azonos értékek számát, t_j-vel a $b_1, b_2, \ldots, b_n$ értékek között található j-vel azonos értékek számát, $\omega^{(m,n)}(s_0, s_1, \ldots, s_n; t_0, t_1, \ldots, t_m)$-mel pedig az $\alpha^{(m,n)}(a_1, a_2, \ldots, a_m; b_1, b_2, \ldots, b_n)$ függvénynek az s_i, t_j argumentumokkal kifejezett variánsát.

Az $\omega^{(m,n)}$ függvényt olyan s_i, t_j argumentumokra értelmezzük, melyekre minden s_i és minden t_j nemnegativ egész, továbbá teljesülnek a következők.

$$(5.5) \qquad \sum_{i=0}^{n} s_i = m$$

$$(5.6) \qquad \sum_{j=0}^{m} t_j = n$$

$$(5.7) \qquad \sum_{i=1}^{n} i s_i = \sum_{j=1}^{m} j t_j$$

A soron következő $\mu^{(m,n)}$ függvény szerepeltetése középutat jelent az $\alpha^{(m,n)}$ és $\omega^{(m,n)}$ függvény értelmezése között. Az a_i mennyiségek nemnegativitásának feltételezése esetén az $\alpha^{(m,n)}$ függvénynél szereplő $a_1, a_2, \ldots, a_m$ argumentumcsoport helyett az $s_0, s_1, \ldots, s_n$ argumentumcsoportot szerepeltetve, a $b_1, b_2, \ldots, b_n$ argumentumcsoportot azonban meghagyva, az $\alpha^{(m,n)}$ függvény a

$$\mu^{(m,n)}(s_0, s_1, \ldots, s_n; b_1, b_2, \ldots, b_n)$$

függvénnyé alakul át, melyet olyan s_i, b_j argumentumokra értelmezünk, amelyekre minden s_i nemnegativ egész, minden b_j egész, továbbá teljesülnek a következők.

(5.8) $$\sum_{i=0}^{n} s_i = m$$

(5.9) $$\sum_{i=1}^{n} i s_i = \sum_{j=1}^{n} b_j .$$

Az 1 – 4. szakaszokban ismertetett problémák megoldása történhet akár az $\alpha^{(m,n)}$, akár az $\omega^{(m,n)}$, akár a $\mu^{(m,n)}$ függvény egy értékének meghatározása utján. Látni fogjuk, hogy az $\omega^{(m,n)}$ függvény használata az $\alpha^{(m,n)}$ függvény helyett lényegesen csökkentheti a megoldás alapjául szolgáló rekurziós formulákban szereplő tagok számát, azonban kevésbé áttekinthetővé teszi a formulákat. A $\mu^{(m,n)}$ függvény egyesiti az $\alpha^{(m,n)}$ függvény és az $\omega^{(m,n)}$ függvény előnyeit, kiküszöböli mindkettő hátrányát, ezért használata a szimmetria elromlása ellenére is bizonyos esetekben jó kompromisszumnak fog bizonyulni.

Végül megfogalmazzuk a 4.1. tételnek a $\mu^{(m,n)}$ és $\omega^{(m,n)}$ függvényekre vonatkozó változatát.

5.1. **Tétel.**

a.) *Ha* s_i *nemnegativ egész* $i = 0,1, \ldots, n$ *esetén,* b_j *egész* $j = 1,2, \ldots, n$ *esetén, továbbá teljesül* (5.8) *és* (5.9) *akkor a*

$$\sigma_0^{s_0} \sigma_1^{s_1} \ldots \sigma_n^{s_n}$$

polinomban

$$y_1^{b_1} y_2^{b_2} \ldots y_n^{b_n}$$

együtthatója

$$\mu^{(m,n)}(s_0, s_1, \ldots, s_n;\ b_1, b_2, \ldots, b_n).$$

b.) *Ha* s_i *nemnegativ egész* $i = 0,1, \ldots, n$ *esetén,* t_j *nemnegativ egész* $j = 0,1, \ldots, m$ *esetén, továbbá teljesül* (5.5) – (5.7), *akkor a*

$$\sigma_0^{s_0} \sigma_1^{s_1} \ldots \sigma_n^{s_n}$$

polinomban

$$\omega^{(m,n)}(s_0, s_1, \ldots, s_n;\ t_0, t_1, \ldots, t_m)$$

az együtthatója minden olyan

$$y_1^{b_1} y_2^{b_2} \ldots y_n^{b_n}$$

tagnak, melyben a $b_1, b_2, \ldots, b_n$ *kitevők értékei között* s_0 *számú* 0, s_1 *számú* $1, \ldots, s_n$ *számú* n *érték szerepel.*

Bizonyitás. Mindkét állitás következik a 4.1. tétel állitásából, ha a

$$\sigma_0^{s_0}\ \sigma_1^{s_1}\ \ldots\ \sigma_n^{s_n}$$

szorzatban minden egyes $\sigma_j^{s_j}$ hatványt s_j számú σ_j tényezővel helyettesitünk.

6. Az alapproblémával kapcsolatos egzisztenciatételek

Az 5. szakasz elején tett megjegyzés szerint $0 \leqslant a_i \leqslant n$ $(i = 1,2, \ldots, m)$, $0 \leqslant b_j \leqslant m$ $(j = 1,2, \ldots, n)$ szükséges (de nem elégséges) feltétel arra, hogy az $\alpha^{(m,n)}$ függvény értéke pozitiv legyen; hogy létezzék az 1. szakaszban leirt követelményeknek eleget tevő táblázatkitöltés; hogy a (2.1) – (2.3) feltételrendszernek legyen megoldása; hogy létezzék olyan a 3. szakaszban leirt páros gráf, melyben a csúcsok foka $a_1, a_2, \ldots, a_m$, illetve $b_1, b_2, \ldots, b_n$; hogy a

$$\prod_{i=1}^{m} \sigma_{a_i} \text{ polinomban } \prod_{j=1}^{n} y_j^{b_j}$$

pozitiv együtthatóval szerepeljen. Mindezekre egy szükséges és egyúttal elégséges feltételt ad a következő, Gale-től [1] és Ryser-tól [3] származó tétel, melynek bizonyitása [2]-ben is megtalálható.

6.1. **Tétel.** *Tegyük fel, hogy* a_i, b_j *nemnegativ egészek* $(i = 1,2, \ldots, m;\ j = 1,2, \ldots, n)$ *melyekre* (1.1) *teljesül, és* $b_1 \geqslant b_2 \geqslant \ldots \geqslant b_n$. *Ebben az esetben a* (2.1) – (2.3) *rendszernek akkor és csak akkor van megoldása, ha*

$$(6.1) \qquad \sum_{l=1}^{j} b_l \leqslant \sum_{i=1}^{m} \min(a_i, j)$$

fennáll $j = 1,2, \ldots, n-1$ *esetén.*

A fenti tétel állitása nyilván úgy is fogalmazható, hogy az előzetes feltételek fennállása esetén $\alpha^{(m,n)}(a_1, a_2, \ldots, a_m;\ b_1, b_2, \ldots, b_n) > 0$ akkor és csak akkor teljesül, ha (6.1) fennáll $j = 1,2, \ldots, n-1$ esetén. Most még kimondjuk a fenti tételnek a $\mu^{(m,n)}$ függvényre vonatkozó következményét.

6.2. **Tétel.** *Tegyük fel, hogy* s_i, b_j *nemnegativ egészek* $(i = 0,1, \ldots, n;\ j = 1,2, \ldots, n)$ *teljesül* (5.8) *és* (5.9), *továbbá* $b_1 \geqslant b_2 \geqslant \ldots \geqslant b_n$. *Ebben az esetben*

$$\mu^{(m,n)}(s_0, s_1, \ldots, s_n;\ b_1, b_2, \ldots, b_n) > 0$$

akkor és csak akkor áll fenn, ha

$$(6.2) \qquad \sum_{l=j+1}^{n} (l-j) \cdot s_l \leqslant \sum_{l=j+1}^{n} b_l$$

teljesül $j = 1,2, \ldots, n-1$ esetén.

Bizonyitás. A $\mu^{(m,n)}$ függvénynek az 5. szakaszban történt származtatása szerint s_l jelenti az $a_1, a_2, \ldots, a_m$ értékek között szereplő l-lel azonos értékek számát, ezért a (6.1) egyenlőtlenség jobboldala a következőképpen alakitható.

$$\sum_{i=1}^{m} \min(a_i, j) = \sum_{l=0}^{n} s_l \cdot \min(l, j) =$$

$$= \sum_{l=0}^{j} l s_l + \sum_{l=j+1}^{n} j s_l = \sum_{l=1}^{n} l s_l + \sum_{l=j+1}^{n} (j - l) \cdot s_l = \tag{6.3}$$

$$= \sum_{l=1}^{n} b_l + \sum_{l=j+1}^{n} (j - l) \cdot s_l.$$

Az utolsó átalakitás (5.9) felhasználásával történt. Azt kaptuk, hogy (6.1) ekvivalens a

$$\sum_{l=1}^{j} b_l \leqslant \sum_{l=1}^{n} b_l + \sum_{l=j+1}^{n} (j - l) \cdot s_l$$

egyenlőtlenséggel ebből pedig átrendezéssel adódik a (6.2) egyenlőtlenség. Igy a 6.2 tétel állitását igazoltuk.

A b_j értékeknek a 6.1. és 6.2. tételekben kikötött monotonitása nem szükiti az általánosságot, mivel sem az $\alpha^{(m,n)}$ függvény értéke, sem pedig az $\omega^{(m,n)}$ függvény értéke nem változik, ha a $b_1, b_2, \ldots, b_n$ argumentumok értékét tetszőlegesen permutáljuk.

Végül a 6.1. és 6.2. tételek egy kiegészitését bizonyitjuk. Nevezetesen azt, hogy ha minden $j = 1,2, \ldots, n-1$ esetén (6.1)-ben illetve (6.2)-ben egyenlőség áll fenn, akkor

$$\alpha^{(m,n)}(a_1, a_2, \ldots, a_m;\ b_1, b_2, \ldots, b_n) = 1 \qquad \text{illetve}$$

$$\mu^{(m,n)}(s_0, s_1, \ldots, s_n;\ b_1, b_2, \ldots, b_n) = 1, \qquad \text{azaz a}$$

(2.1) – (2.3) rendszernek egyetlen megoldása van.

Az (5.9) és (6.2) egyenlőségekből a b_j-k egyértelmüen adódnak:

$$b_j = \sum_{l=j}^{n} s_l \qquad (j = 1,2, \ldots, n \quad \text{esetén}) \tag{6.4}$$

Most a (2.1) – (2.3) rendszer megoldását az x_{ij} változók meghatározásának az alábbi algoritmus szerinti sorrendjében végezve könnyen látható, hogy a (2.1) – (2.3) rendszernek egyetlen megoldása van.

Első lépés ($l = 1$): $x_{ij} = 0$ az s_0 számú azon i indexre, melyekre $a_i = 0$ és tetszőleges j indexre. $x_{i1} = 1$ mindazon i indexekre, melyekre $a_i \neq 0$. (s_0 számú i indexre már $x_{ij} = 0$ és (6.4) szerint $b_1 = \sum_{l=1}^{n} s_l = m - s_0$.) Legyen $l = 2$ és folytassuk az algoritmust az l-edik lépésnél.

Az l-edik lépés: $x_{ij} = 0$ az s_{l-1} számú azon i indexre, melyekre $a_i = l - 1$, és tetszőleges l-nél nem kisebb j indexre. (Az ilyen i indexek esetén már $x_{i1} = x_{i2} = \ldots = x_{i(l-1)} = 1$) $x_{il} = 1$ mindazon i indexekre, melyekre $a_i \geqslant l$. ($s_0 + s_1 + \ldots + s_{l-1}$ számú i indexre már $x_{il} = 0$ és (6.4) szerint

$$b_l = s_l + s_{l+1} + \ldots + s_n = m - (s_0 + s_1 + \ldots + s_{l-1}).)$$

Ha most $l = n$, akkor az algoritmus véget ér, ellenkező esetben 1-gyel növeljük l értékét, és a megnövelt értékkel újra végrehajtjuk az l-edik lépést.

Az algoritmus menete mutatja, hogy a (2.1) – (2.3) rendszernek nincs más megoldása azon kivül, amelyet az algoritmus előállit. Könnyen verifikálható, hogy az algoritmussal előállitott x_{ij}-k valóban megoldását adják a (2.1) – (2.3) rendszernek.

7. Az alapprobléma generátorfüggvénye

A táblázatkitöltési alapprobléma generátorfüggvényének nevezzük és

$G^{(m,n)}(y_1, y_2, \ldots, y_m;\ z_1, z_2, \ldots, z_n)$-nel jelöljük

azt a kifejezést, amely úgy adódik, hogy összegezzük a $0 \leqslant a_i \leqslant n$ $(i = 1,2, \ldots, m)$, $0 \leqslant b_j \leqslant m$ $(j = 1,2, \ldots, n)$ kikötéseknek és (1.1)-nek eleget tevő argumentumokhoz tartozó $\alpha^{(m,n)}$ értékeknek

$$y_1^{a_1} y_2^{a_2} \ldots y_m^{a_m} z_1^{b_1} z_2^{b_2} \ldots z_n^{b_n} \text{-nel}$$

való szorzatát. Ha az (1.1) kikötést feloldjuk, és annak nem teljesülése esetén az $\alpha^{(m,n)}$ függvényhez 0 értéket rendelünk, akkor irhatjuk, hogy

$$G^{(m,n)}(y_1, y_2, \ldots, y_m;\ z_1, z_2, \ldots, z_n) =$$

$$(7.1) \qquad = \sum_{a_1=0}^{n} \sum_{a_2=0}^{n} \ldots \sum_{a_m=0}^{n} \sum_{b_1=0}^{m} \sum_{b_2=0}^{m} \ldots \sum_{b_n=0}^{m} \alpha^{(m,n)}(a_1, a_2, \ldots, a_m;$$

$$b_1, b_2, \ldots, b_n) \cdot y_1^{a_1} y_2^{a_2} \ldots y_m^{a_m} z_1^{b_1} z_2^{b_2} \ldots z_n^{b_n}.$$

7.1. **Tétel.** *Tetszőleges pozitiv egész* m *és* n *esetén*

$$(7.2) \qquad G^{(m,n)}(y_1, y_2, \ldots, y_m; z_1, z_2, \ldots, z_n) = \prod_{i=1}^{m} \prod_{j=1}^{n} (1 + y_i z_j).$$

Bizonyitás. Jelöljük S-sel az $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant n$, i,j egész feltételeknek eleget tevő (i,j) számpárok halmazát. A (7.2) egyenlőség jobboldalán álló kifejezés polinom alakja

$$(7.3) \qquad \sum_{S' : S' \subset S} \left[\prod_{(i,j) \in S'} y_i z_j \right],$$

ahol $\in$ elemként való tartalmazást, $\subset$ pedig valódi vagy nem valódi részhalmazként való tartalmazást jelent.

A (7.3) alatti összeg minden tagja $y_1^{a_1} y_2^{a_2} \ldots y_m^{a_m} z_1^{b_1} z_2^{b_2} \ldots z_n^{b_n}$ alakú, ahol a_i, b_j egész, $0 \leqslant a_i \leqslant n$, $0 \leqslant b_j \leqslant m$, $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, ezért a 7.1. tétel bizonyitásához már csak azt kell belátnunk, hogy a $\prod_{i=1}^{m} \prod_{j=1}^{n} (1 + y_i z_j)$ polinomban

$y_1^{a_1} y_2^{a_2} \ldots y_m^{a_m} z_1^{b_1} z_2^{b_2} \ldots z_n^{b_n}$ együtthatója

$\alpha^{(m,n)}(a_1, a_2, \ldots, a_m; b_1, b_2, \ldots, b_n)$ minden szóbajövő a_i, b_j értékrendszerre.

Tekintsük az 1. szakaszban szereplő $m \cdot n$ cellából álló táblázat A-elemmel és B-elemmel történő minden lehetséges teljes kitöltését. Az $m \cdot n$ cella mindegyikébe vagy A-elem vagy B-elem kerül, tehát a lehetséges kitöltések száma $2^{m \cdot n}$, s ez megegyezik a (7.3) alatt álló összeg tagjainak a számával. Minden ilyen kitöltéshez kölcsönösen egyértelmüen rendelhető a táblázat azon celláiból álló halmaz, amelyek A-elemet tartalmaznak. A táblázat cellái az i,j egész, $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant n$ feltételeknek eleget tevő (i,j) indexpárokkal jellemezhetők, igy a táblázat celláiból alkotott minden halmazhoz kölcsönösen egyértelmü módon rendelhető a (7.3)-ban szereplő S' részhalmazok valamelyike, továbbá a (7.3) alatt álló összegnek az ehhez tartozó

$\prod_{(i,j) \in S'} y_i z_j$ tagja.

Ilymódon kölcsönösen egyértelmű megfeleltetést létesitettünk a táblázat lehetséges kitöltései és a (7.3) alatti összeg tagjai között. A

$$\prod_{(i,j) \in S'} y_i z_j$$

szorzatban y_i annyiszor szerepel tényezőként, ahány A-elem van a megfelelő kitöltésnél a táblázat i-edik sorában, hasonlóan z_j annyiszor szerepel tényezőként, ahány A-elem van a megfelelő kitöltésnél a táblázat j-edik oszlopában. Következésképpen az

$$y_1^{a_1} y_2^{a_2} \ldots y_m^{a_m} z_1^{b_1} z_2^{b_2} \ldots z_n^{b_n}$$

szorzat annyiszor fordul elő a (7.3) alatti összeg tagjai között, ahány olyan teljes kitöltése van a táblázatnak, melyre a táblázat i-edik sorának cellái összesen a_i számú A-elemet tartalmaznak minden $i = 1,2, \ldots, m$ esetén és a táblázat j-edik oszlopának cellái összesen b_j számú A-elemet tartalmaznak minden $j = 1,2, \ldots, n$ esetén.

8. A táblázatkitöltési probléma módositása a táblázat egy cellája elemének rögzitésével

Az 1. szakaszban szereplő táblázatkitöltési problémánál általánosabb problémához jutunk, ha további megkötésként előirjuk, hogy a táblázat bizonyos celláiba az A-elem és B-elem közül melyik kerüljön. Ha csak egy cellára irjuk elő, hogy az A-elem és B-elem közül melyiket kell tartalmaznia, akkor a megoldások számára az alábbi tétel nyújt információt.

8.1. **Tétel.** *Adott egy m sorból, n oszlopból és $m \cdot n$ cellából álló táblázat, adottak ezenkivül az $a_i (i = 1,2, \ldots, m)$ és $b_j (j = 1,2, \ldots, n)$ egész számok, melyekre* (1.1) *teljesül. A táblázat olyan A-elemekkel és B-elemekkel történő teljes kitöltéseire szoritkozva, melyeknél az i-edik sor celláiba összesen a_i számú A-elemet és $n - a_i$ számú B-elemet helyezünk minden $i = 1,2, \ldots, m$ esetén, a j-edik oszlop celláiba pedig összesen b_j számú A-elemet és $m-b_j$ számú B-elemet helyezünk minden $j = 1,2, \ldots, n$ esetén, továbbá a (k,l) cellába csak A-elemet helyezhetünk, akkor a különböző lehetséges kitöltések száma*

$$(8.1) \qquad \sum_{\substack{u_1+u_2+\ldots+u_m=b_l \\ u_i=0 \text{ vagy } 1 \ (i=1,2,\ldots,m) \\ u_k=1}} \alpha^{(m,n-1)}(a_1 - u_1, a_2 - u_2, \ldots, a_m - u_m; b_1, b_2, \ldots, b_{l-1}, b_{l+1}, \ldots, b_n)$$

Ha pedig a feltételeket úgy módositjuk, hogy a (k,l) cellába csak B-elemet helyezhetünk, akkor a különböző lehetséges kitöltések száma

$$(8.2) \qquad \sum_{\substack{u_1+u_2+\ldots+u_m=b_l \\ u_i=0 \text{ vagy } 1 \ (i=1,2,\ldots,m) \\ u_k=0}} \alpha^{(m,n-1)}(a_1 - u_1, a_2 - u_2, \ldots, a_m - u_m; b_1, b_2, \ldots, b_{l-1}, b_{l+1}, \ldots, b_n)$$

Bizonyitás. Ha a táblázat l-edik oszlopában valamennyi cellát kitöltjük úgy, hogy az oszlop k-adik cellájába, továbbá az $i_1, i_2, \ldots, i_{b_l - 1}$ – edik cellájába A-elemet, az oszlop többi cellájába B-elemet helyezünk, akkor az eredeti táblázat egy m sorból és $n-1$ oszlopból álló táblázattá redukálódik, a sorokhoz tartozó a_i peremértékek nem változnak ott, ahol az (i,l) cellába B-elemet helyeztünk, 1-gyel csökkennek ott, ahol az (i,l) cellába A-elemet helyeztünk, a megmaradó oszlopokhoz tartozó b_j peremértékek nem változnak.
Legyen $u_i = 1$, vagy $u_i = 0$ attól függően, hogy az (i,l) cellába A-elemet vagy B-elemet helyeztünk $(i = 1,2, \ldots, m)$. A csökkentett táblázat lehetséges kitöltéseinek a száma az α függvény definiciója szerint éppen a (8.1) alatti összegnek a szóbanforgó u_i-khoz tartozó tagja. A 8.1. tétel első részének feltételeit kielégitő kitöltések száma pedig nem más mint a csökkentett, m sort és $n-1$ oszlopot tartalmazó táblához tartozó megoldásszámoknak az l-edik oszlop megfelelő, azaz a

$$\sum_{i=1}^{m} u_i = b_l, \quad u_i = 0 \quad \text{vagy} \quad 1 (i = 1,2, \ldots, m), \quad u_k = 1$$

feltételeknek eleget tevő kitöltésekre való összegezése.

Ezzel a 8.1. tétel első részét igazoltuk. A második rész bizonyitása ugyanigy végezhető el.

9. Rekurziós formulák az α, μ és ω függvényekre.

A 8.1. tételből azonnal adódik egy rekurzió az α függvényre, amely átfogalmazható a μ és ω függvényekre is.

9.1. **Tétel.**

a.) *Az α függvény értékeire az alábbi rekurziós formula érvényes:*

$$\alpha^{(m,n)}(a_1, a_2, \ldots, a_m;\ b_1, b_2, \ldots, b_n) =$$

(9.1)
$$= \sum_{\substack{u_1 + u_2 + \ldots + u_m = b_l \\ u_i = 0 \text{ vagy } 1 \ (i=1,2,\ldots,m)}} \alpha^{(m,n-1)}(a_1 - u_1, a_2 - u_2, \ldots, a_m - u_m;\ b_1, b_2, \ldots, b_{l-1}, b_{l+1}, \ldots, b_n)$$

b.) *A μ függvény értékeire az alábbi rekurziós formula érvényes:*

$$\mu^{(m,n)}(s_0, s_1, \ldots, s_n;\ b_1, b_2, \ldots, b_n) =$$

(9.2)
$$= \sum_{\substack{w_1 + w_2 + \ldots + w_{n-1} = b_l - s_n \\ 0 \leqslant w_i \leqslant s_i \\ w_i \text{ egész}}} \binom{s_1}{w_1} \cdot \binom{s_2}{w_2} \cdots \binom{s_{n-1}}{w_{n-1}} \cdot \quad (i=1,2,\ldots,n-1)$$

$$\cdot \mu^{(m,n-1)}(s_0 + w_1, s_1 - w_1 + w_2, s_2 - w_2 + w_3,$$
$$\ldots, s_{n-2} - w_{n-2} + w_{n-1}, s_{n-1} - w_{n-1} + s_n;\ b_1, b_2, \ldots, b_{l-1}, b_{l+1}, \ldots, b_n)$$

c.) *Az* ω *függvény értékeire az alábbi rekurziós formula érvényes:*

$$\omega^{(m,n)}(s_0,s_1,\ldots,s_n;\ t_0,t_1,\ldots,t_m) =$$

$$(9.3) \qquad = \sum_{\substack{w_1+w_2+\ldots+w_{n-1}=t-s_n \\ 0\leqslant w_i\leqslant s_i \\ w_i \text{ egész}}} \binom{s_1}{w_1}\cdot\binom{s_2}{w_2}\cdots\binom{s_{n-1}}{w_{n-1}}\cdot \quad (i=1,2,\ldots,n-1)$$

$$\cdot\ \omega^{(m,n-1)}(s_0+w_1,s_1-w_1+w_2,s_2-w_2+w_3,$$

$$\ldots,s_{n-2}-w_{n-2}+w_{n-1},s_{n-1}-w_{n-1}+s_n;t_0,t_1,\ldots,t_{l-1},t_l-1,t_{l+1},\ldots,$$

feltéve, hogy $l\geqslant 1$ *és* $t_l\geqslant 1$.

Bizonyitás.

a.) A 8.1. tételből következik, hogy a (8.1) alatti mennyiségnek és a (8.2) alatti mennyiségnek összegül ki kell adniuk az

$$\alpha^{(m,n)}(a_1,a_2,\ldots,a_m;\ b_1,b_2,\ldots,b_n)$$

értéket.

b.) Helyettesitsük a μ függvényt az α függvénnyel, ezután jelöljük w_j-vel az s_j számú j értékü a_i-hez tartozó u_i összegét $j=1,2,\ldots,n-1$, esetén. $a_i=0$ esetén a_i-hez csak $u_i=0$ tartozhat, $a_i=n$ esetén a_i-hez csak $u_i=1$ tartozhat. A w_j szám $\binom{s_j}{w_j}$ féleképpen állitható elő s_j számú 0 vagy 1 értékü u_i összegeként. Igy a (9.1) formulát a μ függvényre alkalmazva (9.2)-höz jutunk.

c.) (9.2)-ben a μ függvényt az ω függvénnyel pótolva adódik (9.3).

10. A polinomiális együtthatók általánositása

A (9.1) alatti rekurziós formula erősen emlékeztet a Pascal háromszögre (illetve annak többdimenziós megfelelőjére) vonatkozó rekurziv képzési szabályra. Ez nem véletlen, mivel az 5.1 tétel szerint és a következő 10.1. tétel szerint is az α,μ és ω függvények értékei a polinomiális együtthatók egyfajta általánositásának tekinthetők.

10.1. **Tétel.**

a.) $$\alpha^{(m,n)}(1,1,\ldots,1;b_1,b_2,\ldots,b_n)=\frac{(b_1+b_2+\ldots+b_n)!}{b_1!\cdot b_2!\ldots b_n!}$$

ha $b_j\geqslant 0$ $(j=1,2,\ldots,n)$ *és* $m=b_1+b_2+\ldots+b_n$.

b.)
$$\mu^{(m,n)}(s_0,s_1,\ldots,s_n;\ b_1,b_2,\ldots,b_n)=\frac{s_1!}{b_1!\cdot b_2!\ldots b_n!}$$

ha $\quad s_i=0 \quad (i=2,3,\ldots,n)$ *és* $b_j\geqslant 0 \quad (j=1,2,\ldots,n).$

c.)
$$\omega^{(m,n)}(s_0,s_1,\ldots,s_n;\ t_0,t_1,\ldots,t_m)=\frac{s_1!}{(2!)^{t_2}\cdot(3!)^{t_3}\ldots(m!)^{t_m}}$$

Bizonyitás. Elég a tétel b.) állitását igazolni, a másik két állitás csak ennek átfogalmazása. A b.) rész bizonyitásához az 5.1. tételt alkalmazzuk arra az esetre, ha $s_2=s_3=\ldots=s_n=0$: az 5.1. tétel a.) állitása szerint a

$\sigma_0^{s_0}\cdot\sigma_1^{s_1}$ polinomban $y_1^{b_1}y_2^{b_2}\ldots y_n^{b_n}$ együtthatója $\mu^{(m,n)}(s_0,s_1,\ldots,s_n;\ b_1,b_2,\ldots,b_n)$ $(s_2=s_3=\ldots=s_n=0)$. Másrészt a polinomiális tétel szerint a

$$\sigma_0^{s_0}\cdot\sigma_1^{s_1}=\sigma_1^{s_1}=(y_1+y_2+\ldots+y_n)^{s_1}$$

kifejezésben

$y_1^{b_1}y_2^{b_2}\ldots y_n^{b_n}$ együtthatója $\dfrac{s_1!}{b_1!b_2!\ldots b_n!}$

Végül bizonyitás nélkül megadjuk egy-egy speciális esetre a $\mu^{(m,n)}$ illetve az $\omega^{(m,n)}$ függvény explicit alakját (10.2 illetve 10.3. tétel).

10.2. **Tétel.** $b_1,b_2,b_3,s_0,s_1,s_2,s_3\geqslant 0$, $m=b_1+b_2+b_3=s_1+2s_2+3s_3$, $n=3$ *esetén* $\mu^{(m,3)}(s_0,s_1,s_2,s_3;\ b_1,b_2,b_3)=$

$$=\sum_{\substack{l_{12}+l_{13}=s_2\\ l_{12}+l_{23}=s_2\\ l_{12},l_{13},l_{23}\geqslant 0,\ \text{egész}}}\frac{s_2!}{l_{12}!l_{13}!l_{23}!}\cdot\frac{s_1!}{(b_1-s_3-l_{12}-l_{13})!(b_2-s_3-l_{12}-l_{23})!(b_3-s_3-l_{13}-l_{23})!}$$

10.3. **Tétel.** $s_0\geqslant 0$, $s_1\geqslant 0$, $s_2=k+1$,
$s_3=s_4=\ldots=s_n=0$, $t_0\geqslant 0$, $t_1=s_1+2k$, $t_2=1$, $t_3=t_4=\ldots=t_m=0$
esetén tetszőleges nemnegativ egész $k,m,n,$ *értékekre*

$$\omega^{(m,n)}(s_0,s_1,\ldots,s_n;\ t_0,t_1,\ldots,t_m)=$$
$$=\frac{t_1!}{2^{k+2}}\cdot[s_1^2+(4k+3)\cdot s_1+4k(k+1)].$$

Irodalom

[1] D. Gale, A theorem on flows in networks, Pacific Journal of Mathematics 7 (1957), 1073 – 1082.

[2] Klafszky, E., Hálózati folyamok, Az operációkutatás matematikai módszerei c. tanfolyam anyaga (Bolyai János Matematikai Társulat, Budapest, 1969.)

[3] H. J. Ryser., Combinatorial properties of matrices of zeros and ones, Canadian Journal of Mathematics 9 (1957), 371 – 377.

[4] G., Kéri , A tableau filling problem and its possible application to matrix inversion, in III. conference on mathematical programming, Mátrafüred, January 31. – February 6, 1975.

Summary

A combinatorial problem of tableau filling and its equivalent variants in connection with different subjects

Gerzson Kéri

The author of this paper shows that the problem of determining

a.) the number of all such different feasible solutions of the transportation problem that have 0 or 1 in all components,

b.) the number of bipartite graphs with given degrees of the vertices,

c.) the coefficients of certain symmetrical polynomials –

are all equivalent with a problem of tableau filling.

Generator functions, recursive formulas, and in several special cases explicit formulas, too are given for the solution of the equivalent problems under discussion.

Р Е З Ю М Е

Об одной комбинаторной проблеме заполнения таблиц и её эквивалентных версиях в связи с разными темами

Гержон Кери

Автор показывает, что следующие проблемы:

а/ определить число всех разных 0-1 решений системы условий транспортной задачи;

б/ определить число всех четных графов, имеющих данные степени вершин;

в/ определить коэффициенты некоторых симметричных полиномов -

все это является эквивалентным с одной проблемой заполнения таблиц.

В статье даются генераторные функции, рекурсивные формулы - и в нескольких специальных случаях тоже эксплицитные формулы - для решения вышеупомянутых эквивалентных проблем.

MTA Számítástechnikai és Automatizálási Kutató Intézete, **Közlemények** 19/1978.

A TÖBBDIMENZIÓS NORMÁLIS ELOSZLÁSFÜGGVÉNY MONTE CARLO INTEGRÁLÁSSAL TÖRTÉNŐ KISZÁMITÁSÁNAK SZÁMITÓGÉPES TAPASZTALATAI

Deák István

1. Bevezetés

A többdimenziós normális eloszlásfüggvény konkrét értékeinek kiszámitása több gyakorlati példában szükséges lehet. Például Prékopa [3] STABIL modelljének megoldása folyamán, amely a következő alakú:

$$(1.1)\qquad \begin{aligned} & P\{g_i(\underline{x}) \geqslant \beta_i,\quad i = 1, \ldots, n\} \geqslant P, \\ & \underline{a}_i' \underline{x} \geqslant b_i,\quad i = 1, \ldots, m, \\ & \min f(\underline{x}); \end{aligned}$$

ahol a $\beta_1, \beta_2, \ldots, \beta_n$ valószinüségi változók együttes eloszlása n dimenziós normális eloszlás. A cikkben leirunk egy szubrutinrendszert, amely egy Monte Carlo integrálási technikával számitja ki a $\Phi(\underline{h})$ konkrét értékeit, ahol $\Phi(\underline{h})$ a zérus várható értékü, 1 szórású és R korrelációsmátrixú normális eloszlásfüggvény, vagyis

$$(1.2)\qquad \Phi(\underline{h}) = \frac{1}{(2\pi)^{n/2}|R|^{1/2}} \int_{-\infty}^{h_n} \cdots \int_{-\infty}^{h_1} \exp\left\{-\frac{1}{2}\underline{x}R^{-1}\underline{x}\right\} d\underline{x}.$$

A felhasznált technikát úgy választottuk, hogy egy konkrét $\Phi(\underline{h})$ értéket a lehető legrövidebb időn belül számitson ki a számitógép és a szubrutinok tetszőleges R korrelációs mátrix és $\underline{h}$ vektor esetén müködjenek az $n = 12$ dimenziós esetben is. Az itt leirt technikát alkalmaztuk a magyar villamosenergiaiparra illesztett STABIL modell számitógépes kiszámitása során [3].

A 2. szakaszban az alkalmazott Monte Carlo technika elvi leirását adjuk meg, a 3. szakaszban a számitógépes program szerkezetét tárgyaljuk – különös tekintettel a minél gyorsabb lefutásra. A 4. szakaszban a számitógépes tapasztalatokat, valamint a szubrutinok sebességére vonatkozó időeredményeket irjuk le.

2. A többdimenziós normális eloszlásfüggvény konkrét értékeinek kiszámitása Monte Carlo integrálással

Legyen feladatunk az

$$(2.1)\qquad I_1 = \Phi(\underline{h}) = \int_{-\infty}^{h_n} \cdots \int_{-\infty}^{h_1} \varphi(\underline{x}) d\underline{x}$$

integrál meghatározása, ahol $\varphi(\underline{x})$ a 0 várható értékü, 1 szórású, R korrelációsmátrixú n dimenziós normális sürüségfüggvény, vagyis

$$\varphi(\underline{x}) = \frac{1}{(2\pi)^{n/2}\,|R|^{1/2}} \exp\left\{-\frac{1}{2}\,\underline{x}R^{-1}\underline{x}\right\},$$

ahol R pozitiv definit szimmetrikus mátrix, melynek a főátlójában 1-esek állnak. Legyen D a következő halmaz:

(2.2) $$D = \{\underline{x} | \underline{x} \leqslant \underline{h}\}.$$

Az I_1 érték meghatározása ekvivalens a $\varphi(\underline{x})$ függvény D tartomány feletti integráljának kiszámításával. A gyakorlati kiszámitás folyamán a D halmaz helyett a

(2.3) $$D^* = \{\underline{x} | -\underline{b} \leqslant \underline{x} \leqslant \underline{h}\}$$

tartomány felett integrálunk, ahol a $\underline{b}$ vektor mindegyik komponensének az értéke néggyel egyenlő. Az integrálási tartomány ilyen csonkolása által elkövetett hiba elhanyagolhatóan kicsi az alkalmazott Monte Carlo módszer hibáihoz képest, viszont egyszerübbé és gyorsabbá teszi a számitásokat.

Az I_1 érték becslésére használjuk a

(2.4) $$\Theta_1 = \frac{1}{N} \prod_{j=1}^{n} (h_j + b) \cdot \sum_{i=1}^{N} \varphi(\underline{v}^{(i)})$$

valószinüségi változót, ahol $\underline{v}^{(i)}$; $i = 1, \ldots, N$ olyan valószinüségi vektorváltozók, amelyek egyenletes eloszlásuak a D^* tartományban. Azért kellett a D tartomány helyett a D^* korlátos tartományt bevezetni, hogy egyenletes eloszlású pszeudovéletlen vektorokat tudjunk az integrálási tartományban elöállitani. A (2.4) képlet lényegében a D^* tartomány területével szorozza az e terület feletti függvényértékek számtani átlagát, ezt a Θ_1 becslést integrálközépnek nevezzük. A számitás gyakorlatilag úgy történik, hogy a $\underline{v}^{(i)}$, $i = 1, \ldots, N$ valószinüségi vektorváltozónak N darab $\underline{r}^{(i)}$, $i = 1, \ldots, N$ realizációját vesszük és ezek segitségével a Θ_1 valószinüségi váltózónak a várható értékére egy becslést kapunk.

Vizsgáljuk meg a (2.4) képleten alapuló eljárás hibáját. Vezessük be a

(2.5) $$\xi_i = \prod_{j=1}^{n} (h_j + b)\varphi(\underline{v}^{(i)})$$

jelölést; Θ_1 a ξ_i valószinüségi változók számtani közepe, vagyis

$$\Theta_1 = \frac{1}{N} \sum_{i=1}^{N} \xi_i.$$

Erre pedig a Csebisev egyenlőtlenség segitségével kapjuk, hogy

$$P\{|\Theta_1 - M(\Theta_1)| \leqslant x_p D(\Theta_1)\} \geqslant p,$$

valamilyen p valószinüségi szinttel. Az x_p változó értéke a p döntési szinttől függ,

például $p = 0{,}997$ szinten $x_p = 3{,}0$ lesz. Ha figyelembe vesszük még, hogy

$$M(\Theta_1) = I_1, \qquad D^2(\Theta_1) = \frac{D^2(\xi_i)}{N},$$

akkor valamilyen, p-hez közeli valószínüséggel igaz lesz, hogy

$$|\Theta_1 - I_1| \leqslant x_p \frac{D(\xi_i)}{\sqrt{N}} \cdot \tag{2.6}$$

Tehát az integrálás pontossága forditottan arányos a mintapontok számának négyzetgyökével és egyenesen arányos a ξ_i valószinüségi változók szórásával, vagyis a hiba nagysága nagymértékben a $D(\varphi(\underline{v}^{(i)}))$ szórástól függ. Kézenfekvőnek látszik a ξ_i valószinüségi változók szórásának csökkentésével javitani a kiszámitási eljárást. Az irodalomban javasolt többféle szóráscsökkentő eljárást (importance sampling, control varietes) kipróbálva nem lehetett a gépidőt – azonos pontosságú eredmény elérését megkivánva – csökkenteni; bár kevesebb mintapontot kellett felvenni, de a kiszámitás bonyolultsága miatt megnőtt a futtatási idő.

A felhasznált ξ_i valószinüségi változók szórását a következő ötlet segitségével lehet azonban csökkenteni. Legyen K a következő n dimenziós kocka:

$$K = \{\underline{x} \mid -\underline{b} \leqslant \underline{x} \leqslant \underline{b}\}.$$

Tekintettel arra, hogy $\varphi(x)$ értéke a K kockán kivül már elég kicsi, ha $\underline{b} \geqslant \underline{4}$, egy elhanyagolható δ hibával fennáll az

$$1 = \int_{R^n} \ldots \int \varphi(\underline{x})d\underline{x} = \int_K \ldots \int \varphi(\underline{x})d\underline{x} + \delta$$

egyenlőség. Ezért a $\varphi(\underline{x})$ függvénynek a $K - D^*$ tartomány feletti integrálját I_2-vel jelölve

$$I_1 = \int_{D^*} \ldots \int \varphi(\underline{x})d\underline{x} = 1 - \int_{R^n} \ldots \int \varphi(\underline{x})d\underline{x} + \int_{D^*} \ldots \int \varphi(\underline{x})d\underline{x} = \tag{2.7}$$

$$= 1 - \int_{K-D^*} \ldots \int \varphi(\underline{x})d\underline{x} - \delta = 1 - I_2 - \delta \approx 1 - I_2,$$

ahol a $\approx$ jel azt mutatja, hogy a gyakorlatban elhanyagolható hibával egyenlőség áll fenn. Az I_2 integrált a következő valószinüségi változóval becsüljük:

$$\Theta_2 = \frac{1}{N}[(2b)^n - \prod_{j=1}^{n}(h_j + b)] \cdot \sum_{i=1}^{N} \varphi(\underline{v}^{(i)}); \tag{2.8}$$

ahol a $\underline{v}^{(i)}$, $i = 1, \ldots, N$ valószinüségi változók egyenletes eloszlásuak a $K - D^*$ tartományban. Tehát (2.8) szerint a Θ_2 becslés a $K - D^*$ tartományban a függvényértékek átlagának szorzataként áll elő.

A gyakorlati alkalmazásokban az I_1 integrál értéke $0.80 - 0.95$ körül szokott lenni, igy az I_2 valószinüség $0.05 - 0.20$ között változik. Ha I_1 helyett az I_2 értéket becsüljük meg, vagyis a Θ_1 összeg helyett a Θ_2 összeget számitjuk ki, akkor a $\xi_i = \varphi(\underline{v}^{(i)})$

valószinüségi változók szórását azáltal csökkentjük, hogy a $\varphi(\nu^{(i)})$ értékek kisebbek a $K - D^*$ tartományban, mint a D^* tartományban.

Tehát érdemesebb ugy meghatározni az I_1 integrál értékét, hogy kiszámitjuk az I_2 egy becslését a Θ_2 összeg segitségével egy adott $\underline{h}$ vektor és rögzitett N esetén, majd az $I_1 \approx 1 - I_2$ közelitő egyenlőség felhasználásával számitjuk ki az I_1 értéket. Ha az I_1 érték közel van 1-hez, akkor ez a kiszámitási eljárás a (2.6) hibabecslés szerint kisebb hibáju lesz, mintha az adott $\underline{h}$ és N esetén az I_1 értéket közvetlenül a Θ_1 összeg meghatározásával számitanánk ki. Mivel egy adott eloszlás és $\underline{h}$ vektor esetén előre nem tudhatjuk, hogy I_1 értéke 1-hez közeli lesz vagy sem, a következő kritériumot használtuk a szubrutinokban annak eldöntésére, hogy a Θ_1 összeg vagy a Θ_2 összeg meghatározásával számitsuk ki I_1 értékét. Jelölje T_{D^*}, T_K, T_{K-D^*} rendre a D^*, K, $K-D^*$ tartományok területét. Ha a

$$\frac{T_{D^*}}{T_K} < (0{,}6)^n \tag{2.9}$$

egyenlőtlenség teljesül, ahol n az eloszlás dimenziószáma, akkor az adott $\underline{h}$ vektor esetében a Θ_1 összeget számitjuk, egyébként a Θ_2 átlag segitségével becsüljük az I_2 értéket és ebből számitjuk ki az I_1 értékét. Szemléletesen nézve a (2.9) feltétel azt vizsgálja, hogy a $\varphi(\underline{x})$ sürüségfüggvénynek az origóban felvett csucsa és a körülötte levő nagyobb értékek hová esnek és azt az eljárást választjuk, amelynek alkalmazásával ezeket a nagyobb $\varphi(\underline{x})$ értékeket elkerülhetjük. Ezzel a választással a Θ_i összegben felhasznált valószinüségi változók szórása kisebb lesz.

A számitógépes program további gyorsitását lehet elérni a következő gondolat segitségével. A $K - D^*$ tartományban egyes helyeken olyan kicsi lesz a $\varphi(\underline{\nu}^{(i)})$ értéke, hogy még a T_{K-D^*} területtel megszorozva is elhanyagolhatóan kis értéket kapunk. A Θ_2 átlag kiszámitását igy a következő módon végezzük. Ha

$$T_{K-D^*} \cdot \varphi(\underline{\nu}^{(i)}) < 0{,}001 \tag{2.10}$$

akkor ez a tag a Θ_2 összegben csak $\frac{1}{N} \cdot 0{,}001$ értéket ad, igy ez a tag a Θ_2 átlagból elhagyható. A Θ_2 átlag meghatározása a következő számitógépes lépésekből tevődik össze:

1.) Generálunk egy $\underline{\nu}^{(i)}$ egyenletes eloszlásu pontot a K kockában, megvizsgáljuk, hogy a $K - D^*$ tartományba esik-e, ha nem, akkor uj véletlen pontot generálunk, egyébként a 2.) lépés következik.

2.) Kiszámitjuk a $\varphi(\underline{x})$ sürüségfüggvényben szereplő

$$S_i = -\frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} R_{jk}\, \nu_k^{(i)} \nu_j^{(i)}$$

kvadratikus alakot, ahol R_{jk} az R korrelációs mátrix inverzének (j,k)-adik eleme.

3.) Meghatározzuk az $\exp(S_i)$ értéket és hozzáadjuk a Θ_2 átlag már eddig kiszámitott részéhez.

4.) Ha már N darab $\underline{y}^{(i)}$ vektort generáltunk a $K - D^*$ tartományban, akkor a Θ_2 összeget beszorozzuk a még hiányzó álandókkal.

Lényeges az, hogy a (2.10) kritérium teljesülését már a 2.) lépés után ellenőrizhetjük, ugyanis a

$$T_{K-D} \frac{1}{(2\pi)^{\frac{n}{2}} |R|^{\frac{1}{2}}} \cdot \exp\{S_i\} \leqslant 0{,}001$$

feltételből átalakitásokkal kapjuk, hogy ha az

$$(2.11) \qquad S_i \leqslant \ln\left\{\frac{1}{T_{K-D^*}} 0{,}001 \cdot (2\pi)^{\frac{n}{2}} \cdot |R|^{\frac{1}{2}}\right\}$$

egyenlőtlenség fennáll (ami ekvivalens a (2.10) feltétellel), akkor a 3. lépést ennek a $\underline{y}^{(i)}$ vektornak az esetében már nem kell elvégezni, a Θ_2 összegben a megfelelő tagot zérusnak tekintjük.

A fenti gondolatmenet segitségével további módositást végzünk még el. A $\underline{b} = \underline{4}$ csonkoló vektor helyett egy olyan $\underline{c}$ vektort vehetünk, amelyre teljesül az, hogy ha $|\underline{x}| \geqslant \underline{c}$ akkor $T_{K-D^*} \cdot \varphi(\underline{x}) \leqslant 0{,}001$ lesz. Vagyis ujabb csonkolást hajtunk végre; a D^* vagy a $K - D^*$ tartomány helyett az integrálást a D^{**} vagy a $K^* - D^{**}$ tartomány felett végezzük, ahol

$$(2.12) \qquad \begin{aligned} D^{**} &= \{|\underline{x}| - \underline{c} \leqslant \underline{x} \leqslant \underline{h}\}, \\ K^* &= \{|\underline{x}| - \underline{c} \leqslant \underline{x} \leqslant \underline{c}\}. \end{aligned}$$

Ennek a lépésnek abban áll a jelentősége, hogy $\underline{c} \leqslant \underline{b}$ és ezért $T_{D^{**}} \leqslant T_{D^*}$ és $T_{K^*} \leqslant T_K$. Igy kisebb területü tartományban véve fel az N számú egyenletes eloszlású $\underline{y}^{(i)}$ pontokat, jobb, pontosabb eredményt kapunk. A gyakorlatban a $\underline{c}$ vektor komponenseire 2,5 – 4,0 közötti értékeket kapunk az R korrelációs mátrixtól függően.

3. A számitógépes program szerkezete

A szubrutinrendszer az MTA CDC 3300 számitógépére készült FORTRAN nyelven és 18 szubrutinból áll. Ezek hosszabb fejlesztési munka eredményeként készültek el, az itt leirt rendszer a negyedik változat. Az előző változatokról lásd [1].

A programok megirásánál a minél kisebb helyfoglalást és a lehető legrövidebb futási idő elérését tüztük ki célul. Az első szempont azért fontos, mert ezeket a szubrutinokat más programokon belül használják fel, a második pedig azért, hogy egy elfogadható gépi időn belül lefussanak a programok. A két szempont némileg ellentétes; az előző változat lényegesen több utasitásból állt, de valamivel gyorsabb volt. Két szubrutin közül azt lehet gyorsabbnak tekinteni, amely azonos p megbizhatósági szinten azonos várható hibával rövidebb idő alatt számitja ki egy adott eloszlás esetén ugyanazt a $\Phi(\underline{h})$ függvényértéket. (Ugyanis különböző $\underline{h}$

vektorok esetén a $\Phi(\underline{h})$ érték kiszámitása különböző gyorsaságú lesz).

A szubrutinok egy része előkészitő müveleteket végez. Meghatározzuk az adott korrelációs mátrix inverzét, kiszámitják az előző szakaszban emlitett D^{**} tartomány előállitásához szükséges $\underline{c}$ csonkoló vektor komponenseit, a (2.11) egyenlőtlenség jobb oldalán szereplő kifejezés értékét és egyéb állandókat.

Egy adott $\underline{h}$ vektor esetén a (2.6) alapján meg tudjuk becsülni, hogy adott $|\Theta_i - I_i|$, $i = 1,2$ hiba esetén és adott p biztonsági szinthez mekkora N mintaszámot kell választani a Monte-Carlo integráláshoz.
A $D(\xi_i)$ szórást a

$$(3.1) \qquad D^*(\xi_i) \approx \frac{1}{N} \sum_{i=1}^{N} \xi_i^2 - \left(\frac{1}{N} \sum_{i=1}^{N} \xi_i \right)^2$$

empirikus szórásnégyzettel becsüljük. Az egyes szubrutinok a következő módon müködnek:

ELJAV2 megadott N esetén a D^{**} tartomány felett integrál.

ELJAV3 megadott N esetén a $K^* - D^{**}$ tartomány felett integrál és a (3.1) összefüggés segitségével kiszámitja, hogy különböző hibák és megbizhatósági szintek esetén mekkora N mintaszám szükséges.

ELJAV4 megadott N esetén a $K^* - D^{**}$ tartomány felett integrál.

ELJAV5 megadott megbizhatósági szintü megadott hibához a (3.1) összefüggés felhasználásával számitja ki a valószinüséget a D^{**} tartomány feletti integrálással.

ELJAV6 ugyanugy müködik mint ELJAV5, csak a D^{**} tartomány helyett a $K^* - D^{**}$ tartomány felett integrál.

A minél rövidebb futási idő elérése érdekében szükségessé vált a véletlenszám generátor ügyes megválasztása és a sürüségfüggvény kitevőjének egyszerüsitett kiszámitása. Az

$$S_i = -\frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} R_{jk} \nu_j^{(i)} \nu_k^{(i)}$$

jelölést használva a

$$(3.2) \qquad \Theta_2 = \frac{1}{N} \cdot T_{K^*-D^{**}} \cdot \sum_{i=1}^{N} \frac{1}{(2\pi)^{\frac{h}{2}} |R|^{\frac{1}{2}}} \exp(S_i)$$

kifejezés kiszámitásának során, az összegképzés minden lépésben egy darab, a $K^* - D^{**}$ tartományban egyenletes eloszlásu $\underline{\nu}^{(i)}$ valószinüségi vektorváltozóra van szükségünk. A következő igen egyszerü és emiatt gyors módszert használtuk a $\underline{\nu}^{(i)}$ vektorok előállitására. Jelöljük $\nu_j^{(i)}$-vel a $\underline{\nu}^{(i)}$ vektor j-edik komponensét, a $\underline{\nu}^{(i+1)}$ vektort a következő rekurziv összefüggésekkel állitottuk elő:

$$v_j^{(i+1)} = v_{j+1}^{(i)}, \quad j = 1, \ldots, n-1,$$

(3.3)

$$v_n^{(i+1)} = v_1^{(i)} + v_n^{(i)}; \quad (\mathrm{mod}\ b);$$

ahol b azonos a D^* tartomány meghatározásában szereplő konstanssal. Tekintettel arra, hogy ilyen kongruenciás véletlenszám generátorok felhasználásával óvatosan kell eljárnunk(lásd [2]), ezért minden 300-ik lépésben teljesen ujrageneráljuk a $\underline{v}^{(i)}$ vektort egy másik, karakterkeverésen alapuló véletlenszám generátorral. A (3.3) összefüggések segitségével előállitott $\underline{v}^{(i)}$ vektort megvizsgáljuk, hogy a $K^* - D^{**}$ tartományba esik-e, ha igen, akkor kiszámitjuk az S_i kifejezés érétékét ezzel a $\underline{v}^{(i)}$ vektorral, egyébként előállitjuk a $\underline{v}^{(i+1)}$ vektort.

A másik gyorsitási lehetőség az S_i érték kiszámitásánál adódott. Az $R^{-1} = \{R_{jk}\}_{j,k}$ mátrix szimmetrikus és igy ezt a $\sum$ jel előtti $-\frac{1}{2}$ tényezővel egybeolvasztva az S_i kifejezés felirható egy Q háromszögmátrix segitségével a következő alakban:

$$(3.4) \qquad S_i = -\frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n}{}' R_{jk} v_j^{(i)} v_k^{(i)} = \sum_{j=1}^{n} \sum_{k=1}^{j}{}' Q_{jk} v_j^{(i)} v_k^{(i)}$$

ahol $Q_{jj} = -\frac{1}{2} R_{jj}$ és $Q_{jk} = -R_{jk}$, $j = 1, \ldots, n$, $k = 1, \ldots, j$.

Ezzel a felirással az S_i kiszámitásánál szereplő tagok számát n^2-ről $\frac{n(n+1)}{2}$-re csökentettük. További gyorsitást ad az a módositás, ha az S_i kétszeres összegét nem egy kétszeres DØ ciklus segitségével számitjuk ki, – ahogyan a legkézenfekvőbb lenne – hanem részletesen kiirjuk az összeget. Ezzel a kissé hosszadalmas felirással elkerüljük a szimbolikus indexezés miatt szükséges időt; a futási idő a harmadára csökken ezáltal (bővebbet erről a következő szakaszban irunk).

4. Számitógépes tapasztalatok és futási idők

A szubrutinok FORTRAN nyelven készültek, ellenőrzésüket több különböző $\underline{h}$ vektor és korrelációs mátrix esetén végeztük el. A különböző korrelációs mátrixokat véletlenszám generátor segitségével állitottuk elő a következő módon. Generáltuk $\underline{u}^{(1)}, \underline{u}^{(2)}, \ldots, \underline{u}^{(n)}$ n darab n dimenziós független vektort, amelyek komponensei a (0,1) intervallumban egyenletes eloszlásu és független pszeudovéletlen számok voltak. Ezekre alkalmaztuk az

$$(4.1) \qquad \underline{u}^{(i)*} = \frac{\underline{u}^{(i)}}{\sqrt{|u^{(i)}|}}$$

transzformációt, ahol

$$|\underline{u}^{(i)}| = \sum_{j=1}^{n} u_j^{(i)2}$$

Legyen az R mátrix (i,j)-edik eleme az $\underline{u}^{(i)*}\,\underline{u}^{(j)*}$ skalárszorzat. Az ilyen módon előállitott $R = \{ u^{(i)*}\, u^{(j)*} \}_{i,j}$ mátrix szimmetrikus, pozitiv definit lesz, a főátlóban mindenütt 1 áll, igy tekinthető a standardizált többdimenziós normális eloszlás korrelációs mátrixának. A $\Phi(\underline{h})$ érték kiszámításánál szereplő $\underline{h}$ vektort szintén véletlenszám generátor segitségével állitottuk elő; a $\underline{h}$ vektor komponensei egymástól független, a (0,5) intervallumban egyenletes eloszlásu pszeudóvéletlen számok voltak.

Több ilyen módon előállitott R mátrix és $\underline{h}$ vektor esetén futtattuk le a szubrutinokat és ebben a szakaszban az igy szerzett tapasztalatokat közöljük. Először a $\varphi(\underline{v}^{(i)})$ értékek kiszámitásának részletes időeredményeivel foglalkozunk, majd az egyes szubrutinok futási idejével és pontosságával kapcsolatos eredményeket adjuk meg és végül a többdimenziós normális eloszlásfüggvény Monte Carlo integrálással történő kiszámitására vonatkozó következtetéseinket adjuk meg.

A $\xi_i = \varphi(\underline{v}^{(i)})$ értékek kiszámitása a 2. szakaszban leirt 1.), 2), 3), lépések szerint történik. Az S_i exponensek kétszeres DØ cikluson belüli (3.4) összefüggés szerinti kiszámitása és az S_i értékeknek a (3.4) szerinti, de részletesen kiirt alakban történő kiszámitása lényeges időbeli különbséget jelent. Az exponens kiszámitása három dimenzióesetén kiirt alakban a következő:

$$S = Q(1,1) * P(1) * P(1) + P(2) * (Q(1,2) * P(2) + Q(2,2) * P(2)) + P(3) * (Q(1,2) * P(1) + Q(2,3) * P(2) + Q(3,3) * P(3)),$$

ahol $P(I)$ jelöli a véletlen vektor komponenseit.

Ezzel a felirási móddal további szorzásokat hagyhatunk el a (3.4) képlethez képest. Az 1), 2) és 3) lépéseket külön-külön ezerszer lefuttatva a következő időeredményeket kaptuk

	4 dimenziós	12 dimenziós
1) Véletlen vektor előállitása	0,6 sec	1,3 sec
2) S_i kétszeres $D\phi$ ciklussal való számitása	2,1 sec	11,7 sec
3) exp (S_i) kiszámitása	0,6 sec	0,4 sec
2)* S_i (4.2) szerinti számitása	0,7 sec	3,6 sec
Igy $N = 1000$ esetén a Θ_2 kiszámitásához szükséges idő	1,9 sec	5,3 sec

1. Táblázat

Az $\exp(S_i)$ érték kiszámitásának ideje azért különböző a 4 és a 12 dimenziós esetben, mert a (2.11) szerinti konstans miatt az 1000 eset közül kevesebben kellett ténylegesen kiszámitani az $\exp(S_i)$ értéket. Az 1. táblázatban szereplő idők átlagértékekek, más kezdőértékkel inditva a véletlenszám generátorokat, kissé eltérő időket kaphatunk.

A szubrutinok helyes müködésének ellenőrzését úgy végeztük el, hogy az egységmátrixot vettük korrelációs mátrixnak, ez a független esetnek felel meg. Ilyenkor a keresett $\Phi(\underline{h})$ érték a $\Phi(\underline{h}_i)$ egydimenziós valószinüségek szorzataként adódnak. Közöljük néhány ilyen eset futási eredményeit. 4 dimenzióban, ha a $\underline{h}$ vektor (4,00; 3,25; 4,00; 3,51), akkor a pontos értéke a valószinüségnek 1.00;

	A valószínüség	A hiba 95%-os biztonsággal	N a mintaszám	Idő
ELJAV2 D^{**} felett integrál	1,000	2,02	2000	2,68 sec
ELJAV3 $K^{*} - D^{**}$ felett integrál	0,999	0,0002	4000	4,48 sec
ELJAV4 $K^{*} - D^{**}$ felett integrál	1,000	0,00001	182079	146,7 sec
ELJAV6 $K^{*} - D^{**}$ felett integrál	0,999	0,0002	4000	3,4 sec

2. Táblázat

5 dimenzióban, ha a $\underline{h}$ vektor (4,00; 4,00; 1,22; 0,10; 3,59) akkor a valószinüség pontos értéke 0,4797, a számitott értékek

	A valószszinüség	A hiba 95%-os biztonsággal	N a mintaszám	Idő
ELJAV2	1,000	1,69	2500	3,2 sec
ELJAV3	0,562	0,117	5000	13,8 sec
ELJAV4	0,492	0,104	10751	27,5 sec
ELJAV6	0,483	0,036	78829	130,0 sec

3. Táblázat

A 6 dimenziós független esetben, ha a $\underline{h}$ vektor a következő (4,00; 1,29; 055; 2,70; 3,41; 0,57,) akkor a valószinüség pontos értéke 0,455, a szubrutinok által adott eredmények pedig a következők:

	A valószínüség	A hiba 95%-os biztonsággal	N a mintaszám	Idő
ELJAV2	1,000	0,679	3000	4,73 sec
ELJAV3	0,426	0,188	6000	22,6 sec
ELJAV4	0,417	0,173	7207	21,5 sec
ELJAV6	0,484	0,030	172854	388,6 sec

4. Táblázat

12 dimenziós független esetben, ha a h vektor a következő (1,33; 4,00; 8,57; 0,30; 0,74; 4,00; 0,26; 0,25; 1,38; 1,56; 2,51; 4,00) akkor a valószinüség pontos értéke 0,172 a számitottértékek pedig

	A valószínüség	A hiba 95%-os szinten	N a mintaszám	Idő
ELJAV2	0,048	0,05	6000	26,1 sec
ELJAV3	0,489	0,52	12000	152,7 sec
ELJAV4	0,004	0,62	12530	77 sec

5. Táblázat

A teljesség kedvéért közöljük egy 4 dimenziós és egy 12 dimenziós korrelált esetet is. 4 dimenzióban:

	A valószínüség	A hiba 95%-os szinten	N a mintaszám	Idő
ELJAV2	1,000	2,40	2000	5,4 sec
ELJAV4	0,922	0,04	6000	13,6 sec
ELJAV6	0,921	0,03	14060	30,9 sec

6. Táblázat

12 dimenzióban:

	A valószínüség	A hiba	N mintaszám	Idő
ELJAV2	0,0001	0,0001	6000	25,2 sec
ELJAV3	0,971	0,038	6000	70,3 sec
ELJAV4	0,545	0,656	13000	68 sec

7. Táblázat

A fenti eredményekből is látható, hogy minél nagyobb a keresett valószinüség, annál könnyebben tudjuk a valószinüség értékét a $K^* - D^{**}$ tartomány feletti integrálással kiszámitani. A szubrutinok magasabb dimenzióban (lásd 7. Táblázat első sora) egyes esetekben megbizhatatlan eredményt adnak a kis mintaszám miatt; ebben az esetben a 95%-os biztonsági szinten ± 0.05 nagyságu hibához (az ELJAV3 által kiszámitott) szükséges mintaszám $N = 2\ 000\ 000$ lenne, ehhez körülbelül 110 perc gépidőre lenne szükség. Más korrelációs mátrix és $\underline{h}$ vektor esetén is a ± 0.05 nagyságu hibával terhelt eredmény kiszámitásához a 12 dimenziós esetben 60-120 sec. idő szükséges (kedvező esetben). Igy végkövetkeztetésképpen kimondhatjuk, hogy 8-12 dimenzió esetén Monte Carlo integrálással csak 0.90-nél nagyobb valószinüség esetén tudjuk a valószinüséget meghatározni 2 percnél rövidebb idő alatt, 8 dimenzióig pedig gyakorlatilag minden esetet ki tudunk számitani. Egyes esetekben – attól a problémától függően, amelyben a többdimenziós normális valószinüségekre szükségünk van – az egyes szubrutinok jó felhasználásával gyorsan kaphatunk eredményt. Például Prékopa András STABIL modelljének [3] a futtatása során a négydimenziós valószinüségeket körülbelül 80 alkalommal számitotta ki a gép, összesen mintegy 10 percre volt szükség ehhez.

A várakozásnak megfelelően egy konkrét esetben a mintaszám négyzetgyökével arányosan csökkent a hiba, viszont az irodalom fellelhető utasitásokkal ellentétben a kiszámitáshoz szükséges idő nem lineárisan, hanem gyorsabban növekszik a dimenziószám emelkedésével.

Irodalom

[1] Deák István: Egy sztochasztikus programozási modell számitógépes kiértékelése, MTA Számitástechnikai Központ Közlemények, (1972) 9. p. 33-49.

[2] G. Marsaglia: Random numbers fall mainly in the planes, Proc. Nat. Acad. Sci. USA G 1 (1968) p. 25-28.

[3] Prékopa A., Ganzer S., Deák I., Patyi K.: A STABIL sztochasztikus programozási modell kisérleti alkalmazása a magyar villamosenergiaiparra. Alkalmazott Matematikai Lapok 1(1975) 3-22.

Р Е З Ю М Е

Значения многомерной функции нормального распределения с помощью интегральной техники Монте Карло, полученных в результате расчетов на ЭВМ.

Иштван Деак

В статье дается одна модифицированная Монте Карло интегральная техника для получения конкретных значений многомерной функции нормального распределения с любой коррекционной матрицей, самое большое в 12-и измерениях.

Подпрограммы разработанные автором основаны на базе описанной техники. Результаты машинного прогона изложены в статье.

S U M M A R Y

Computer experiences of the evaluation of the multidimensional normal distribution function by a Monte Carlo integration technique

István Deák

The paper discusses a modified Monte Carlo integration technique to determine the concrete values of the multidimensional normal distribution function with arbitrary correlation matrix at least of 12 dimensions. The subroutines made by the author and based on the described technique were run and their results are presented.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 19/1978.

TALAJMECHANIKAI FÜGGVÉNYEK SZINTVONAL MEGHATÁROZÁSA

Abaffy József – Varga Gyula

1. Bevezetés

A terepen végzett furások eredményeként adódó talajminták anali zálása sok szempontból fontos a geotechnika számára. A minták különböző szempontok szerinti vizsgálata az altalaj összetételére, kémiai, mechanikai, hidrológiai és egyéb tulajdonságaira enged kvalitativ és kvantitativ következtetéseket levonni. A Földmérő és Talajvizsgáló Vállalat, amely a terepen végzett furások adatait és a furásokból kapott talajminták analizálása során nyert számszerü eredményeket regisztrálja, megbizással fordult az MTA SzTAKI-hoz a kb. 70 éves adathalmaz matematikai feldolgozásának modellezésére és a felállitott modellek beprogramozására, valamint az elkészitett programok futtatására.

Minthogy a talajminták analizálása során kapott számszerü értékek a további feldolgozás pontosságigényeinél jóval pontosabbak, ezért a következőkben az adatok pontosságára vonatkozó problémákkal nem foglalkozunk.

A furások és a belőlük eredményül kapott furási paraméterek a furáspont koordinátáitól függő kétváltozós (az időtől való függésüket jelen cikkünkben nem vizsgáljuk) talajmechanikai függvényeket $(f(\underline{x}), \underline{x} \epsilon R^2)$ definiálnak. A furási paraméterek az igy definiált talajmechanikai függvények bizonyos diszkrét pontokban felvett függvényértékeit adják meg. A továbbiakban bizonyos feltevések mellett az ilyen tipusu talajmechanikai függvények viselkedését vizsgáljuk.

2. A feladat megfogalmazása

A fentiekben emlitett $f(\underline{x})$ talajmechanikai függvények vizsgálatához előljáróban néhány olyan tulajdonságot kell róluk feltételezni, amelyek a gyakorlati tapasztalatok és elméleti meggondolások alapján teljesülnek.

a.) Legyen $D = \text{dom } f(\underline{x})$, akkor D általában többszörösen összefüggő.

b.) D belsejében létezik grad $(f(\underline{x}))$ és az folytonos.

c.) $\|\text{grad } f(\underline{x})\|_2 < 0.02\ f(\underline{x})$,

d.) Létezhet $D_1 \subset D$, amelyen

$$f(\underline{x}) = \text{konst} \quad \forall\, \underline{x} \epsilon D_1$$

e.) $f(\underline{x}) \geqslant 0 \qquad \forall\, \underline{x} \epsilon D$

Az ilyen tulajdonságokkal rendelkező talajmechanikai függvények viselkedését a különböző függvényértékekhez tartozó szintvonalak mutatják a legcélszerübben. A továbbiakban feladatul tüzzük ki a talajmechanikai függvények diszkrét pontokban ismert értékei alapján az illető függvények szintvonalainak plotteren történő kirajzolását.

3. Alapvető meggondolások

A következő részben végrehajtandó interpolációs lépések megkönnyitése érdekében a terepen szabálytalanul elhelyezkedő furáspontokban ismert függvényértékek segitségével egy négyzetrács csucspontjaiban szándékozunk a vizsgált talajmechanikai függvény értékeit meghatározni ugy, hogy a rendszertelenül elhelyezkedő furáspontokban ismert függvényértékek összeségéről a rácspontbeli függvényértékek összeségére való áttérés közben az információ veszteség minimális legyen, és bizonyos pontossági követelmények is teljesüljenek. Ez utóbbi szempontot is figyelembe véve határozzuk meg a térképszelvényre eső furáspontok számának ismeretében a rácshálózat élhosszuságát.

Legyen az adott térképszelvényre eső furáspontok száma n.

Legyen

$$d_1 = h/[\sqrt{n}\,],$$

ahol h a négyzetalaku térképszelvény oldalhossza.

Legyen

$$m_1 = [h/d_1] + 1.$$

Ha m_1 páros, akkor legyen

$$m_1 = m_1 + 1.$$

Ezután

(3.1) $m = \min(101, m_1)$ és

(3.2) $d = h/m$

ahol d a rács élhosszusága, m a felosztás.

A (3.1) kifejezésben szereplő 101-es számot az alkalmazott gép gyorsmemóriájának nagysága indokolja.

Megjegyezzük, hogy h szokásos értéke 500 mm, valamint d legkisebb értéke 5 mm.

A négyzetrács elkészitése azt jelenti, hogy alkalmas interpolációs eljárás segitségével m^2 db d élhosszuságu résznégyzet középpontjában kell a talajmechanikai függvény értékeit kiszámitanunk. A kiszámitandó függvényértékek tárolására szolgáló mxm-es mátrix elemeit a számitás megkezdése előtt -1-es értékkel töltjük fel. A rácsszélesség kiszámitásra szolgáló képletekben figyelembe kell venni azt is, hogy n a tényleges és un. fiktiv furáspontok számának összege, ahol a fiktiv pontok az értelmezési tartomány konvex burkán belül kimaradó területek körülhatárolására szolgálnak.

Megjegyezzük, hogy a fiktiv furáspontokkal körülhatárolt területek minimális szélessége

2.5 x d és a határoló szomszédos fiktiv furáspontok egymástól való távolságának kisebbnek kell lenni d-nél. Ha ugyanis a fiktiv furáspontokkal körülhatárolt területrész átmérője kisebb, mint 2.5 x d, akkor ezt a területrészt a program nem észleli, hanem átinterpolálja.

Ha a szomszédos furáspontok távolsága nagyobb d-nél, akkor uj fiktiv furáspontokat kell beiktatni és az m és d meghatározására szolgáló algoritmust ismételni kell, addig, amig a feltétel nem teljesül.

4. A függvényértékek kiszámitása a résznégyzetek középpontjaiban

Legyen a Σ halmaz a következőképpen definiálva:

$$\Sigma = \bigcup_{i=1}^{m^2} S_i$$

ahol

$$S_i = T_i \cup F_i$$

ahol

$T_i = \{ \underline{x}_j : j \epsilon K_i$, ahol K_i az i-edik résznégyzetbe eső tényleges furások indexeinek halmaza $\}$

és

$F_i = \{ \underline{x}_j : j \epsilon J_i$, ahol J_i az i-edik résznégyzetbe eső fiktiv furások indexeinek halmaza $\}$.

Legyen továbbá

$C = \{ \underline{c}_i : i \leqslant m^2$, $\underline{c}_i$ az i-edik résznégyzet középpontjának vektora $\}$,

$\underline{c}_i = (c_{i,1}, c_{i,2})^T$.

Legyen P_i az i-edik résznégyzetbe eső $S_i = T_i \cup F_i$ által meghatározott halmaz poligoniális burkával megadott konvex tartomány, valamint a Q_i-t hasonlóan definiáljuk az F_i halmazra.

Ha

$\underline{c}_i \epsilon P$ és $\underline{c}_i \epsilon Q_i$, akkor

$f(c_i) \equiv -1$

különben ha

$\underline{c}_i \epsilon P_i$, $\overline{\overline{T}}_i \geqslant 4$, $\underline{c}_i \epsilon Q_i$ és

az

$$(4.1) \qquad R_i = \{ \underline{y}_j : \| \underline{y}_j - \underline{c}_i \|_2 < d/4, \quad j \epsilon K_i \}$$

halmaz nem üres,

akkor

$$f(\underline{c}_i) = \sum_{j=1}^{p} f(\underline{y}_j)/p \qquad \text{ahol} \qquad p = \overline{\overline{R}}_i,$$

$$\underline{y}_j \epsilon R_i \qquad \text{és} \qquad \underline{y}_j \epsilon T_i.$$

Ha $\underline{c}_i \epsilon P_i$, $\underline{c}_i \notin Q_i$ és $R_i = \emptyset$

akkor

$$f(\underline{c}_i) = g(\underline{c}_i) \qquad \text{ahol}$$

$$g(\underline{c}_i) = \overline{a} c_{i,1} + \overline{b} c_{i,2} + \overline{c}, \qquad \overline{a},\overline{b},\overline{c} \epsilon R^1, \qquad \text{és}$$

$\overline{a},\overline{b},\overline{c}$ a következő minimum feladat megoldásaként határozható meg:

$$\sum_{l \epsilon K_i} p_l (g(\underline{x}_l) - f(\underline{x}_l))^2 = \min$$

ahol

$$(4.2) \qquad p_l = p_l(r_l) = \frac{1}{r_l} e^{-\alpha r_l}, \qquad r_l = \| x_l - c_i \|_2$$

és

$$\alpha = \frac{n}{h^2} \cdot \ln \sqrt{1 + \sqrt{3}}.$$

A fenti szélsőérték feladat megoldása az alábbi tulhatározott lineáris egyenletrendszer normál megoldásának a megkeresését jelenti:

$$(4.3) \qquad MB\underline{w} = M\underline{z}$$

ahol

$$M = \text{diag}(\sqrt{p_{l_1}}, \ldots, \sqrt{p_{l_\sigma}}), \qquad \sigma = \overline{\overline{K}}_i$$

$$B = \begin{pmatrix} x_{l_1,1} & x_{l_1,2} & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_{l_\sigma,1} & x_{l_\sigma,2} & 1 \end{pmatrix}, \qquad \underline{w} = (\overline{a},\overline{b},\overline{c})^T \quad \text{és}$$

$$\underline{z} = (f(\underline{x}_{l_1}), \ldots f(\underline{x}_{l_\sigma}))^T.$$

Legyen $A \equiv MB$ és $\hat{\underline{z}} = M\underline{Z}$, ekkor (4.2) a következőképpen irható fel:

(4.4) $\quad A\underline{w} = \hat{\underline{z}}$

Ebből $\underline{w} = A^{+}\hat{\underline{z}}$, ahol

A^{+} az A mátrix Moore-Penrose féle inverze.

A (4.1) meghatározásban a d/4 választást az alábbi becsléssel indokoljuk:

Minthogy a tekintett négyzetben fekvő összes pont figyelembevételével a négyzet középpontjában sulyozott átlagolással kiszámitott közelitő függvényérték rosszabb közelitést ad, mint a legkisebb négyzetek módszerével kapott közelités, ezért a d/4 sugaru körben elhelyezkedő pontokkal végrehajtott átlagolással számitott függvényértéknek az előbbi sulyozott átlagolással számitott függvényértékhez viszonyitott relativ hibája felső becslést ad a legkisebb négyzetek módszerével kapott függvényértékhez viszonyitott relativ hibára.

Legyen

$$\bar{f} = \frac{\sum_{j=1}^{k} f(\underline{y}_j)}{k} \qquad \text{ahol} \quad k = \bar{\bar{R}}_i$$

és

$$\bar{\bar{f}} = \frac{\sum_{j=1}^{k} f(\underline{y}_j) + \sum_{l=k+1}^{n} p_l f(\underline{y}_l)}{k + \sum_{l=k+1}^{n} p_l} \leqslant \frac{\sum_{j=1}^{k} f(\underline{y}_j) + \sum_{l=k+1}^{n} p f(\underline{y}_l)}{k + (n-k)p} = \hat{f}$$

$n = \bar{\bar{T}}_i$ és $p = 4/d\, e^{-\alpha d/4}$

Az $\bar{\bar{f}}$-ban és $\hat{f}$ megfelelő részében a sulyokat azért kell elhagynunk, mert

$$p_j \to \infty \qquad \text{ha} \qquad \underline{y}_j \to \underline{c}_i .$$

A relativ hibára az alábbi egyenlőtlenséget irhatjuk fel:

$$\left| \frac{\bar{f} - \hat{f}}{\hat{f}} \right| \leqslant \left| \frac{k + (l-k)p}{k + (l-k)p + p(l-k)\vartheta} - 1 \right| \qquad \text{ahol} \qquad l = \bar{\bar{T}}_i$$

$$(\bar{\bar{T}}_i \leqslant 50) \quad \text{és} \quad \vartheta \geqslant \max_i \left| \frac{\bar{f} - f_j}{\bar{f}} \right|, \qquad j \epsilon k_i \cap M_i'$$

ahol $M_i' = \{ j \epsilon k_i : \| \underline{y}_j - \underline{c}_i \|_2 < d/4 \}$

Figyelembe véve a 2/c feltételt

$$\vartheta \approx 0.32$$

A számitást az $l = 5$, $n = 10^3$, $h = 5.10^2$mm, $d = 5$ mm értékekre elvégezve és a furáspontok elhelyezkedését egyenletes eloszlásunak feltételezve

$$\left| \frac{\bar{f} - \hat{f}}{\hat{f}} \right| < 0.11$$ adódik.

Visszatérve a $\bar{\bar{T}}_i \geqslant 4$ feltételre, megjegyezzük, hogy amennyiben ez nem teljesül, a vizsgálandó résznégyzetet az őt körülvevő 8 (a tartomány határán ez 5-re, illetve sarkoknál 3-ra redukálódik) szomszédos résznégyzettel bővitjük, és a fent leirt eljárást ismételjük. Ugyancsak a fenti eljárást ismételjük akkor is az előbb leirt bővités végrehajtásával, ha az $r(A) < 3$. Ha 1 bővités nem bizonyul elegendőnek, akkor ujabb bővitést végzünk a megnagyobitott négyzetet körülvevő 16 (határoknál kevesebb) résznégyzetek hozzácsatolásával. Ha ez sem ad eredményt, akkor a résznégyzet középpontjában a függvényértéket nem tudjuk kiszámitani.

A legkisebb négyzetek módszerének alkalmazása során felhasznált lineáris közelitő függvények helyett alkalmazhattunk volna kvadratikus függvényeket is. Ehhez az általános kétváltozós kvadratikus függvény 6 együtthatóját kellett volna kiszámitani, szemben a számitásainkban felhasznált lineáris függvény 3 együtthatójával, anélkül, hogy az ily módon kiszámitott függvényértékek pontossága számottevően növekedett volna, az eljárás végrehajtásához szükséges nyolcszoros müveletigény arányában. Ez a tény, tekintetbe véve azt a körülményt, hogy az eljárást maximálisan 10 000 résznégyzetre kell végrehajtani, a gépidő szükségletet jelentősen megnövelné, és ezért ezt a lehetőséget elejtettük.

5. A szintvonalak plotteren történő kirajzolása

Miután kiszámitottuk a szóbanforgó talajmechanikai függvény értékeit a résznégyzetek közzéppontjaiban (nem okvetlenül valamennyiében), a maximális és minimális függvényérték közé eső valamely paraméter értékkel (γ) hozzáfoghatunk a neki megfelelő szintvonal megkereséséhez és kirajzolásához.

Tekintsük a négyzetrács valamelyik résznégyzetét, illetve a résznégyzet csucspontjaiban a talajmechanikai függvény értékeit. Ha a négy csucspont közül egyben nincs kiszámitva a függvényérték (azaz értéke negativ), akkor az illető pontban önkényesen értéket adunk neki ugy, hogy függvényértékként a két mellette fekvő pontban ismert függvényérték összegének és a vele szemben lévő pontban felvett függvényértéknek a különbségét tulajdonitjuk. Ha a négyzet két egymás mellett lévő csucspontjában nincs kiszámitva a függvényérték, akkor csak abban az esetben rajzolhatunk szintvonalat a másik két csucsponton át, ha a függvényérték mindkettőben megegyezik a tekintett paraméterértékkel. Ha két átellenes pontban vagy kettőnél több pontban nincs kiszámitva a függvényérték, akkor az illető négyzeten át nem halad szintvonal.

A továbbiakban tekintsük azt az általános esetet, amikor a négyzet valamennnyi csucspontjában ki van számitva a függvényérték. A szintvonal négyzeten való áthaladásának a feltétele:

$$\min(t_1,t_2,t_3,t_4) \leqslant \gamma \leqslant \max(t_1,t_2,t_3,t_4)$$

ahol t_i, $i = 1, \ldots,$ 4-ig a függvényértékek a résznégyzet csucspontjaiban az óramutató járásával ellenkező irányban véve. A szintvonal áthaladásának elégséges feltétele

$$\min(t_1,t_2,t_3,t_4) < \max(t_1,t_2,t_3,t_4),$$

$$\gamma\epsilon[\min(t_1,t_2,t_3,t_4),\ \max(t_1,t_2,t_3,t_4)],$$

$\gamma = t_i = t_j = t_k$ különböző i,j,k-ra nem áll fenn, valamint $\gamma = t_{i_0} = t_{j_0}$ esetén $\gamma\epsilon[t_{l_1},t_{l_2}]$ ahol az l_1,l_2 indexek különböznek az $i_0 j_0$ indexektől.

Az elégséges feltétel teljesülése esetén a négyzet oldalain lineáris interpolációval megkeressük azokat a pontokat, amelyekben a függvényérték γ-val megegyezik. A figyelembe vehető lehetséges pontok száma 2,3 vagy 4.

Feltételként szabjuk ki azt, hogy a szintvonalak a szomszédos négyzetekbe való áthaladási pontokban lehetőleg egyszer folytonosan diferenciálhatók legyenek, valamint véges sok pontot kivéve bármely szintvonal tetszőleges pontjának elegendően kicsiny környezetébe eső egyéb szintvonalak konvexitása megegyezzék. Ezeknek a feltételeknek a segitségével az elemi függvények közül kiválaszthatunk olyanokat, amelyeknek a menete a tényleges szintvonalakat megközeliti. Az alkalmazott elemi függvények az alábbiak:

1.) szögfüggvények
2.) reciprok függvény
3.) $ax^p + by^p = c \qquad (p\epsilon\,[1,2])$.

A plotteren való kirajzolás tényleges végrehajtásánál figyelembe kellett vennünk azt a körülményt is, hogy a CDC 3300 gépnél rendelkezésünkre álló rajzmező szélessége a térképszelvény felével egyezik meg, és ezért a rajzolást a két fél szelvényre külön-külön kellett végrehajtani. A plotter felesleges tollmozgatásainak elkerülése céljából a vizsgálandó résznégyzetek sorrendjét ugy állapitottuk meg, hogy a bal alsó sorokból elindulva az első sor végéig haladunk, majd ott a második sorba lépve visszatérünk a rajzmező bal szélére és egy sorral feljebb lépve ezt ismételjük. A különböző paraméterértékekhez tartozó szintvonalak különféle vonaltipusuak vagy szinüek lehetnek. Végül a két fél szelvény összeragasztható.

Summary

Determination of the line of levels of soil mechanics functions

József Abaffy – Gyula Varga

The paper investigates the behaviour of soil mechanics functions resulting from drillings on the terrain. It substitutes the drilling points located at haphazard by a sequare grid system and gives an algorithm to the system obtained for the levels of the soil mechanics funtion in question.

Р Е З Ю М Е

Определение линий уровня функций грунтовой механики

Йожеф Абаффи - Дюла Варга

В статье исследуется поведение линий уровня функций грунтовой механики, полученных по полевым буравлениям. Точки буравлений, расположение случайных образцов, заменяются системой квадратных сеток и для полученной системы описывается алгоритм к начертанию линий уровня функции грунтовой механики.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 19/1978.

EGY MÓDSZER ASSEMBLEREK ELŐÁLLITÁSÁRA

Csörnyei Zoltán

Az assemblerek hasonlósága szükségessé és lehetővé tette egy olyan rendszer kidolgozását, amellyel ezek a fordítóprogramok egyszerűen és gyorsan előállithatók. A létrehozott fordítóprogramok két részből állnak: egy törzsből, amelyik közös mindegyik fordítóprogramban, és egy leiró táblából, amely az adott fordítóprogramtól függ. A törzs határozza meg a fordítóprogram jellegét, például, hogy hány passzból áll, milyen a szimbólumkezelése, input-outputja, mig a leiró tábla definiálja a fordítóprogram direktiváit és utasitásait.

A fordítóprogramok gyors előállitásának alapgondolata az, hogy a fordítóprogramok a fordítás folyamán sok hasonló, vagy azonos funkciót végeznek, igy ezek a különböző fordítóprogramokban azonosak is lehetnek. A következőkben ezeket a közös funkciókat határozzuk meg.

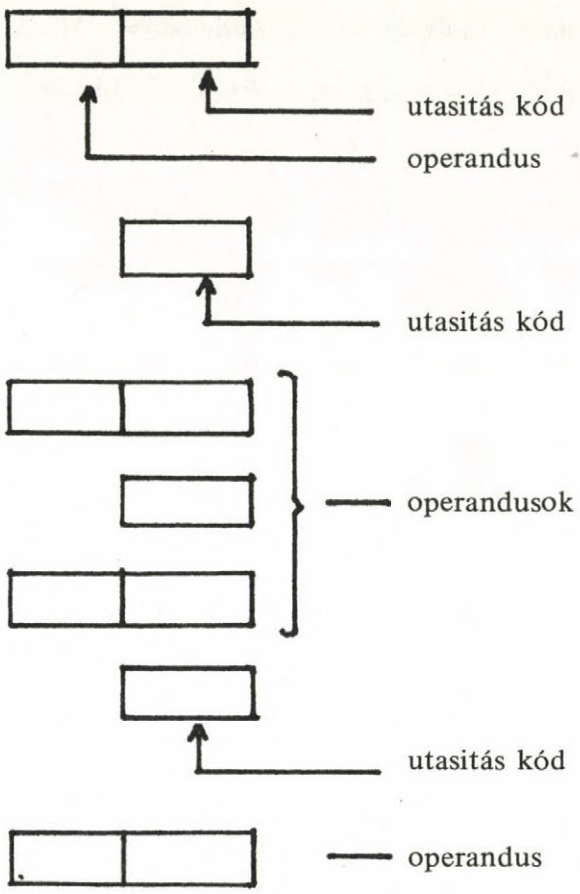
A fordítóprogramnak az a feladata, hogy a szimbólikus nyelven irt programból (forrásprogramból) létrehozzon egy azonnal, vagy csak szerkesztés után futtatható programot (tárgyprogramot). A tárgyprogram lehet gépi kódú, vagy valamilyen interpreter által végrehajtandó program. A forrásprogramok közös tulajdonságait határozzuk meg először.

A forrásprogramok közös tulajdonságai

A forrásprogram sorokból áll, minden egyes sor egy utasitás szimbólikus nyelvű leirása. Előfordulhat, hogy egy sor az adathordozón nem fér el fizikailag egy sorban, például, ha egy utasitásnak nagyon sok operandusa van; ilyenkor biztositani kell a sor tördelhetőségét, egy speciális jel (pl. <) szolgálhat a törés jelzésére.

A sor karakterekből álló karakterlánc, a sor végét egy, vagy két speciális karakter (kocsivissza, soremelés, új-sor, stb.) jelzi. Az egy ilyen sorból leforditott bináris információ két részből áll: az utasitás kódjából és az operandusokból. Az utasitáskód terjedelme két-három bittől két byte-ig terjed, az operandus az utasitástól függően lehet csak egy-két bit, vagy akár három-négy, vagy még több byte is.

A forrásprogram sorai zónákra tagolhatók, egy sor négy zónából áll: cimke, utasitás, operandus és komment zóna.



A cimkezóna lehet üres, vagy egy nevet (szimbólumot, cimkét) tartalmaz, amely az utasitássort azonositja. Általában ez a név az utasitás kód memóriarekeszének szimbolikus neve. A cimkezónába irható szimbólumok betüvel kezdődő, betüket és számokat tartalmazható karaktersorozatok, a karaktersorozatok hosszát általában maximálják. A szimbólumokban speciális jelek is előfordulhatnak (pl. : vagy @), ezeket célszerű a programozónak a szimbólum tipusjelzésére használni, feltéve, hogy a forditóprogram ezt nem követeli meg.

Az utasitászónában helyezkedik el az utasitás szimbólikus kódja (mnemonik). A tárgyprogram szempontjából a forrásprogram utasitásait három csoportra oszthatjuk. Az első csoportba azok az utasitások tartoznak, amelyek a sor lefordításakor a tárgyprogramba információt adnak. Ezek a gépi kódú utasitások és az adat (konstans) megadások. A második csoport utasitásai közvetlenül nem adnak információt, ilyen pl. egy szimbólum helyének, vagy értékének definiálása. A harmadik csoportba a forditóprogramnak szóló utasitások, a direktivák tartoznak.

Az utasitászónában lévő információtól függ, hogy a többi zónának kell, lehet, vagy nem lehet információt tartalmaznia.

Az operandus mezőbe kerülnek az operandusok. Több tagból is állhatnak az operandus tagjait legalább egy karakterrel (pl. vessző, vagy szóköz) kell elválasztani. Az itt lévő információból készülnek az utasitáskód utáni bináris értékek. A forrásprogram operandus tagjai lehetnek szimbólumok, számok, kifejezések. A kifejezés müveleti jelekkel összekapcsolt szimbólumok és számok sorozata, hossza álatlában nincs korlátozva. Az operandus tagjainak számát az utasitászónában lévő utasitás határozza meg.

A komment zóna kezdőkaraktere általában egy speciális karakter (pl. * vagy ;). A komment zóna a sor elején is kezdődhet, az ilyen sort komment sornak nevezzük. A komment a tárgyprogramba információt nem ad.

A zónák kötött, vagy szabad formában követhetik egymást. A kötött formában a zónák mindig egy előirt pozición kezdődnek, mig a szabad formában a zónák között egy terminátor karakter (általában szóköz, vagy tabulátor) helyezkedik el.

A forditóprogramok közös tulajdonságai

A forditóprogramok azonos tulajdonságait két szempontból vizsgáljuk. Az első a forditóprogram szolgáltatása, amelyet egy dialógussal lehet definiálni. A forditóprogramok háromféle eredményet adnak:

- bináris információ (tárgyprogram)
- lista
- cross-referencia lista.

Az utóbbi kettővel általában együtt jár a tárgyprogram készitése. A felhasználó a dialógus folyamán kijelöli, hogy milyen szolgáltatást kér; ez a forditóprogramban flagek állitását jelenti. A forditóprogramok általában mindig előállitják mindhárom output-ot, információhordozón való megjelenésüket tiltják, vagy engedélyezik a flagek. A dialógus programjára csak a forditás elején van szükség, ez a terület a forditás alatt még felhasználható.

A másik szempont a forditóprogramok müködése. A forditóprogramok kétpasszosak, bár néha lehet látni egypasszos forditókat is. Minden forditóprogram tartalmaz egy szervező részt, amely többnyire más funkcióval egybeépitve (pl. szimbólumtábla kezelés) a passzok szervezését végzi. A passz sorszám is egy flagben tárolódik, igy a feldolgozó programok mindkét passzban azonosak. A passztól függő funkciót a flag vizsgálatával választják ki.

A forditóprogramok a forrásprogramot soronként dolgozzák fel. Az első teendő a forrásprogram (következő) sorának beolvasása a háttértárolóról. Már a beolvasáskor átalakithatja a sort, és előkészitheti a listázásra. Kiszűri a felesleges karaktereket (szóközök, javitások), tabulál, stb. Az input feladata a sor hosszának megállapitása is.

Ha a beolvasott sor komment sor, azonnal indulhat a listázása, további munka az ilyen sorral nincs.

A nem-komment soroknál az első feladat az utasitás zónában lévő karaktersorozat azonositása. A lehetséges utasitásokat és a hozzájukrendelt kódokat általában egy táblázat tartalmazza. Ez a táblázat tartalmaz arra is utalást, hogy a többi zónában milyen információt kell keresni, azaz hogyan kell azokat feldolgozni.

Az utasitás azonositása után a cimke feldolgozása következik. Két tipusú cimke használata az általános: az egyik az utasitáskód memória rekeszének neve, a másik az értékadás. A két tipusú cimkefeldolgozó programnak használnia kell a szimbólumtáblákat is, hiszen a cimkét oda fel kell irni, vagy a multidefiniáltságot meg kell határozni. A szimbólumtáblát kezelő programnak a következő belépései a szokásosak:

– egy adott szimbólum felirása,

– egy adott szimbólum keresése,

– egy adott szimbólum törlése.

Az első mindig tartalmazza a második funkciót is. Az első a cimke, a második az operandus feldolgozásakor szerepel. Mivel kétpasszos forditóprogramnál két szimbólumtábla (predefinit és postdefinit cimkéknek) használatos, a felirás és keresés is többféle lehet. A harmadik a program, vagy a program egy részének végekor müködik.

Az operandus feldolgozása általában a legbonyolultabb. Ha az utasitástól függően nincs más korlátozás, az operandus tagjai kifejezések. Egy kifejezés feldolgozásakor a szimbólumtábla-kezelő, számátalakitó és természetesen a kifejezésben lévő müveleteket végző programokat kell használni. Mennél több kényelmet biztosit a forditóprogram a felhasználónak, annál bonyolultabb a kifejezésfeldolgozás a forditóprogramban.

Az operandus feldolgozásával a beolvasott sor feldolgozása is befejeződött. Az első passz csak a szimbolumtáblát és különböző pointereket kezel, a második passzban pedig a sor feldolgozása után a kijelölt output-okat kell végrehajtani. A tárgyprogrammal különösebb teendő nincs, ez egy háttértárolón tárolható. A lista többféle lehet. Általában négy részből áll a lista egy sora:

– az utasitássor sorszáma a forrásprogramban,

– memória cim,

– a memória cim tartalma,

– a forrásnyelvi utasitássor.

Az utasitássorban lévő utasitástól függ, hogy a négy rész közül a listában melyik szerepel.

A fordítóprogramok hibajelzései kétfélék: az egyik tipus a sor feldolgozása folyamán detektált hibákat jelzi, a másik a fordítás befejezésekor ismeretlen szimbólumokat irja. Mivel a hibajelző programot a fordítóprogram majdnem minden részéből hivhatja, a hibajelzést egy vagy több szubrutin végzi.

A fordítás befejezése után több szolgáltatás biztositja a felhasználó kényelmét, a fordítóprogramok kiirhatják a tárgyprogram hosszát, központi memóriában való elhelyezhetőségét, a szegmensek listáját, stb.

A fordítóprogram törzse

A fenti részben nem beszéltünk arról, hogy a fordítóprogram a forrásprogramot milyen tárgyprogrammá forditja, azaz az adott fordítóprogram milyen utasításkészletet ismer fel, milyen operandusai lehetnek, mi az operandusok sorrendje, milyen utasításoknak nem lehet cimkéje. Ez csak az adott forrásnyelv utasításainak definiciójától függ.

Még arról sem volt szó, hogy milyen a szimbólumábrázolás, milyen a forrásprogram struktúrája. Ez is a forrásnyelv definiciójától függ, de előfordulhat, hogy ez már több forrásnyelv definiciójában azonos.

A fordítóprogramok fenti jellemzéséből, valamint ezekből az észrevételekből kitünik, hogy a fordítóprogramok alapvetően két részre oszthatók. Az egyik rész forrásnyelv-specifikus a másik fordítóprogram-specifikus. A fordítóprogram-specifikus rész több fordítóprogramban lehet azonos. Ezt a részt nevezzük a fordítóprogram törzsének. A másik rész a forrásnyelv utasításait irja le, ez a leiró tábla. Ha a törzs tartalmaz egy interpretert, a leiró rész interpretativ leirású lehet, amivel lehetővé válik a forrásprogram utasításainak kis memória-igényü leirása.

Ha a törzs jól szegmentált, feladat-orientált felépitésü, akkor az egymástól lényegesen különböző forrásprogramok fordítóprogramjai is könnyen elkészithetők, feltéve, hogy ezek a fordítóprogramok is rendelkeznek a korábban leirt közös tulajdonságokkal. Egy törzsből a megfelelő módositások elvégzésével, de a törzs szerkezetének megtartásával egy egészen új tipusú fordítóprogram hozható létre.

A fordítóprogram törzse tartalmazza azokat a funkciókat megvalósító programokat, amelyeket a közös tulajdonságukról szóló részben irtunk le. Ezeket a programokat célszerü úgy irni, hogy az interpretativ leirás moduljai legyenek, azaz a leiró táblában egy kóddal lehessen rájuk hivatkozni. Az interpreter a leiró tábla kódjait értelmezi és inditja a kódhoz tartozó programot.

A leiró tábla

A fordítóprogramnak ez a része tartalmazza a fordítás szabályait, ez a rész az utasítások leirásából áll.

Itt vannak a mnemonikok és a hozzájuk tartozó szintaktikus szabályok. A leirás interpretativ, egy egy-byte-os interpretert választva a leirás rövid, kisebb mint a törzs egynegyede.

A forrásprogramban a felhasznált különböző utasitások darabszámának eloszlása nem egyenletes, igy lehet egy utasitás szimbólikus nevének keresése lineáris, ha a tábla elejére a leggyakrabban használt, végére a legritkábban használt utasitások kerülnek. Ha a keresést a tábla elején kezdjük, az utasitás felismeréséhez szükséges összehasonlitások száma kevesebb, mint rendezetlen tábla esetén. Nem-lineáris keresési algoritmussal lehet az összehasonlitások számát csökkenteni, de legtöbbször ez a program nagyságát növeli. Igy ha az utasitások száma nem nagy, a lineáris keresés a fordítás idejét lényegesen nem növeli.

A leiró táblában az utasitások leirása nem egyforma hosszúságú, ezért a kereséshez a táblában a láncolási cimeket el kell helyezni, a láncolási cim lehet pl. a következő utasitás leirásától való távolság is.

A forrásprogram utasitásainak egy jellemzője, hogy sok utasitás és szintaktikájuk csak a mnemonikban és az utasitáskódban különbözik egymástól. Ezeket az utasitásokat természetesen célszerü a leiró táblában úgy leirni, hogy csak a mnemonik felismerésének és a bináris kód megadásának leirása különbözzön, és a leirás többi része már csak egy helyen szerepeljen. Ehhez egy olyan interpretativ utasitás kell, amelyik a leiró táblában ugrásokat hajt végre.

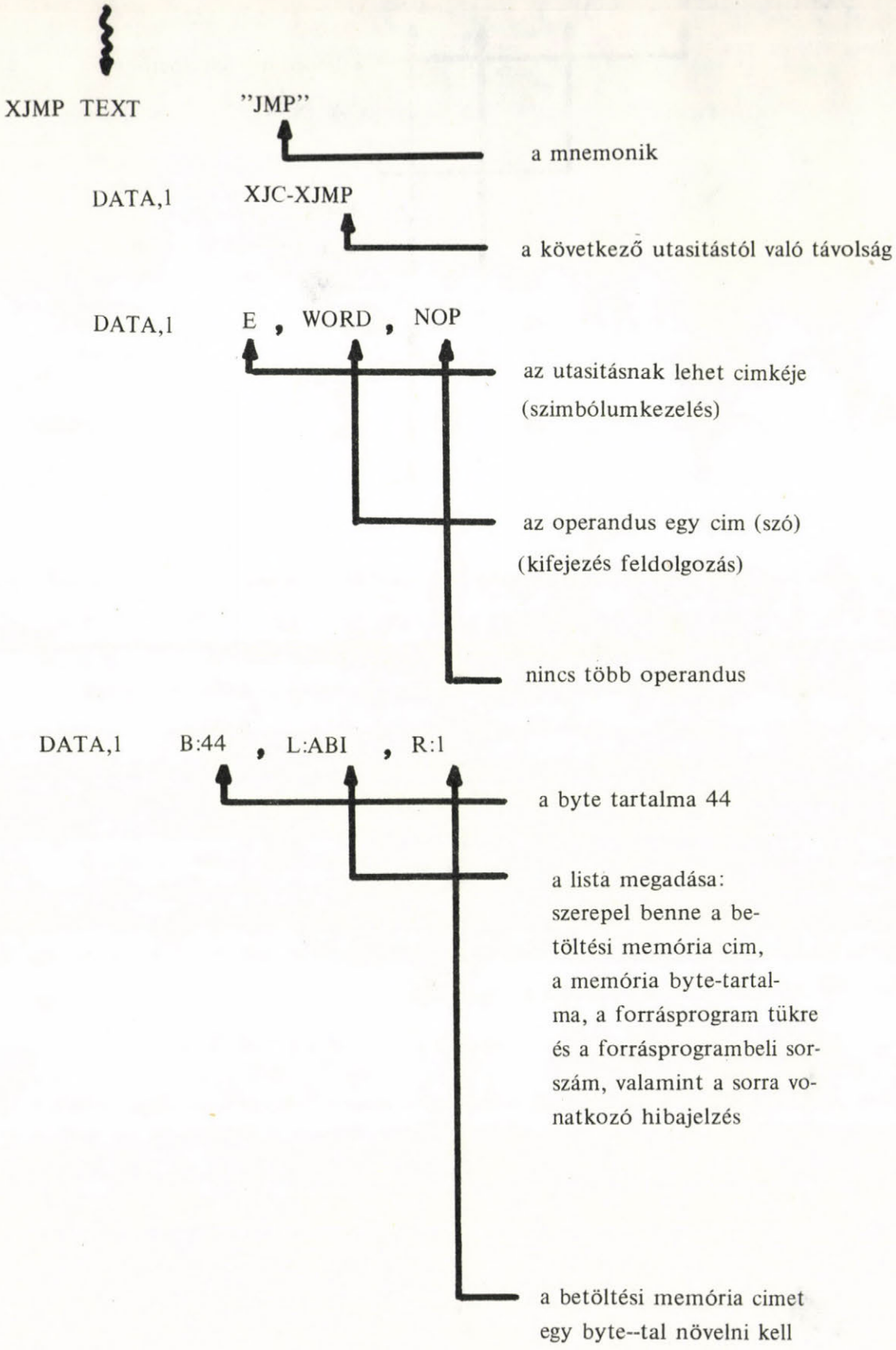
A forditóprogramok

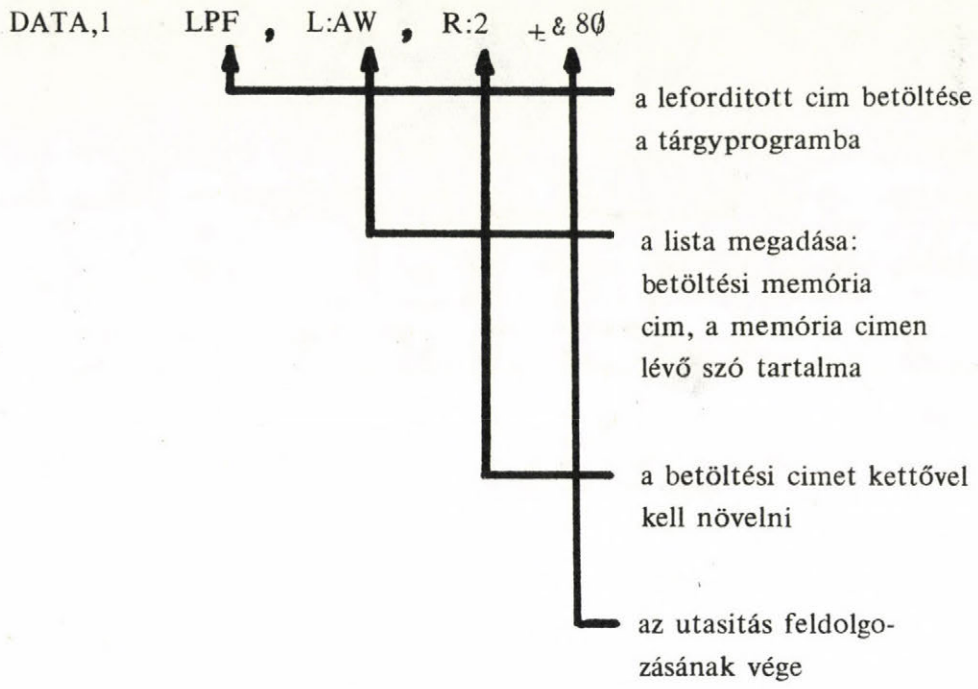
Az előállitott forditóprogramok memória mérete és a fordítás ideje lehet, hogy nem optimális. Mégis ami a módszer mellett szól, az a könnyen áttekinthető szerkezet és a gyors ellőállithatóság. A forditóprogram módositása, bővitése, szükitése sem okoz problémát, akár az utasitások leirását, akár a program törzsét kell megváltoztatni, elsősorban a tagolt, áttekinthető felépités miatt.

Ezzel a módszerrel már három forditóprogram készült el. Mindhárom R-10 számitógépen müködik. Az egyik egy R-10-en futó program fordítására, a másik kettő az R-10-en szimulált számitógépek tárgyprogramjainak előállitására szolgál.

Az első program a PROCESS-2/8K autókodja. A második az MTA SzTAKI-ban kifejlesztett MFB (Mikroprocesszoros Folyamatirányitó Berendezés) assemblere, a harmadik az MFB autokódja. Az utóbbi kettő program törzse is azonos, csak a leiró táblák különböznek. Az első és a másik kettő forditóprogram törzseinek felépitése hasonló, a törzs egyes moduljaiban van csak eltérés. Más pl. a szimbólumok formátuma, a lebegőpontos számábrázolás, igy az ezek feldolgozását végző modulok különbözőek.

Egy példával szemléltetjük az ilyen tipusú forditóprogramok leirását. A példában az MFB assembler leiró táblájának egy része szerepel, a JMP utasitás leirása. Az utasitás három byte-os, az első byte az utasitáskód, értéke 44, a második és harmadik byte egy memóriacim.


XJMP TEXT
"JMP"
a mnemonik
DATA,1
XJC-XJMP
a következő utasitástól való távolság
DATA,1
E , WORD , NOP
az utasitásnak lehet cimkéje
(szimbólumkezelés)
az operandus egy cim (szó)
(kifejezés feldolgozás)
nincs több operandus
DATA,1
B:44 , L:ABI , R:1
a byte tartalma 44
a lista megadása:
szerepel benne a betöltési memória cim,
a memória byte-tartalma, a forrásprogram tükre
és a forrásprogrambeli sorszám, valamint a sorra vonatkozó hibajelzés
a betöltési memória cimet
egy byte--tal növelni kell


DATA,1 LPF , L:AW , R:2 +& 8Ø
a leforditott cim betöltése a tárgyprogramba
a lista megadása: betöltési memória cim, a memória cimen lévő szó tartalma
a betöltési cimet kettővel kell növelni
az utasitás feldolgozásának vége


XJC TEXT "JC"
a következő utasitás

Irodalom

[1] Donovan, J.J.: Systems Programming, McGraw-Hill Book Company, New York, 1972.

[2] Gries, D.: Compliler Construction for Digital Computers, John Wiley and Sons, Inc. New York, 1971.

[3] Lee, J.A.N.: The Anatomy of a Compiler, Reinhold Publishing Corp., New York, 1967.

[4] Varga László: Rendszerprogramozás, ELTE TTK, Tankönyvkiadó, Budapest, 1975.

Summary

A method for the generation of assemblers

Zoltán Csörnyei

Similarities to be found in assemblers made the development of a system, intended to be used for the simple and rapid generation of these assembler programs necessary and possible. Assemblers produced this way consist of two parts: the body, which is common in every single assembler, and the description table, which depends on the nature of the given assembler. The main features of the assembler are determined by the body, for example the number of passes needed, the handling of symbols, its inputs and outputs etc., while the directives and the assembler's repertory of instructions are defined by the description table.

Р Е З Ю М Е

Один метод для разработки ассамблеров

Золтан Чернеи

Сходство ассамблеров привело к необходимости и возможности разработки такой системы, спомощью которой эти транслирующие программы разрабатываются просто и без больших временных затрат. Выполненные таким образом трасляторы состоят из двух частей: из общей части для всех трансляторов и из таблицы описания характеризующей данную транслирующую программу. Общая часть определяет характер транслятора, например, число этапов трансляций, обращение символами, вход и выход, а таблицей описания определяются директивы и команды ассамблера.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 19/1978.

МОДЕЛИРОВАНИЕ АДАПТИВНОЙ СИСТЕМЫ С ИДЕНТИФИКАТОРОМ

Чадеев **В.М.** (Институт Проблем Управления Академии Наук СССР),

Алмаши Г. (Исследовательский Институт Вычислительной Техники и Автоматизации Венгерской Академии Наук).

I. Введение

Адаптивные системы управления с идентификатором в цепи обратной связи (АСИ) имеют широкие перспективы применения в промышленности, особенно для управления крупными объектами, например, - прокатными станами. Это связано с тем, что такие системы не требуют предварительного точного знания модели объекта, а строят ее в процессе работы. Наличие модели позволяет осуществлять управление по возмущению для нестационарных объектов, а крупные объекты, как правило, нестационарны.

В настоящее время достаточно хорошо разработана теория адаптивной идентификации в разомкнутых системах [I,2]. В литературе имеются описания работающих в промышленности АСИ [3,4]. Многие вопросы в этих АСИ решены эвристически, в частности, - вопросы адаптивной идентификации в замкнутой системе. Получены также весьма общие результаты о гиперустойчивости АСИ в смысле Попова [5,6]. Однако, многие важные для практики вопросы оказываются слишком сложными, чтобы их можно было исследовать теоретически. Это влияние ограничений на качество идентификации, определение целесообразного уровня идентификации при переходе к управлению и т.п.

В настоящей работе предпринимается попытка исследования АСИ с помощью статистического моделирования.

2. Описание АСИ и постановка задачи.

Рассмотрим стандартную схему АСИ, показанную на рис. I. На входе объекта О действует вектор случайных возмущений

$$x_N^T = (x_{1,N}\ x_{2,N} \cdots x_{n-1,N}) \qquad (I)$$

где ^{T} - знак транспонирования,
$x_{i,N}$ - i -я компонента вектора x_N в момент времени N,
и скалярное управляющее воздействие u_N. Введем в рассмотрение также вектор входа

$$X_N^T = (x_{1,N}, \ldots, x_{n-1,N}; u_N) \qquad (2)$$

Скалярный измеримый выход объекта y_N связан с входными переменными уравнением

$$y_N = H_N^T X_N + \varepsilon_N , \qquad (3)$$

где ε_N - шум измерения,
H_N - вектор неизвестных параметров объекта размерности n

$$H_N^T = (h_{1,N}, h_{2,N}, \ldots, h_{n-1,N}, h_{n,N}). \qquad (4)$$

Входные x_N и выходные y_N переменные объекта поступают в идентификатор I, где с помощью алгоритма идентификации преобразуются в оценки объекта

$$K_{N-1}^T = (k_{1,N-1}, k_{2,N-1}, \ldots, k_{n,N-1}). \qquad (5)$$

В дальнейшем будем предполагать, что структуры модели и объекта совпадают и модель описывается уравнением

$$y_N^{\text{м}} = K_{N-1}^T X_N \qquad (6)$$

где y_N^* – оценка выхода объекта (выход модели),

K_{N-1} – оценки параметров объекта H_N .

Оценки параметров объекта K_{N-1} поступают в управляющую часть системы C , где используются для выработки управляющего воздействия u_N .

Для системы стабилизации, которую впоследствии мы только и будем рассматривать, уравнение управления имеет вид

$$K_{N-1}^T X_N = 0 \qquad (7)$$

Это уравнение соответствует случаю, когда стабилизируется нулевое значение выхода. Это наиболее трудный для идентификации случай и мы ограничим свое рассмотрение только им. Дело в том, что уравнение (7) приводит к линейной зависимости между компонентами входного вектора. По этой причине в замкнутой системе невозможно использовать обычный МНК (метод наименьших квадратов), так как матрица исходных данных будет вырождена. Решая уравнение (7), получим явное выражение для управления

$$u_N = - \frac{1}{k_{n,N-1}} k_{N-1}^T x_N , \qquad (8)$$

где

$$k_{N-1}^T = (k_{1,N-1}),\ldots,k_{n-1,N-1}). \qquad (9)$$

Для идентификации объекта как в замкнутом, так и в разомкнутом состоянии, будем использовать одношаговый алгоритм идентификации

$$K_N = K_{N-1} + \Delta_N \cdot \dot{X}_N , \qquad (10)$$

где скаляр Δ_N задается формулой

$$\Delta_N = \frac{y_N - y_N^*}{\gamma + x_N^T x_N}, \qquad (II)$$

а константа γ выбирается заранее, в зависимости от уровня шума [2].

Алгоритм (I0), (II) подробно исследован в [2]. Он обеспечивает абсолютную сходимость во всей области начальных оценок. В частности, доказано, что если компоненты вектора x_N представляют собой независимые случайные величины с нулевыми математическими ожиданиями и одинаковыми дисперсиями, а сами векторы δ -коррелированы, γ равно нулю, то ошибка идентификации стационарного объекта (H_N = const.) будет уменьшаться по экспоненте в соответствии с формулой

$$\theta_N^T \theta_N = (1 - \frac{1}{n})^N \theta_0^T \theta_0, \qquad (I2)$$

где вектор ошибки идентификации θ_N равен

$$\theta_N = H_N - K_N, \qquad (I3)$$

n - размерность объекта.
Формула (I2) верна, когда H_N - константа.

Однако, в замкнутой системе скорость сходимости, по-видимому, будет меньше. Оценка скорости сходимости в замкнутой системе представляет значительные трудности.

Наша задача будет состоять в получении верхней границы скорости сходимости алгоритма (I0) в замкнутой системе и в получении путем моделирования зависимостей средней ошибки управления и средней ошибки идентификации от параметров системы управления.

3. Максимальная скорость сходимости.

Вычислим оценку максимальной скорости сходимости алгоритма (I0), (II) в замкнутой стационарной системе без помех. Как известно [2], ошибка идентификации за один шаг изменяется в соответствии с формулой

$$\theta_N^T\theta = \theta_{N-1}^T \theta_{N-1} - \frac{(\theta_{N-1}^T X_N)^2}{X_N^T X_N} \qquad (I4)$$

В замкнутой системе ошибка идентификации тоже будет изменяться в соответствии с этой формулой; однако, компоненты входного вектора будут связаны между собой дополнительно уравнением (7). Подставляя значение u_N из (8) в (I4), получим

$$\theta_N^T\theta_N = \theta_{N-1}^T \theta_{N-1} - \frac{(c^T x_N)^2}{a^2 x_N^T x_N = (k_{N-1}^T x_N)^2} \qquad (I5)$$

где

$$c = (c_1,\ldots,c_{n-1}),$$

$$c_i = k_{n,N-1}\,\theta_{i,N-1} - k_{i,N-1}\,\theta_{n,N-1}, \qquad (I6)$$

$$a \equiv k_{n,N-1}.$$

Вычисление математического ожидания ошибки $\theta_N^T\theta_N$ даже для случая, когда входные переменные статистически независимы, очень трудно. Поэтому вычислим значение ошибки, соответствующее наилучшему распределению компонент вектора x_N. Будем выбирать вектор x_N таким образом, чтобы минимизировать (I5). Решая систему уравнений

$$\frac{\partial\, \theta_N^T \theta_N}{\partial\, x_{i,N}} = 0, \quad (i=1,2,\ldots,n-1). \qquad (I7)$$

получим, что значения компонент вектора x_N , доставляющие минимум ошибке $\theta_N^T \theta_N$, равны

$$x_{j,N} = b \frac{h_{j,N} K_{N-1}^T K_{N-1} - k_{j,N} H_N^T K_{N-1}}{h_{1,N} K_{N-1}^T K_{N-1} - k_{1,N} H^T K_{N-1}}, \quad (18)$$

$$(j=1,2,\ldots,n-1)$$

где b - произвольная, но не равная нулю константа.

Ошибка идентификации будет изменяться в соответствии с формулой

$$\theta_N^T \theta_N = \frac{G^2}{K_{N-1}^T K_{N-1}}, \quad (19)$$

где

$$G = H^T K_o - K_o^T K . \quad (20)$$

Поскольку ошибка идентификации $\theta_N^T \theta_N$ может только монотонно уменьшаться, то из (19) следует, что норма вектора оценок в замкнутой системе может только увеличиваться. Отсюда, в частности, следует, что в замкнутой системе ошибка идентификации, хотя и уменьшается монотонно, но не обязательно стремится к нулю. Ошибка идентификации была бы равна нулю (точнее могла бы быть), если начальная оценка лежит на сфере

$$K_o^T K_o - K_o^T H = 0$$

В заключение подчеркнем еще раз, что оптимальный алгоритм идентификации (18) конечно же не может быть реализован на практике. Смысл полученных результатов состоит в оценке предельных возможностей идентификации при случайном совпадении наиболее благоприятных условий.

4. Моделирование АСИ.

Моделирование АСИ осуществлялось, в основном, по схеме рис. I, но дополнительно были введены дополнительные элементы. Были введены ограничения на области изменения u_N и K_N, исследовался алгоритм автоматического перехода из режима управления в режим обучения и обратно. Управление включалось, если выполнялся критерий

$$A_{crit} \geqslant A, \tag{2I}$$

где

$$A = D^{*}\Delta Y / D^{*}Y , \tag{22}$$

где, в свою очередь,

$D^{*}\Delta Y$ – оценка дисперсии ошибки предсказания,
$D^{*}Y$ – дисперсия выхода объекта.

Исследовалось влияние уровня переключения A_{crit} на среднюю ошибку идентификации и управления. Были рассмотрены два алгоритма повторного использования данных. Подробнее программа моделирования описана в следующем параграфе.

5. Программа моделирования.

При моделировании предполагалось, что процесс изменения параметров объекта описывается стохастическим уравнением авторегрессии первого порядка

$$H_N = (E - Z_H)H_{N-1} + Z_H \xi_{H,N} ,$$

где E – единичная матрица,
Z_H – матрица весов процесса авторегрессии,
$\xi_{H,N}$ – вектор независимых случайных переменных с нулевым средним.

Изменение входных переменных при моделировании описывалось стохастическим уравнением авторегрессии первого порядка

$$x_N = (E-Z_x)x_{N-1} + Z_x \cdot \xi_{x,N},$$

где значения переменных те же самые, что и выше.

Автоматический переход от управления к чистой идентификации и обратно осуществлялся в соответствии с условием

$$A \geqslant A_{crit}.$$

Если A удовлетворяет этому условию, то система от управления переходит к идентификации, а управляющее воздействие попеременно принимает значения

$$u_N = \text{lastu} + \Delta u$$

и

$$u_N = \text{lastu} - \Delta u,$$

где lastu - последнее значение управляющего воздействия при работе системы в режиме управления. Значение Δu задается в программе.

Программа моделирования предусматривает возможность повторной идентификации на основе одношагового алгоритма, когда число тактов работы системы в режиме управления превысит заданное число тактов l_{max}. Разработаны два варианта повторного использования данных. В первом варианте память заполняется в режиме обучения и делается пустой после использования данных из нее. Во втором варианте память при использовании данных сохраняется.

Программа позволяет вводить ограничения на оценки параметров объекта $k_{i,N}$ и управляющее воздействие u_N.

Логическая схема программы показана на рис. 2.

6. Результаты моделирования.

Следует отметить, что результаты данного параграфа необходимо рассматривать как предварительные, поскольку дополнительно должно быть исследовано влияние на результаты различных распределений исходных данных.

Однако в результате моделирования выявлены некоторые важные для практики особенности АСИ, которые могут быть использованы уже сейчас.

Дисперсия помехи выхода 2500, ограничения на управление ± I0. Вектор входов x_N имел нулевое математическое ожидание и одинаковые дисперсии компонент, независимых между собой и во времени, значение z_H при моделировании было равно 0,0IE.

Подробнее мы остановимся на двух результатах.

а). Влияние уровня переключения A_{crit}.

Были исследованы два случая. Первый случай, когда ограничения на оценки параметров были очень велики ± I000 % для каналов возмущений и + I000 % ; - 99 % - для канала управления. В этом случае, как видно из рис. 3, средняя ошибка стабилизации имеет минимум при некотором среднем значении уровня переключения A_{crit}, а ошибка идентификации медленно уменьшается с увеличением A_{crit}.

Второй случай, когда ограничения на оценки параметров достаточно узки ± 50 %. В этом случае картина радикально меняется. Ошибка стабилизации с увеличением A_{crit} сначала увеличивается, а затем уменьшается и, достигнув после единицы некоторой постоянной величины, дальше не изменяется. Ошибка идентификации монотонно увеличивается с увеличением A_{crit} (рис. 4).

Таким образом, в системе управления, когда параметры объекта заранее неизвестны, уровень переключения должен быть выбран меньше единицы; если же параметры объекта заранее известны достаточно точно (± 50 % как в нашем случае), то управление должно включаться практически сразу.

б). Влияние повторного использования данных.

Одношаговый алгоритм идентификации (I0), (II) почти всегда, за исключением экзотического случая ортогональности, только частично использует информацию, содержащуюся в исходных данных. Поэтому повторное их использование должно было улучшить качество идентификации. Так оно и оказалось, однако одновременно с этим увеличилась ошибка стабилизации.

Таким образом, в замкнутой системе, где процессы идентификации и управления связаны, повторное использование данных, улучшая идентификацию, ухудшает управление. Если критерием является ошибка стабилизации, то использовать старые данные повторно не слудет, если ошибка идентификации, - то следует.

Литература

1. Перельман И.И.: Текущий регрессионный анализ и его применение в некоторых задачах автоматического управления. Изд. АН СССР, ОТН. "Энергетика и автоматика", 1960, № 2.

2. Райбман Н.С., Чадеев В.М.: Построение моделей процессов производства. Изд-во "Энергия", М., 1975.

3. Адаптивное управление точностью труб. Под ред. Данилова Ф.А. и Райбмана И.С. М., "Металлургия", 1973.

4. Rajbman N.S., Chadeev V.M. Tibe Rolling mills adaptive control systen. - Preprints of 3-rd IFAC/IFIP Conference on Digital Computer Applicafions of Process Control. Helsinki, Finland. 1971.

5. Juan M. Martin - Sanches. A new solufion of Control. Proc. IEEE, v. 64, № 8, August 1976.

6. Ципкин Я.З.: Основы теории обучающихся систем. М., "Наука", 1970.

Summary

Simulation of an adaptive system with identification

W.M. Chadeev – Gedeon Almásy

The system is composed

– of a linear multi-input, single-output static process with slowly varying parameters,
– of a simple adaptive identification algorithm, and
– of a control law, adjusting a simple control variable in order to obtain zero output.

A heuristic strategy against model degeneration is investigated by simulation.

Összefoglaló

Egy adaptiv identifikátoros rendszer szimulációs vizsgálata

Csagyejev, V.M. – Almásy Gedeon

A rendszer összetevői:

– egy lineáris több-bemenetü, egykimenetü lassan változó paraméterü statikus folyamat,
– egy egyszerü adaptiv identifikációs algoritmus, és
– szabályozás zérus kimenetre egy beavatkozó változóval.

A dolgozat egy modelelfajulás elleni heurisztikus stratégia szimulációs vizsgálatát ismerteti.

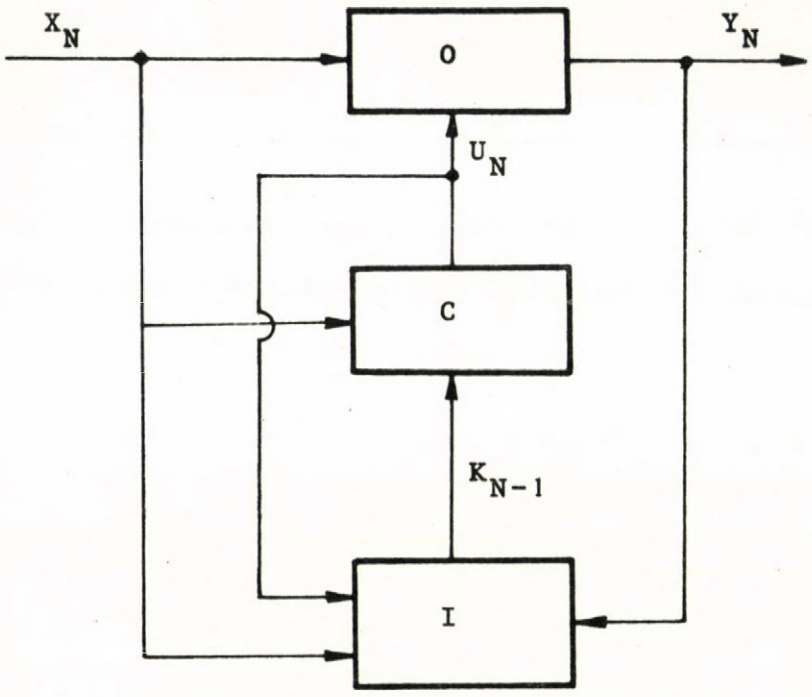


Рис. 1.

Блок-схема адаптивной системы с идентификатором (АСИ)

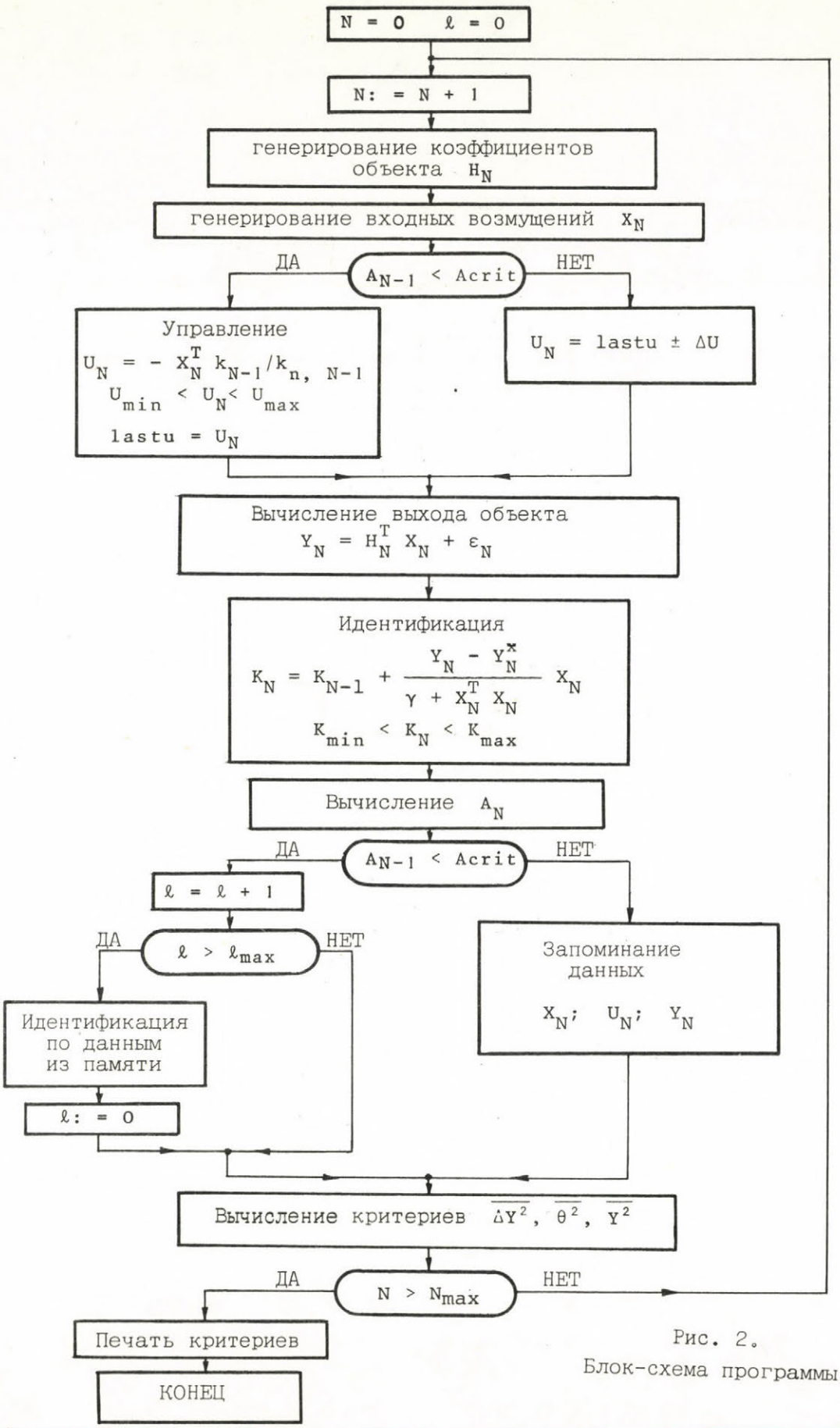


Рис. 2.
Блок-схема программы.

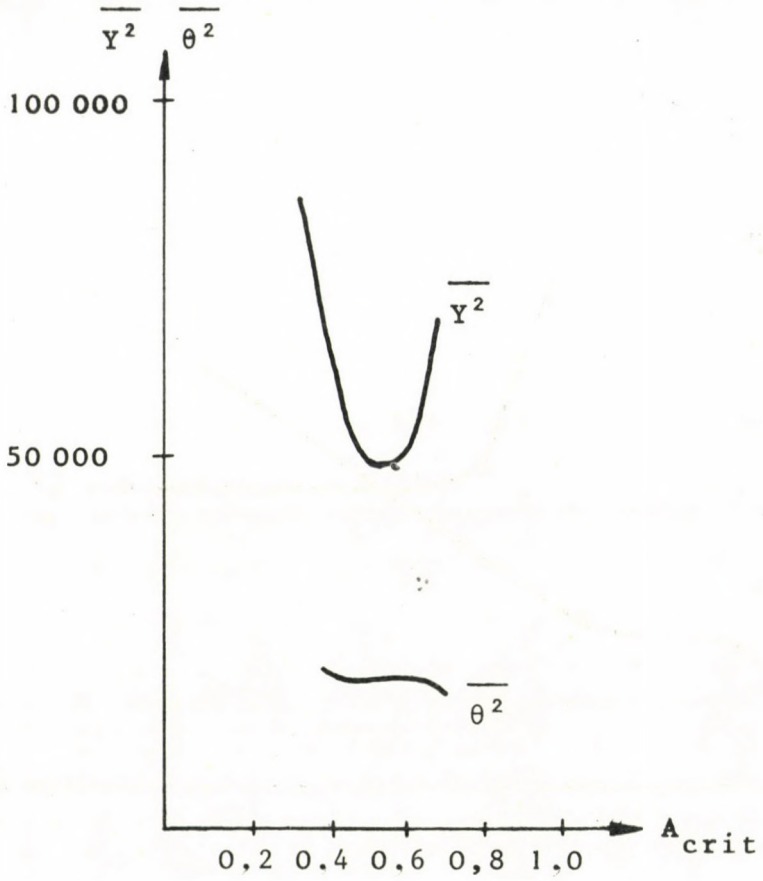


Рис. 3.

Моделирование при широких ограничениях

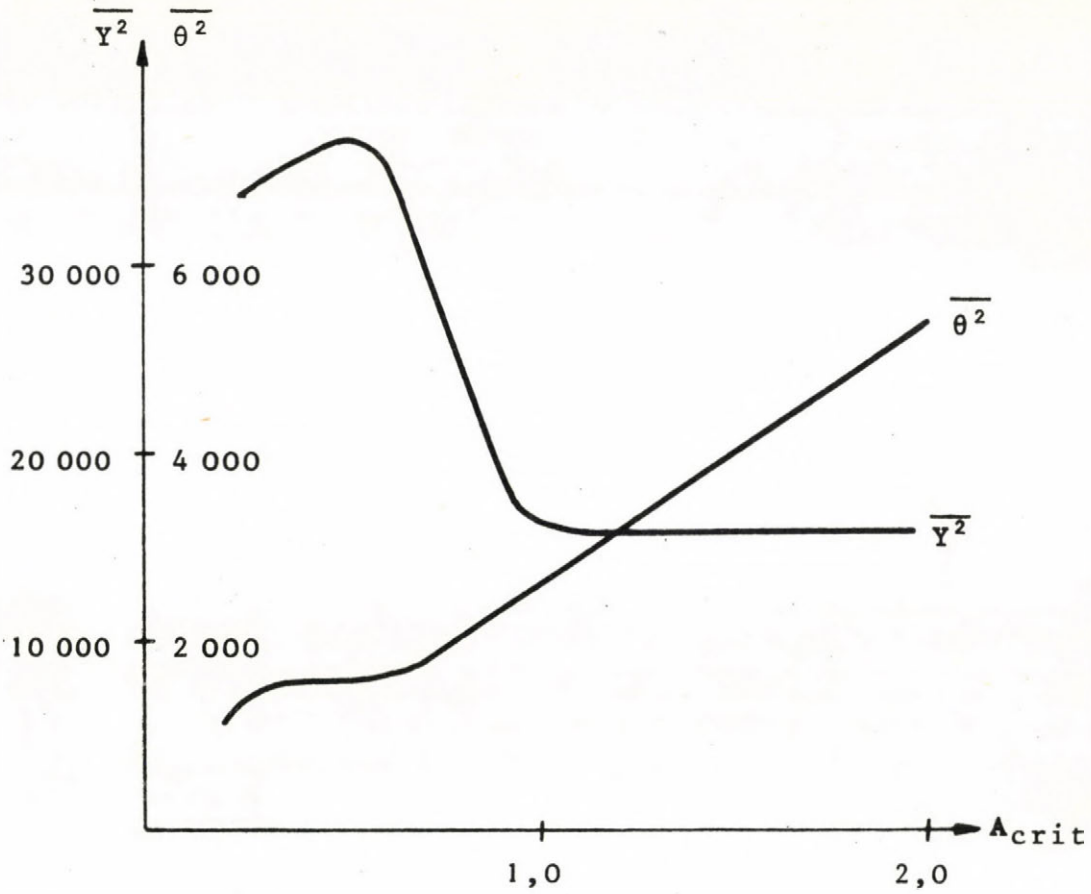


Рис. 4.

Моделирование при узких ограничениях